

## **Oracle Database 12c: High Availability New Features**

**Student Guide**

D79794GC10

Edition 1.0

July 2013

D82517

**ORACLE®**

**Authors**

Peter Fusek  
Anupama Mandya  
Mark Fuller  
Jim Womack

**Technical Contributors and Reviewers**

Andrey Gusev  
Larry Carpenter  
Dominique Jeunot  
Harendra Mishra  
Janet Stern  
Jim Williams  
Joel Goodman  
John McHugh  
Jonathan Creighton  
Mark Scardina  
Markus Michalewicz  
Prasad  
Bagal  
Raj Kammend  
Rick Wessman  
Subhransu Basu  
Harald van Breederode  
Sean Kim  
Douglas Williams  
Branislav Valny  
Allan Graves

**Editors**

Daniel Milne  
Anwesha Ray

**Graphic Designer**

Seema M Bopaiah

**Publishers**

Syed Imtiaz Ali  
Srividya Rameshkumar

**Copyright © 2013, 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.

## Contents

### **1 Introduction**

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

### **2 Flex Clusters**

- Objectives 2-2
- Flex Clusters: Overview 2-3
- Flex Cluster Architecture 2-4
- Flex Cluster Scalability 2-5
- Leaf Node Characteristics 2-6
- Grid Naming Service and Flex Clusters 2-7
- Cluster Mode Overview 2-8
- Configuring the Cluster Mode 2-9
- Configuring the Node Role 2-10
- Configuring the Hub Size 2-11
- Configuring Miss Count for Leaf Nodes 2-12
- Configuring a Flex Cluster with OUI: Selecting the Cluster Type 2-13
- Configuring a Flex Cluster with OUI: Configuring GNS 2-14
- Configuring a Flex Cluster with OUI: Selecting the Node Type 2-15
- Flex Clusters and Node Failure 2-16
- Quiz 2-17
- Summary 2-19
- Practice 2: Overview 2-20

### **3 Policy-Based Cluster Management**

- Objectives 3-2
- Policy-Based Cluster Management Enhancements: Overview 3-3
- Server Categorization 3-4
- Administering Server Categorization: Server Attributes 3-5
- Administering Server Categorization: Server Categories 3-6
- Administering Server Categorization: Server Pools 3-8
- Policy Set Overview 3-9

Policy-Based Cluster Management and QoS Management	3-11
Viewing the Policy Set	3-12
Configuring a User-Defined Policy Set: Method 1	3-13
Configuring a User-Defined Policy Set: Method 2	3-14
Modifying a User-Defined Policy Set	3-15
Activating a User-Defined Policy	3-16
Quiz	3-17
Summary	3-20
Practice 3 Overview: Configuring and Using Policy-Based Cluster Management	3-21

#### **4 What-If Command Evaluation**

Objectives	4-2
What-If Command Evaluation	4-3
Performing What-If Command Evaluation on Application Resources with CRSCTL	4-4
Performing What-If Command Evaluation on Oracle Clusterware Resources with CRSCTL	4-5
Formatting the Output for What-If Command Evaluation on Oracle Clusterware Resources	4-6
Performing What-If Command Evaluation with SRVCTL	4-7
Evaluating Failure Consequences with SRVCTL	4-8
Quiz	4-9
Summary	4-11
Practice 4 Overview: Using What-If Command Evaluation	4-12

#### **5 Other Clusterware New Features**

Objectives	5-2
Shared GNS: Overview	5-3
Shared GNS Architecture	5-4
Configuring a GNS Server Cluster	5-5
Configuring a GNS Client Cluster	5-6
Migrating to Shared GNS	5-7
Shared GNS Naming	5-9
Moving GNS to Another Cluster	5-10
Quiz	5-11
Cluster Health Monitor Enhancements: Overview	5-12
Cluster Health Monitor Services	5-13
Grid Infrastructure Management Repository	5-14
Managing Cluster Health Monitor: Routine Monitoring	5-15
Managing Cluster Health Monitor: New oclumon Commands	5-16

Grid Infrastructure Script Automation for Installation and Upgrade	5-17
Bundled Agents	5-18
IPv6 Support for VIPs and Network Agents	5-19
Quiz	5-20
Summary	5-21

## 6 Flex ASM

Objectives	6-2
Flex ASM: Overview	6-3
Flex ASM and Flex Clusters	6-4
ASM Instance Changes	6-5
ASM Network	6-6
ASM Listeners	6-7
ADVM Proxy	6-8
Configuring Flex ASM on a Standard Cluster	6-9
Configuring Flex ASM on a Flex Cluster	6-10
Managing Flex ASM Instances	6-11
Stopping, Starting, and Relocating Flex ASM Instances	6-12
Setting the Cardinality for Flex ASM Instances	6-13
Monitoring Flex ASM Connections	6-14
Relocating an ASM Client	6-15
Flex ASM Deployment Example	6-16
Quiz	6-18
Summary	6-21
Practice 6 Overview: Database Failover with Flex ASM	6-22

## 7 Other ASM New Features

Objectives	7-2
ASM Fast Mirror Resync: Review	7-3
Controlling the Resources Used by Resync	7-4
More Efficient Disk Replacement	7-5
Dealing with Transient Failure on a Failure Group	7-6
Resync Time Estimate	7-7
Resync Checkpoint and Auto-Restart	7-8
ASM Disk Group Rebalance: Review	7-9
Rebalance Work Estimates	7-10
Priority Ordered Rebalance	7-11
Proactively Validating Data Integrity	7-12
Proactive Content Checking During Rebalance	7-13
On-Demand Scrubbing	7-14
Errors During Scrubbing	7-15

Managing Password Files: Background	7-16
Storing Password Files Inside ASM	7-17
Managing Password Files Inside ASM	7-18
Even Read	7-19
Specifying the Content Type for a Disk Group	7-20
Improved Error Reporting for Cluster Validation Failures	7-22
Exadata Copy Offload	7-23
Bulk File Ownership Changes	7-24
Changing ASM Privileges on Open Files	7-25
ASM File Access Control Available on Windows	7-26
Quiz	7-27
Summary	7-29
Practice 7 Overview:	7-30

## 8 Cloud FS New Features

Objectives	8-2
Introducing Oracle Cloud File System	8-3
High Availability NFS: Overview	8-4
Configuring High Availability NFS	8-5
Cloud FS Snapshot Enhancements	8-6
Cloud FS Support for All Oracle Database Files	8-7
Configuration Settings for Database Files on Cloud FS	8-8
Cloud FS Auditing: Overview	8-9
Cloud FS Audit Trail Files	8-10
Cloud FS Audit Trail Contents	8-11
Configuring Cloud FS Auditing	8-12
Initializing Auditing Roles and Enabling Audit Sources	8-13
Enabling Auditing of Command Rules in a Security Realm	8-14
Managing the Audit Trail	8-15
Archiving Audit Files	8-16
Reviewing Audit Files	8-17
Purging Audit Files	8-18
Cloud FS Plug-in Infrastructure: Overview	8-19
Using the Cloud FS Plug-in	8-20
Using Cloud FS Replication in Conjunction with Cloud FS Security and Encryption	8-21
Generic API for Cloud FS Tagging	8-22
Cloud FS Resource Enhancements	8-23
Implementing Node-Specific File System Dependencies	8-25
Enhanced Platform Support for Cloud FS Data Services	8-26
Miscellaneous Cloud FS Enhancements	8-27

Quiz 8-28  
Summary 8-32  
Practice 8 Overview: 8-33

## **9 Application Continuity**

Objectives 9-2  
The Situation Prior to Application Continuity 9-3  
Introducing Transaction Guard and Application Continuity 9-4  
Key Concepts for Application Continuity 9-5  
Workflow of a Database Request 9-7  
What Is Transaction Guard? 9-8  
How Transaction Guard Works 9-9  
Using Transaction Guard 9-10  
Benefits of Transaction Guard 9-11  
What Is Application Continuity? 9-12  
How Does Application Continuity Work? 9-13  
Using Application Continuity 9-14  
Application Continuity Processing Phases 9-15  
Restrictions 9-17  
Potential Side Effects 9-18  
Actions That Disable Application Continuity 9-19  
When Is Application Continuity Transparent? 9-20  
Benefits of Application Continuity 9-21  
Application Assessment for Using Application Continuity 9-22  
Handling Request Boundaries 9-24  
Handling Request Boundaries: Example 9-25  
Disabling Replay by Using the disableReplay API 9-26  
Connection Initialization Options 9-27  
Mutable Objects and Application Continuity 9-29  
Keeping Mutable Objects for Replay 9-30  
Configuring the JDBC Replay Data Source 9-31  
Configuring Database Services for Application Continuity 9-32  
Resource Requirements for Application Continuity 9-33  
Quiz 9-34  
Summary 9-37  
Practice 9 Overview: Using Application Continuity 9-38

## **10 RAC New Features**

Objectives 10-2  
Lesson Overview 10-3  
RAC and Application Continuity 10-4

RAC and Flex ASM	10-5
RAC and Oracle Multitenant	10-6
RAC and Policy-Based Cluster Management	10-7
RAC and What-If Command Evaluation	10-8
Restricting Service Registration with Valid Node Checking	10-9
Role-Separated Installation Support for RAC on Windows	10-11
Summary	10-12

## **11 Oracle Data Guard New Features**

Objectives	11-2
Road Map	11-3
Data Guard 12c: Far Sync	11-4
Far Sync: Redo Transport	11-6
Far Sync: Alternate Redo Transport Routes	11-8
Far Sync: Role Transitions	11-9
Oracle Data Guard 12c: Far Sync Creation	11-11
Benefits: Far Sync	11-12
Far Sync: Alternate Design	11-13
Data Guard 11g, Release 2: Maximum Availability Protection Mode	11-14
Data Guard 12c: Fast SYNC	11-15
Data Guard 12c: Configuring Fast SYNC	11-16
Benefits: Fast Sync	11-17
Real-Time Cascade	11-18
Benefits: Real-Time Cascade	11-19
Road Map	11-21
Active Data Guard: DML on Temporary Tables	11-22
Active Data Guard: Support for Global Sequences	11-23
Active Data Guard: Support for Session Sequences	11-24
Benefits: Temporary Undo and Sequences	11-25
Road Map	11-26
Data Guard Simple Rolling Upgrades	11-27
Rolling Upgrades of Database Software	11-28
DBMS_ROLLING: Concepts	11-29
DBMS_ROLLING: Key Features	11-31
Database Rolling Upgrade: Specification and Compilation Stages	11-32
Specification Stage Examples	11-33
Compilation Stage Examples	11-34
Database Rolling Upgrade: Execution Stage	11-35
Logical Standby: New Data Type Support	11-36
Road Map	11-37
Simpler Role Transitions	11-38

Support for Moving Online Data Files	11-39
Support for Separation of Duties	11-40
Data Guard Support for Oracle Multitenant	11-41
Road Map	11-42
Data Guard Broker 12c Enhancements	11-43
Broker Validation for Role Changes	11-44
Broker Resumable Switchover	11-45
Broker Automatic Lag Monitoring	11-46
Configurable Broker Tracing	11-47
Broker Support for Far Sync	11-48
Broker Support for Complex Redo Routing	11-49
Defining RedoRoutes Using DGMGRL	11-50
RedoRoutes Usage Guidelines	11-51
How to Read Redo Routing Rules	11-52
Far Sync Example with RedoRoutes	11-53
Cascading Databases Example with RedoRoutes	11-54
Broker Support for Fast Sync	11-55
Using DBMS_ROLLING Package with Data Guard Broker	11-56
Quiz	11-57
Summary	11-59

## **12 Oracle Global Data Services Overview**

Objectives	12-2
Global Data Consolidation	12-3
Global Data Services	12-4
Oracle Global Data Services	12-5
The Global Data Services Framework	12-6
Logical Global Data Services Components	12-7
Logical Global Data Services Components: The Global Data Services Configuration	12-8
Logical Global Data Services Components: Global Data Services Pool	12-9
Logical Global Data Services Components: Global Services	12-10
Logical Global Data Services Components: Global Data Services Region	12-11
Physical Global Data Services Components: Global Service Manager	12-12
Physical Global Data Services Components: Global Data Services Catalog	12-14
Physical Global Data Services Components: Databases	12-15
Physical Global Data Services Components: Oracle Notification Servers	12-16
Physical Global Data Services Components: The gdsctl Utility	12-17
Global Service: Overview	12-18
Global Service Attributes	12-19
Global Services in a RAC Database	12-20

Global Services in an Data Guard Broker Configuration	12-21
Database Placement of a Global Service	12-23
Replication Lag and Global Services	12-25
Global Connection Load Balancing	12-26
Client-Side Load Balancing	12-27
Server-Side Load Balancing	12-28
Region Affinity for Global Services	12-29
Any-Region Affinity	12-30
Affinity to a Local Region	12-31
Affinity to a Local Region with Interregion Failover	12-32
Global Runtime Connection Load Balancing	12-33
Failover of Global Services	12-35
Role-Based Services	12-36
GDS Use Cases	12-38
GDS with Active Data Guard	12-39
GDS with GoldenGate (Multi-Master)	12-41
GDS with Active Data Guard	12-42
GDS Employing Reader Farms	12-44
GDS with GoldenGate (Master-Replica)	12-45
Oracle Database 12c Global Data Services: Summary	12-46
Quiz	12-47
Summary	12-49

# 1

## Introduction



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Course Objectives

After completing this course, you should be able to:

- Describe the Oracle Database 12c high availability new features contained in Oracle Grid Infrastructure (including Clusterware, Automatic Storage Management [ASM] and Cloud FS), Oracle Real Application Clusters (RAC) and Oracle Data Guard
- Perform essential installation and configuration tasks for each new feature
- Perform essential administration tasks for each new feature
- Describe Oracle Global Data Services (GDS)



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Oracle Database 12c, Oracle Cloud File System (Cloud FS) is the new name for the capabilities provided by ASM Cluster File System (ACFS) and ASM Dynamic Volume Manager (ADVM).

## Audience and Prerequisites

- This course is primarily designed for administrators of Oracle Grid Infrastructure, Oracle RAC, and Oracle Data Guard.
- Assumed prior knowledge:
  - Working knowledge of Oracle Database 11g, release 2, including Clusterware, ASM, RAC, and Data Guard
- Recommended prior training:
  - Oracle Grid Infrastructure 11g: Manage Clusterware and ASM release 2
  - Oracle Database 11g: RAC Administration release 2
  - Oracle 11g: Data Guard Administration



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This course is primarily designed for administrators of Oracle Grid Infrastructure, including Clusterware and Automatic Storage Management (ASM), Oracle Real Application Clusters (RAC), and Oracle Data Guard.

It is assumed that students have a working knowledge of Clusterware, ASM, RAC, and Data Guard in Oracle Database 11g, release 2.

For students that do not meet these prerequisites, the recommended prior training includes the following courses:

- *Oracle Database 11g: Manage Clusterware and ASM Release 2, and*
- *Oracle Database 11g: RAC Administration Release 2*
- *Oracle 11g: Data Guard Administration*

Alternatively, students may substitute the following course in place of the first two courses listed above:

- *Oracle 11g: RAC and Grid Infrastructure Administration Accelerated Release 2*

## Course Contents

1. Introduction (this lesson)
2. Flex Clusters
3. Policy-Based Cluster Management
4. What-If Command Evaluation
5. Other Clusterware New Features
6. Flex ASM
7. Other ASM New Features
8. Cloud FS New Features
9. Application Continuity
10. RAC New Features
11. Oracle Data Guard New Features
12. Oracle Global Data Services Overview



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows the ordering of lessons in this course.

## Additional Resources

- Recorded Demonstrations
  - <http://www.oracle.com/goto/oll>
  - Enter the Oracle Learning Library and search for demonstrations with “Oracle Database 12c” in the title.
- Oracle Technology Network (OTN) Oracle RAC page
  - <http://www.oracle.com/us/products/database/options/real-application-clusters/overview/index.html>
- OTN Oracle Data Guard Page
  - <http://www.oracle.com/goto/dataguard>
- OTN Database High Availability home page
  - <http://www.oracle.com/technetwork/database/features/availability/index.html>
- OTN Maximum Availability Architecture (MAA) home page
  - <http://www.oracle.com/technetwork/database/features/availability/maa-096107.html>
- OTN High Availability Discussion Forums
  - <https://forums.oracle.com/forums/category.jspa?categoryId=509>



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 1 Overview: Laboratory Introduction

In this practice, you will familiarize yourself with the laboratory environment for this course.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# 2

## Flex Clusters

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the Flex Cluster architecture and components
- Install and configure a Flex Cluster
- Describe the effect of node failure in a Flex Cluster



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Flex Clusters: Overview

- In previous releases, most large clusters contain between 32 and 64 nodes.
- With Oracle Clusterware 12c, Flex Clusters are designed to scale well up to 2,000 nodes. Use cases include:
  - Large pools of highly available application resources
  - Multiple databases and applications running in one cluster



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

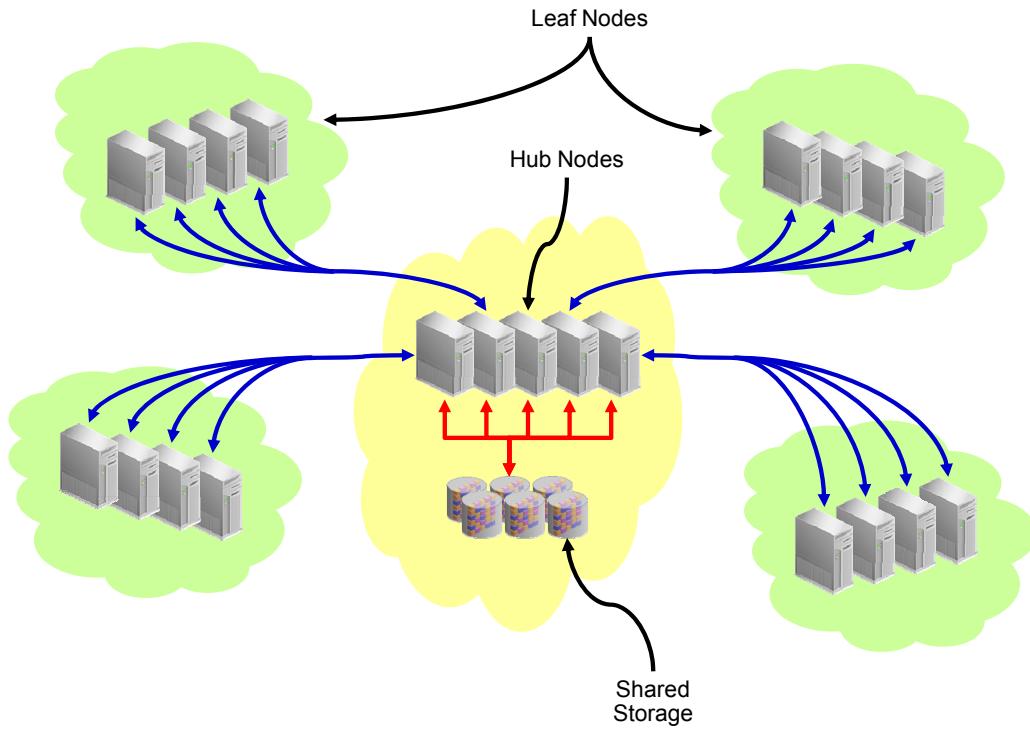
Previous releases of Oracle Clusterware have been used to build large production clusters containing between 32 and 64 nodes. A few clusters larger than 100 nodes have been successfully deployed.

With Oracle Clusterware 12c, a new set of features enables Flex Clusters. In this release, Flex Clusters are designed to scale well up to 2,000 nodes.

In release 12.1, you can use Flex Clusters to:

- Manage large pools of application resources with high-availability and failover protection.
- Efficiently support multiple highly available databases and applications running in a single cluster.

# Flex Cluster Architecture



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flex Clusters use a hub-and-spoke topology, as illustrated in the slide.

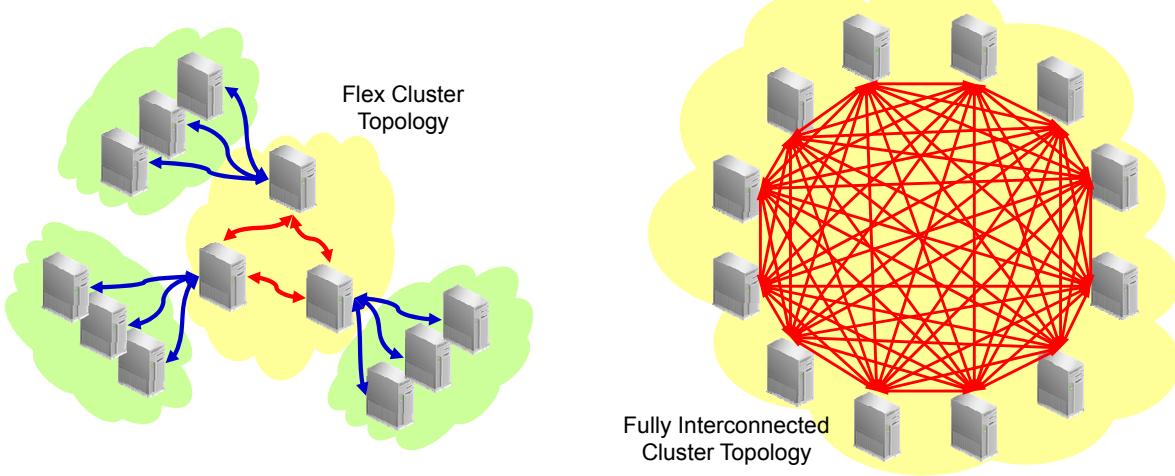
The core of a Flex Cluster is a group of Hub Nodes. The group is essentially the same as a release 11.2 cluster, and can scale up to the size of an existing release 11.2 cluster. There must be one, and only one, group of Hub Nodes in a Flex Cluster deployment, and like a release 11.2 cluster, each Hub Node must be connected to storage that is shared across the group of Hub Nodes.

Zero or more Leaf Nodes may be connected to a Flex Cluster. Each Leaf Node is connected to the cluster through a Hub Node. Leaf Nodes do not require direct access to the shared storage connected to the Hub Nodes.

## Flex Cluster Scalability

The Flex Cluster hub-and-spoke topology segments the cluster into more manageable groups of nodes.

- Only the Hub Nodes require direct access to the OCR and voting disks.
- Fewer interactions are required between nodes.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

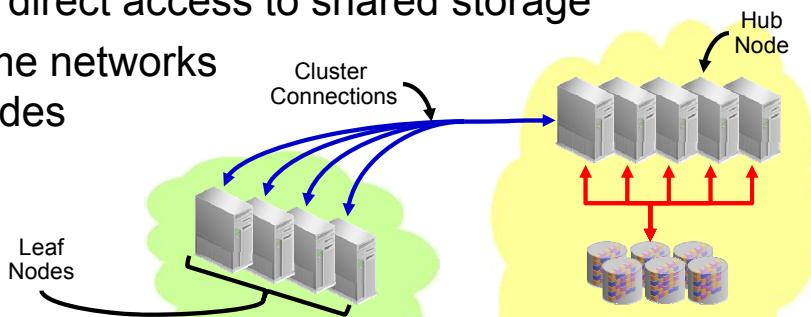
The two-layered hub-and-spoke topology is the key architectural feature that allows a Flex Cluster to scale well beyond previous limits. In essence, the hub-and-spoke topology segments the cluster into groups of nodes, and each group contains a manageable number of nodes. This segmentation has two fundamental impacts:

- First, by limiting the size of the hub, contention for key Clusterware resources, such as the Oracle Cluster Registry (OCR) and voting disks, does not increase significantly due to the addition of the Leaf Nodes. This is important because contention for the voting disks can lead to nodes being evicted from a cluster.
- Second, fewer network interactions are required between nodes in the cluster. Consequently, there is less administrative network traffic, such as heartbeats, exchanged between the nodes. This is illustrated in the diagram in the slide. On the left side, the 12-node Flex Cluster contains 12 interaction paths. On the right side, the fully interconnected 12-node cluster contains 66 possible interaction paths. For a 1000-node cluster, the difference would be far more noticeable. Assuming 40 Hub Nodes, with 24 Leaf Nodes per Hub Node, a Flex Cluster contains 1740 possible interaction paths. In comparison, a 1000-node fully interconnected cluster contains 499500 interaction paths.

## Leaf Node Characteristics

### Leaf Nodes:

- Are more loosely coupled to the cluster than Hub Nodes
- Automatically discover the Hub Nodes at startup
- Connect to the cluster through a Hub Node
  - Failure of the Hub Node or network failure results in eviction of associated Leaf Nodes.
  - Functioning Leaf Nodes can be brought back into the cluster.
- Do not require direct access to shared storage
- Are on the same networks as the Hub Nodes



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In comparison to Hub Nodes, Leaf Nodes are loosely coupled to the cluster. When Oracle Clusterware is started on a Leaf Node, the Leaf Node automatically discovers the Hub Nodes and is associated with a single Hub Node. For cluster membership purposes, Hub Nodes periodically exchange heartbeat messages with their associated Leaf Nodes. Similar mechanisms are used for other services.

If a Hub Node fails, or if the network link between a Hub Node and a Leaf Node fails, the associated Leaf Nodes may be removed from the cluster. In any case, if there is no fault with the Leaf Node, it can be brought back into the cluster by restarting Oracle Clusterware on the Leaf Node.

A Leaf Node does not require direct access to shared storage. This means that Leaf Nodes can participate in the cluster without storage-related hardware and network connections, such as Fibre Channel network connections and host bus adapters.

In release 12.1, all Leaf Nodes are on the same public and private networks as the Hub Nodes.

## Grid Naming Service and Flex Clusters

Clients on Leaf Nodes use GNS to locate Hub Node services.

- The GNS server location is stored in the cluster profile.
- Leaf Node services issue DNS queries to GNS.
  - Particularly during Leaf Node startup
- A fixed GNS VIP is required on one of the Hub Nodes.
  - So Leaf Node clients have a reliable, well-known location to contact.
- DNS forwarding is not required for Flex Clusters.
  - But it can be implemented to better integrate GNS with DNS.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flex Clusters require the Grid Naming Service (GNS) to be configured with a fixed virtual IP (VIP) on one of the Hub Nodes.

GNS is used to dynamically register and resolve names within the cluster. In particular, GNS is referenced as the Leaf Node is associated with a Hub Node for cluster membership purposes during the Clusterware start-up process on a Leaf Node. This requires access to GNS through a fixed VIP running on one of the Hub Nodes, so that Leaf Nodes have a reliable, well-known naming service within the cluster.

Domain name server (DNS) forwarding is not required to facilitate discovery of Clusterware services by Leaf Nodes; however, it can still be configured to integrate GNS with a wider network-naming service.

## Cluster Mode Overview

- Oracle Clusterware 12c introduces a new cluster mode setting to enable Flex Cluster functionality.
  - Users must explicitly enable Flex Cluster functionality.
- The default cluster mode setting disables Flex Cluster functionality.
  - Users who do not implement Flex Clusters are not exposed to the new code.
  - Performance and stability of standard clusters is not impacted by Flex Cluster functionality.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can choose to enable or disable Flex Cluster functionality by using the new cluster mode setting. By default, Flex Cluster functionality is disabled.

# Configuring the Cluster Mode

- Showing the current cluster mode:

```
$ crsctl get cluster mode status
```

- Converting from a standard cluster to a Flex Cluster:

- Ensure that GNS is configured with a fixed VIP:

```
# srvctl add gns -i <Fixed GNS VIP address> -d <cluster domain>
```

- Enable Flex ASM in the cluster using ASMCA.

- Set the cluster mode:

```
# crsctl set cluster mode flex
```

- Stop Oracle Clusterware on each node:

```
# crsctl stop crs
```

- Start Oracle Clusterware on each node:

```
# crsctl start crs
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

At any time, a cluster administrator or system administrator can check the current cluster mode by using the `crsctl get cluster mode status` command.

To convert from a standard cluster to a Flex Cluster, an administrator should first ensure that GNS is configured with a fixed VIP. If GNS is not configured with a fixed VIP, the remainder of the procedure will fail. In addition, Flex ASM must be enabled on the cluster prior to setting the cluster mode (as discussed in the “Flex ASM” lesson). Next, the system administrator (root) can set the cluster mode by using the `crsctl set cluster mode flex` command. Finally, the system administrator must restart the cluster by using the `crsctl stop crs` command on each node in the cluster followed by the `crsctl start crs` command on each node in the cluster. Note that you cannot avoid cluster downtime when changing the cluster mode.

To convert from a Flex Cluster to a standard cluster, the system administrator must set the cluster mode by using the `crsctl set cluster mode standard` command and Clusterware must be stopped and restarted across the cluster. There is no requirement to reconfigure GNS or Flex ASM, because the configuration required for Flex Cluster mode is also compatible with standard cluster mode.

Note that any node that is unable to join the reconfigured cluster is left out of the cluster and eventually dropped from it. This can occur when, for example, a Leaf Node having no access to shared storage cannot join a cluster converted to standard mode.

# Configuring the Node Role

- Showing the current node role:

```
$ crsctl get node role status -node <hostname>
```

```
$ crsctl get node role status -node c00n02  
Node 'c00n02' active role is 'hub'
```

- Setting the node role:

```
# crsctl set node role { hub | leaf | auto } -node <hostname>
```

```
# crsctl set node role leaf -node c00n02  
# crsctl get node role config -node c00n02  
Node 'c00n02' configured role is 'leaf'  
# crsctl get node role status -node c00n02  
Node 'c00n02' active role is 'hub'
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

One of the key configuration tasks for a Flex Cluster is specifying which nodes are Hub Nodes and which nodes are Leaf Nodes.

At any time, a cluster administrator or system administrator can check the current role for a node by using the following command:

```
crsctl get node role status -node <hostname>
```

Configuring the node role can be achieved in two ways:

- A system administrator can explicitly specify the node role as `hub` for a Hub Node, or `leaf` for a Leaf Node, by using the `crsctl set node role` command. Explicitly setting the node role ensures the node type. The example in the slide shows the node `c00n02` being configured as a Leaf Node. Note that node role changes do not take effect until the next time that Oracle Clusterware is started on the node. This is evident in the example in the slide, where the configured node role is `leaf` but the active node role is still `hub` immediately after the node role change.
- A system administrator can also set the node role to `auto` by using the `crsctl set node role` command. This setting allows the cluster to decide which role a node will perform based on the composition of the cluster. The cluster administrator must ensure that the node can fulfill either role, `hub` or `leaf`, in order to use the `auto` setting.

# Configuring the Hub Size

- Showing the current hub size:

```
$ crsctl get cluster hubsize  
CRS-4950: Current hubsize parameter value is 32
```

- Setting the hub size:

```
# crsctl set cluster hubsize <number>
```

```
# crsctl set cluster hubsize 16  
# crsctl get cluster hubsize  
CRS-4950: Current hubsize parameter value is 16
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `auto` node role setting works hand-in-hand with the `cluster hubsize` setting. When a node with the `auto` node role setting attempts to join the cluster, Oracle Clusterware examines the `hubsize` setting. If the number of Hub Nodes is smaller than the `hubsize` setting, the node joins the cluster as a Hub Node. Otherwise, the node joins the cluster as a Leaf Node.

The examples in the slide show how administrators can examine and set the cluster hub size. Note that setting the cluster hub size requires system administrator privileges. Note also that the hub size setting is effective immediately, but it does not impact the node role of any nodes already in the cluster.

# Configuring Miss Count for Leaf Nodes

Viewing and setting leafmisscount:

```
# crsctl get css leafmisscount
CRS-4678: Successful get leafmisscount 30 for Cluster Synchronization
Services.
# crsctl set css leafmisscount 45
CRS-4684: Successful set of parameter leafmisscount to 45 for Cluster
Synchronization Services.
# crsctl get css leafmisscount
CRS-4678: Successful get leafmisscount 45 for Cluster Synchronization
Services.
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

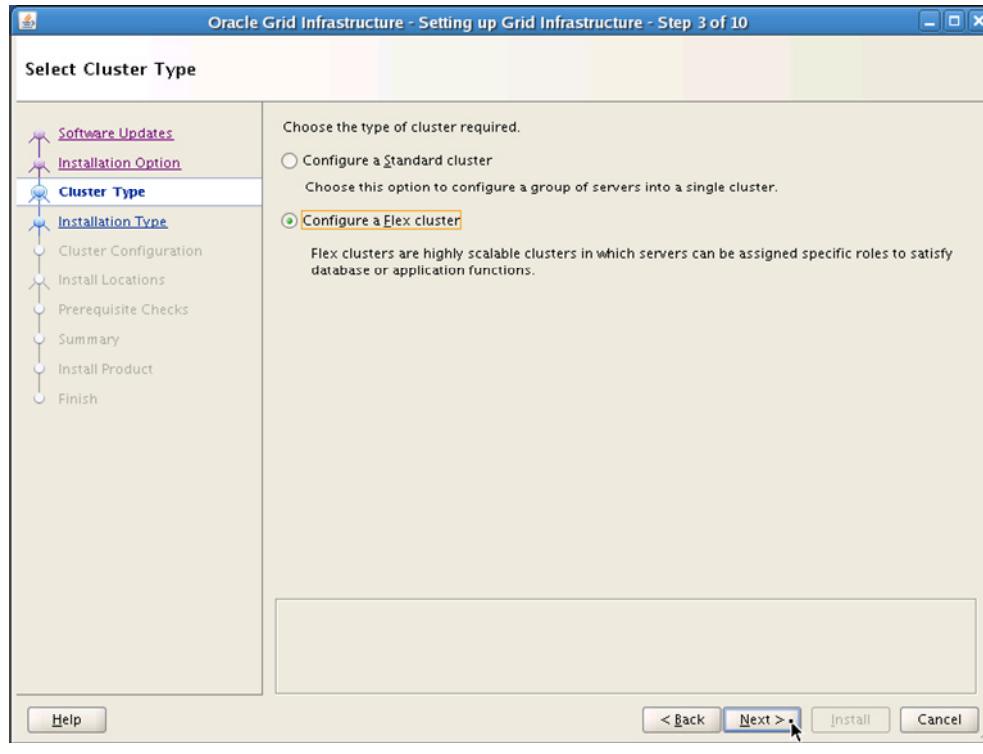
The `leafmisscount` attribute determines how Cluster Synchronization Services (CSS) handles network connectivity issues between a Leaf Node and the Hub Node that connects it to the cluster.

The `leafmisscount` setting defines the threshold duration (in seconds) for tolerable communication failures. If communication between a Hub Node and associated Leaf Node is interrupted and restored before the amount of time specified by `leafmisscount`, then the cluster continues to operate normally. If communication is lost for a period exceeding the `leafmisscount` setting, then the interruption is assumed to be significant and the Leaf Node is evicted from the cluster. The default `leafmisscount` setting is 30 seconds.

The examples in the slide show how to query and set the `leafmisscount` attribute.

Note that the `leafmisscount` attribute is separate from the `misscount` attribute that existed in previous releases and that continues to exist in release 12.1.

# Configuring a Flex Cluster with OUI: Selecting the Cluster Type



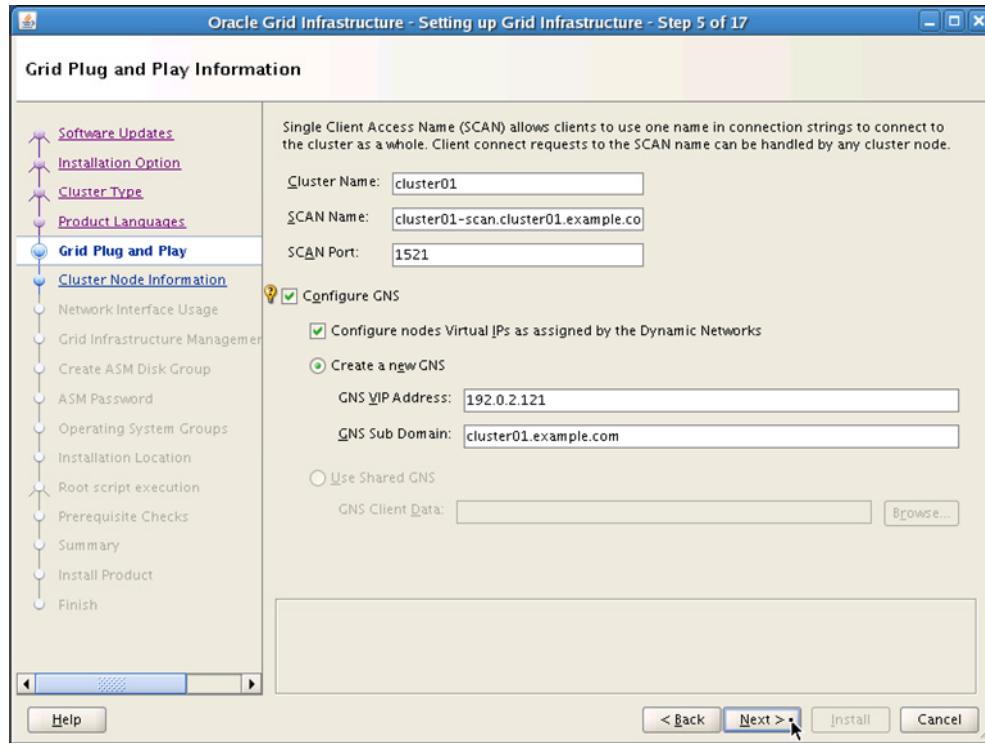
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

The previous four pages introduced the commands required to configure a Flex Cluster and convert a standard cluster to a Flex Cluster. For new clusters, Oracle Universal Installer (OUI) has been updated to facilitate the configuration of Flex Clusters.

The screenshot in the slide shows the OUI interface for the step where administrators select the cluster type. To configure a Flex Cluster, administrators must select the “Configure a Flex cluster” option.

# Configuring a Flex Cluster with OUI: Configuring GNS

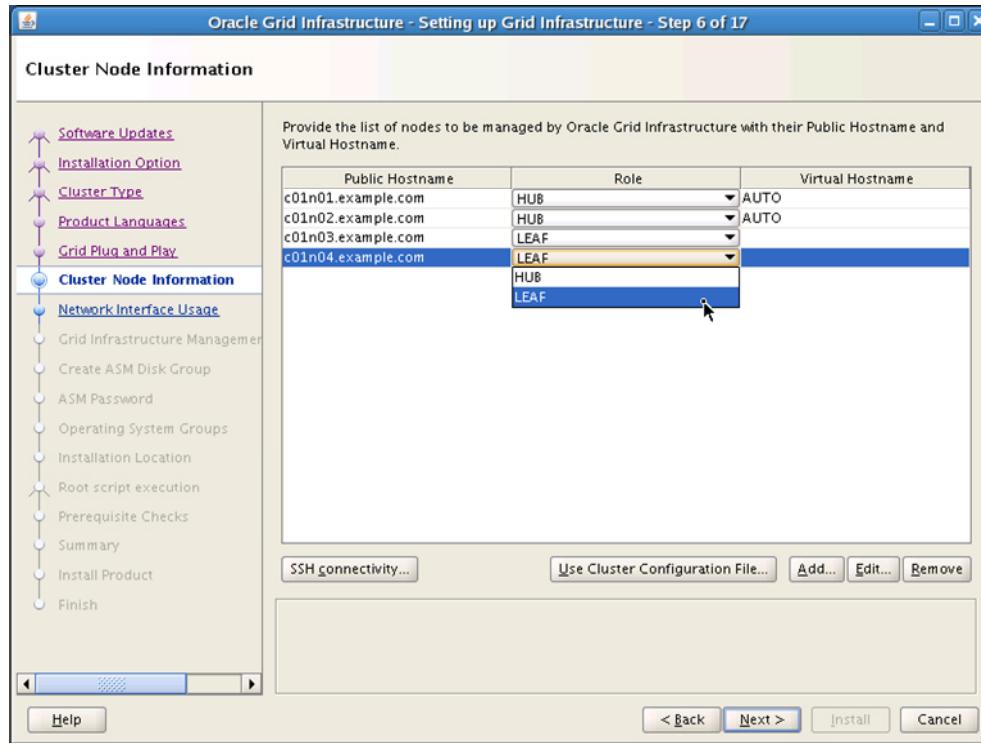


ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flex Clusters require the Grid Naming Service (GNS) to be configured with a fixed virtual IP (VIP) on one of the Hub Nodes. In line with this, OUI will not progress before GNS is configured on the “Grid Plug and Play Information” screen. The screenshot in the slide shows an example of the required configuration.

# Configuring a Flex Cluster with OUI: Selecting the Node Type



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When “Configure a Flex cluster” is selected in the Select Cluster Type screen, the Cluster Node Information screen looks like the example shown on this slide. In this interface, the user may specify the cluster nodes. For each cluster node, the user can set the node type to HUB or LEAF.

A cluster that is created with this method is configured with nodes that have explicitly defined roles.

To configure the cluster by using the `auto` node role setting in conjunction with the cluster `hubsize` setting, administrators must wait until after the cluster is initially configured with OUI. Then, they can use the commands introduced earlier in this lesson to adjust the node role and cluster `hubsize` settings.

## Flex Clusters and Node Failure

- Nodes that are evicted from the cluster do not require a restart; only a cluster software restart.
- If a Hub Node fails:
  - The node is evicted from the cluster.
    - Services are relocated to other Hub Nodes if possible.
  - Corresponding Leaf Nodes are also evicted from the cluster.
    - Services are relocated to other Leaf Nodes if possible.
- If a Leaf Node fails:
  - The node is evicted from the cluster.
    - Services are relocated to another Leaf Node where possible.
  - The impact of the failure is contained where possible.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Earlier, the effect of network connectivity issues between Hub and Leaf Nodes was introduced. Now, the effect of node failure in Flex Clusters is discussed.

In previous releases of Oracle Clusterware, any node that was evicted from the cluster would panic the operating system kernel and cause an immediate shutdown of the node. This measure is very effective at protecting the cluster from the effects of a rogue node; however, restarting the node results in the cluster working with diminished capacity for many minutes. With Oracle Clusterware 12c, node eviction does not require a node restart. Rather, where possible, only the cluster software is restarted, which significantly reduces the amount of time that the cluster is affected.

If a Hub Node fails, the node is evicted from the cluster in essentially the same way as any node in a standard cluster. As part of dealing with the node failure, the cluster attempts to start as many services as possible on surviving cluster nodes. Leaf Nodes that were associated with the failed Hub Node are also evicted from the cluster.

If a Leaf Node fails, the node is evicted from the cluster and the cluster attempts to relocate services running on the Leaf Node to other Leaf Nodes connected to the same Hub Node. This means that the effect of a Leaf Node failure is usually contained within the group of Leaf Nodes connected to the same Hub Node. Thus, the performance and availability of the rest of the cluster is not affected by the failure of a Leaf Node.

## Quiz

Identify the use cases supported by Flex Clusters:

- a. Large-scale decision support databases where parallel query operations can be spread across database instances running on the Leaf Nodes
- b. Large-scale online transaction processing (OLTP) databases where many thousands of user connections can be spread across database instances running on the Leaf Nodes
- c. Mixed environments where databases run on the Hub Nodes and highly available application resources run on the Leaf Nodes



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: c

In release 12.1, Oracle Database instances are only supported on Hub Nodes. Leaf Nodes can be used to host application services, which can leverage the high-availability framework of Oracle Clusterware.

## Quiz

Flex Clusters achieve greater scalability than standard clusters because:

- a. Fewer physical network connections are required between the cluster nodes.
- b. Leaf Nodes do not require direct access to shared storage.
- c. By limiting the size of the hub, contention for key Clusterware resources is controlled.
- d. Fewer network interactions are required between the cluster nodes to maintain the cluster.
- e. The cluster hub size can be set to a larger value than in previous versions.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: c, d

The two-layered hub-and-spoke topology of a Flex Cluster achieves greater scalability because of two fundamental reasons:

1. By limiting the size of the hub, contention for key Clusterware resources, such as the OCR and voting disks, does not increase significantly because of the addition of the Leaf Nodes.
2. Fewer network interactions are required between nodes in the cluster, and consequently there is less administrative network traffic, such as heartbeats, exchanged between the nodes.

Answer A is not correct because the number of physical network connections is the same for both cluster types.

Answer B is a correct statement; however, this fact does not by itself improve cluster scalability.

Answer E is not correct because the cluster hub size setting does not by itself improve cluster scalability.

## Summary

In this lesson, you should have learned how to:

- Describe the Flex Cluster architecture and components
- Install and configure a Flex Cluster
- Describe the effect of node failure in a Flex Cluster



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 2: Overview

- Practice 2-1: Configuring a New Flex Cluster with Flex ASM and RAC
  - In this practice, you will install and configure a new Flex Cluster with Flex ASM and a RAC Database.
- Practice 2-2: Configuring Highly Available Application Resources on Flex Cluster Leaf Nodes
  - In this practice, you will create a series of highly available application resources running on one of the Flex Cluster Leaf Nodes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Policy-Based Cluster Management

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the architecture and components of policy-based cluster management
- Administer server categorization
- Administer a policy set
- Activate a policy



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Policy-Based Cluster Management Enhancements: Overview

In previous releases:

- A cluster can be logically divided into server pools.
  - The server pools collectively define a policy.
  - All nodes are assumed to be equal.
- Quality of Service Management uses a different policy.
  - Potential for overlap, confusion, and inconsistency

With Oracle Clusterware 12c, policy-based cluster management is enhanced to provide:

- Extended server attributes to govern node placement
- A library of policy definitions with an easy way of switching between policies
- Unification of policies for Oracle Clusterware and Quality of Service Management



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

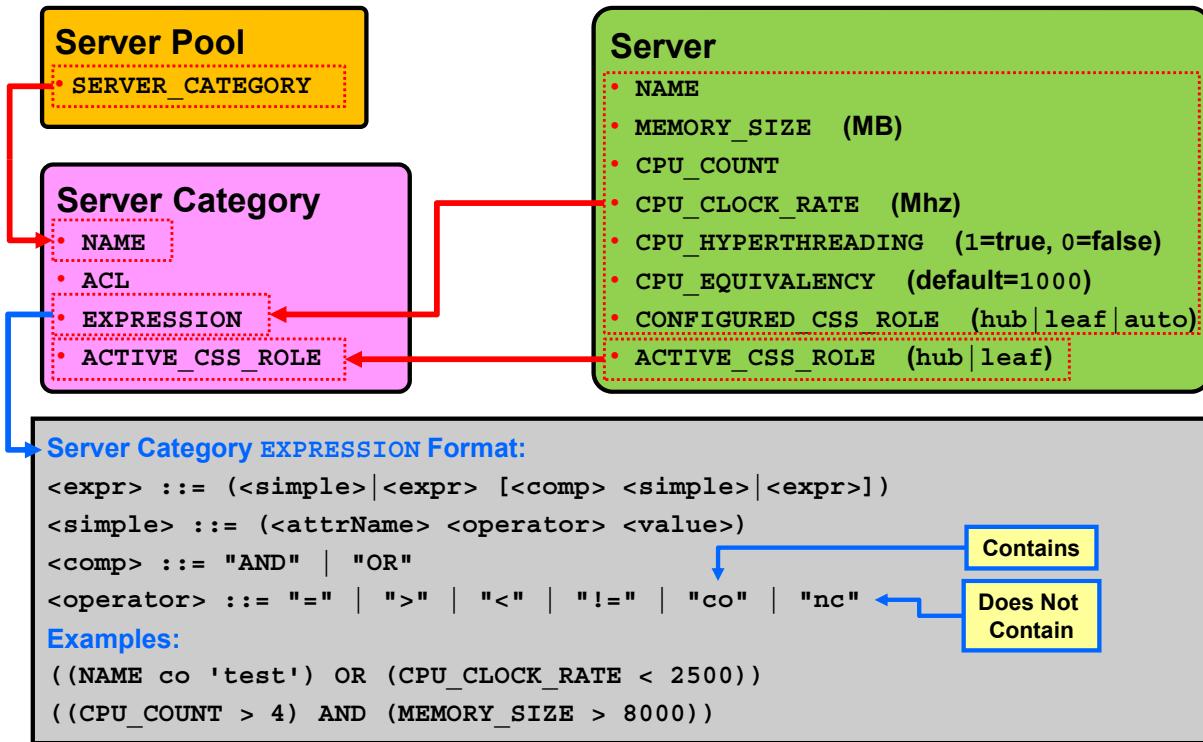
Oracle Clusterware 11g, release 2 introduced policy-based cluster management. With this capability, a cluster can be logically divided into groups of servers known as server pools. The placement of nodes in each server pool is governed by the relative importance assigned to each server pool, along with other attributes such as the minimum and maximum number of nodes assigned to each server pool.

The server pool definitions effectively define a policy, which enables dynamic capacity assignment and fast resource failover when the number of nodes in the cluster changes. With release 11.2, all nodes are assumed to be equal; that is, there is no way to specify server attributes that favor server placement in a particular server pool.

With release 11.2, the Quality of Service (QoS) Management feature defines a separate policy for cluster resource management, which can be confusing for administrators unless they are familiar with all the policies. In addition, there exists a potential to create policies that are contrary to each other.

With Oracle Clusterware 12c, policy-based cluster management is enhanced in three important ways. First, numerous server attributes allow for greater flexibility and control over node assignments to different server pools. Second, an extended policy framework allows administrators to maintain a library of policies and easily switch between them as required. Finally, policy-based cluster management has been unified with QoS Management.

# Server Categorization



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, the assignment of servers to server pools was based on the relative importance of each server pool and a few basic attributes, such as the minimum and maximum number of servers associated with the server pool. Because there was no way to differentiate between servers, all servers were assumed to be homogeneous with respect to CPU, memory, and other resources.

With Oracle Clusterware 12c, the notion of server categorization is introduced. Categorization allows servers to be differentiated and provides a mechanism for automatically controlling the composition of each sever pool. Server categorization works as follows:

- Every server contains a set of new attributes. Most of the attributes specify key physical characteristics of the server, such as MEMORY\_SIZE and CPU\_COUNT, or they contain configuration settings relating to Oracle Clusterware, such as CONFIGURED\_CSS\_ROLE. For a complete list of server attribute definitions, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.
- A new Clusterware object defines server categories. The main attribute of each server category is an expression that is evaluated against the attributes of each server to determine whether the server belongs to the category.
- A new attribute, SERVER\_CATEGORY, is added to the each server pool definition. This attribute allows a server category to be associated with each server pool, thereby governing which servers can be in the pool.

## Administering Server Categorization: Server Attributes

- Most server attributes are automatically discovered by Oracle Clusterware.
- Viewing attribute settings example:

```
$ crsctl status server c00n01 -f
NAME=c00n01
MEMORY_SIZE=4006
CPU_COUNT=1
CPU_CLOCK_RATE=2857
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=UNAVAILABLE
PHYSICAL_HOSTNAME=UNAVAILABLE
STATE=ONLINE
ACTIVE_POOLS=Free
STATE_DETAILS=
ACTIVE_CSS_ROLE=hub
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Most server attribute settings are automatically discovered by Oracle Clusterware. An exception is the SERVER\_LABEL attribute, which administrators can set by using the crsctl modify server command.

Administrators can view the server attributes by using the crsctl status server command. An example is displayed in the slide. Note the use of the -f option to display a full listing of all server attributes.

# Administering Server Categorization: Server Categories

- Creating a new server category:

```
$ crsctl add category <catName> -attr "<attrName>=<value>[,...]"  
$ crsctl add category small -attr "EXPRESSION='(CPU_COUNT = 1)'"
```

- Modifying an existing server category:

```
$ crsctl modify category <catName> -attr "<attrName>=<value>[,...]"  
$ crsctl modify category small -attr "ACTIVE_CSS_ROLE='hub'"
```

- Viewing a category:

```
$ crsctl status category <catName>  
$ crsctl status category small  
NAME=small  
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--  
ACTIVE_CSS_ROLE=hub ←  
EXPRESSION=(CPU_COUNT = 1) ←
```

- Deleting a category:

```
$ crsctl delete category <catName>
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows examples of the commands that can be used to create, modify, view, and delete a server category.

When creating or modifying a server category, note that the key attribute is the EXPRESSION that defines which servers can belong to the category. With the ACTIVE\_CSS\_ROLE attribute, administrators can specifically define different server categories for Hub Nodes and for Leaf Nodes. The ACTIVE\_CSS\_ROLE attribute should not be referenced in the EXPRESSION string.

# Administering Server Categorization: Server Categories

- Listing servers in a category:

```
$ crsctl status server -category <catName>
$ crsctl status server -category small
NAME=c00n01
STATE=ONLINE
...
```

- Listing categories for a server:

```
$ crsctl status category -server <serverName>
$ crsctl status category -server c00n01
NAME=ora.hub.category
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=

NAME=small
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
ACTIVE_CSS_ROLE=hub
EXPRESSION=(CPU_COUNT = 1)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After a category is defined, it may be useful to understand which servers are members of that category. This can be achieved by using the `crsctl status server` command with the `-category` option, as shown in the slide.

It is also possible to list all of the categories that apply to a specific server by using the `crsctl status category` command with the `-server` option. Note that a server can belong to multiple categories at the same time, as shown in the example in the slide. The `ora.hub.category` category is an internal category that is used to categorize Hub Nodes.

## Administering Server Categorization: Server Pools

- Specifying the SERVER\_CATEGORY attribute:

```
$ crsctl add serverpool hr -attr "SERVER_CATEGORY='medium'" ...
$ crsctl modify serverpool dev -attr "SERVER_CATEGORY='small'"
```

- Viewing the SERVER\_CATEGORY attribute:

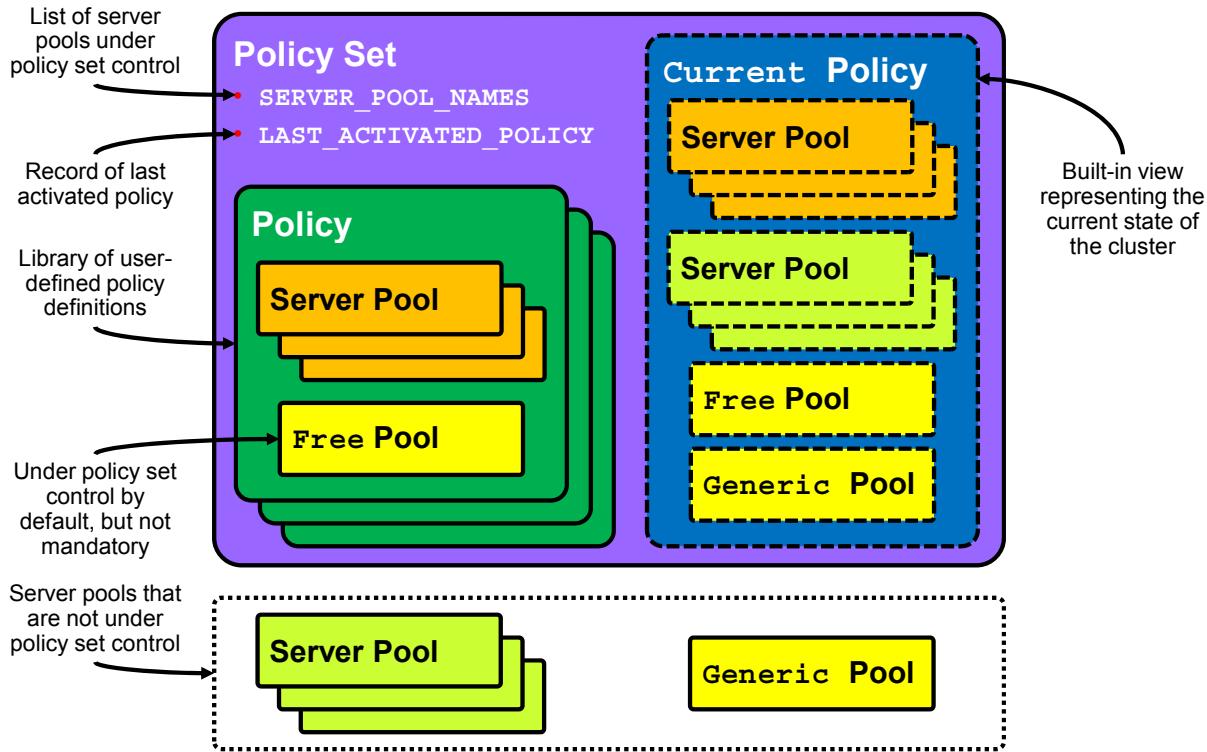
```
$ crsctl status serverpool dev -f
NAME=dev
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--
SERVER_CATEGORY=small
ACTIVE_SERVERS=c00n01 c00n02
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Server categories are applied to server pools by using the new SERVER\_CATEGORY attribute. This attribute can be specified for new and existing server pools, as shown in the examples in the slide. To view the setting of the SERVER\_CATEGORY attribute, use the crsctl status serverpool command with the -f option.

# Policy Set Overview



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide illustrates the policy set contained in Oracle Clusterware 12c.

There is always exactly one policy set defined and used by the cluster. The policy set contains two attributes. The SERVER\_POOL\_NAMES attribute defines the names of all server pools controlled by the policy set. The policy set also contains an attribute that records the last activated policy.

The policy set contains zero or more user-defined policies. Each policy contains exactly one definition for each server pool controlled by the policy set. Typically, administrators create policies to address different priorities at different times.

Server pools may also exist outside the control of the policy set. For example, a server pool may be created with the `srvctl add serverpool` command and not be listed in the SERVER\_POOL\_NAMES policy set attribute. Server pools that are outside the control of the policy set are not directly affected by policy changes; however, such server pools may be indirectly affected by policy changes.

For example, a server pool outside the control of the policy set must yield a server when a policy change increases the number of servers allocated to a server pool with a higher priority, and there are no free servers or lower priority server pools elsewhere. Likewise, a server pool outside policy set control may grow in size because a policy change frees up a server.

The policy set always contains a special built-in policy, named `Current`, representing the configuration that is currently in effect. The `Current` policy includes all server pools that are not under policy set control. When a user-defined policy is activated, its attributes are reflected in the `Current` policy. Over time, the `Current` policy may cease to reflect the last activated policy because of changes made outside the policy set; for example, when a server pool associated with a release 11.2 policy-managed database is added to the system.

In previous versions, two built-in server pools existed: `Generic` and `Free`. With Oracle Clusterware 12c, these built-in server pools remain; however, they are handled differently. By default, the `Free` server pool is implicitly controlled by the policy set. However, administrators can choose to remove the `Free` server pool from the `SERVER_POOL_NAMES` list if they want to place the `Free` pool outside direct policy set control. The `Generic` server pool is never under direct policy set control. It is listed as a server pool in the `Current` policy view.

# Policy-Based Cluster Management and QoS Management

Two methods for configuring and running policy-based cluster management:

- User-defined policy management
  - Clusterware administrators manually configure the policy set.
  - Clusterware administrators activate different policies as required.
    - Administrators can use a job scheduling system to automatically activate specific policies at different times.
- Quality of Service (QoS) Management
  - QoS Management interfaces are used to configure the policy set.
    - Administrators cannot directly modify the policy set.
  - QoS Management automatically adjusts resource allocations in response to workload demands.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are two distinct modes of operation that apply to policy-based cluster management.

With user-defined policy management, Clusterware administrators manually configure the policy set, policies, server pools, and their attributes. After configuration, it is the responsibility of the Clusterware administrator to activate the required policy. Policies can also be activated automatically by using a job scheduling system or another program that uses the supplied commands and application programming interfaces (APIs).

During implementation of QoS Management, the QoS Management interfaces are used to configure a QoS Management policy set, which also contains a Clusterware policy set definition. When QoS Management is activated, the associated Clusterware policy set definition is activated and locked so that Clusterware administrators cannot manually modify the policy set. This allows QoS Management to automatically adjust policy settings in order to fulfill the prescribed performance objectives.

In essence, QoS Management extends the functionality that is available with user-based policy management. Direct policy manipulation outside the QoS Management interfaces is not possible while QoS Management is enabled.

The remainder of this lesson focuses on user-based policy management as a new feature of Oracle Clusterware 12c. QoS Management is outside the scope of this course.

## Viewing the Policy Set

```
$ crsctl status policyset
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r-x
LAST_ACTIVATED_POLICY=
SERVER_POOL_NAMES=Free
POLICY
  NAME=Current
  DESCRIPTION=This policy is built-in and managed automatically to
reflect current configuration
  SERVERPOOL
    NAME=Free
    ACTIVE_SERVERS=c00n01 c00n02
    EXCLUSIVE_POOLS=
    IMPORTANCE=0
    MAX_SIZE=-1
    MIN_SIZE=0
    PARENT_POOLS=
    SERVER_CATEGORY=
    SERVER_NAMES=
...
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A policy set is implicitly created in every cluster. Because there is exactly one policy set per cluster, policy sets cannot be created or deleted and no name is required to identify the policy set.

The `crsctl status policyset` command can be used to view the current policy set attributes, including all policies and server pools that are defined on the cluster. The output also includes information about the current state of the cluster, which is listed under the special built-in policy named `Current`.

# Configuring a User-Defined Policy Set: Method 1

1. Set the SERVER\_POOL\_NAMES policy set attribute:

```
$ crsctl modify policyset -attr "SERVER_POOL_NAMES='dev test'"
```

2. Add the policies:

```
$ crsctl add policy day -attr "DESCRIPTION='The day policy'"  
$ crsctl add policy night -attr "DESCRIPTION='The night policy'"
```

3. Set the server pool attributes for each policy:

```
$ crsctl modify serverpool dev -attr  
"IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=1,SERVER_CATEGORY=small" -policy day  
  
$ crsctl modify serverpool test -attr  
"IMPORTANCE=5,MAX_SIZE=2,MIN_SIZE=1" -policy day  
  
$ crsctl modify serverpool dev -attr  
"IMPORTANCE=5,MAX_SIZE=2,MIN_SIZE=0,SERVER_CATEGORY=small" -policy night  
  
$ crsctl modify serverpool test -attr  
"IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=2" -policy night
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The policy set can be configured by using the `crsctl` command-line utility. The slide outlines one method and shows a series of example commands. The procedure is:

1. Set the SERVER\_POOL\_NAMES policy set attribute. This attribute formally defines the scope of the server pools that are controlled by the policy set. In addition, any server pool named in the SERVER\_POOL\_NAMES policy set attribute is implicitly created if it did not previously exist.
2. Add the policies. Each policy that is created in this phase is automatically created with a default set of attributes describing each of the server pools named in the previous step.
3. Set the attributes for the server pools in each policy.

## Configuring a User-Defined Policy Set: Method 2

### 1. Create a policy set definition file:

```
$ cat policyset.txt
SERVER_POOL_NAMES=dev test
POLICY
  NAME=day
  DESCRIPTION=The day policy
  SERVERPOOL
    NAME=dev
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=1
    SERVER_CATEGORY=small
  SERVERPOOL
    NAME=test
    IMPORTANCE=5
    MAX_SIZE=2
    MIN_SIZE=1
```

```
POLICY
  NAME=night
  DESCRIPTION=The night policy
  SERVERPOOL
    NAME=dev
    IMPORTANCE=5
    MAX_SIZE=2
    MIN_SIZE=0
    SERVER_CATEGORY=small
  SERVERPOOL
    NAME=test
    IMPORTANCE=10
    MAX_SIZE=2
    MIN_SIZE=2
```

### 2. Modify the policy set:

```
$ crsctl modify policyset -file policyset.txt
```

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The example in this slide shows another way of configuring the policy set. It yields the same result as the commands for method 1.

This method of policy set configuration uses a text file that contains the policy set attribute definitions that are to be implemented. You can use the text file shown in this example as a template for configuring your own policy sets. In this example, the policy set contains two policies and two server pools; however, any number of policies and server pools can be specified this way. After the policy set definition file is created, the policy set can be modified by using the `crsctl modify policyset` command shown in the slide.

Administrators can also use the `crsctl status policyset` command with the `-file` option to dump the current policy set definition into a text file. The resulting text file can then be edited and loaded back into the system by using the `crsctl modify policyset` command shown in the slide. This method is an effective way to configure the policy set when administrators start with existing server pool definitions that were created with previous Clusterware releases.

# Modifying a User-Defined Policy Set

- Method 1
  - Modify the policy set directly by using `crsctl` commands.
  - Examples:

```
$ crsctl add policy day -attr "DESCRIPTION='The day policy'"  
$ crsctl modify serverpool dev  
-attr "IMPORTANCE=10,MAX_SIZE=2,MIN_SIZE=1,SERVER_CATEGORY=small"  
-policy day
```

- Method 2
  1. Create a policy set definition file:
  2. Edit the policy set definition file.
  3. Modify the policy set:

```
$ crsctl status policyset -file policyset.txt
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are essentially two ways to modify a user-defined policy set:

1. Administrators can directly modify the policy set by using specific `crsctl` commands. Examples of direct manipulation are listed in the slide.  
By default, attempts to modify policies and server pools fail if the modification causes a cluster-managed resource, such as a database instance, to shut down. Alternatively, the `-f` command line option can be added to force the change. In addition, what-if command evaluation (described later) can be used to test the effect of a change prior to implementation.
2. Administrators can use the `crsctl status policyset` command with the `-file` option to dump the current policy set definition into a text file. The resulting text file can then be modified to define an updated policy set. Finally, the updated policy set can be loaded into the system by using the `crsctl modify policyset` command with the `-file` option.

This method also provides an effective way to configure the policy set when administrators start with existing server pool definitions that were created with Oracle Clusterware, release 11.2.

# Activating a User-Defined Policy

- Set the LAST\_ACTIVATED\_POLICY policy set attribute:

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY='day'"
```

- Verify the policy settings:

```
$ crsctl status policyset
...
LAST_ACTIVATED_POLICY=day
SERVER_POOL_NAMES=dev test Free
POLICY
  NAME=Current
...
  SERVERPOOL
    NAME=dev
    ACTIVE_SERVERS=c00n01
...
  SERVERPOOL
    NAME=test
    ACTIVE_SERVERS=c00n02
...

```

```
$ crsctl status policy -active
POLICY
  NAME=Current
...
  SERVERPOOL
    NAME=dev
    ACTIVE_SERVERS=c00n01
...
  SERVERPOOL
    NAME=test
    ACTIVE_SERVERS=c00n02
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After the policy set is initially configured, none of the defined policies is active. To activate a policy, the LAST\_ACTIVATED\_POLICY policy set attribute must be set. An example is shown at the top of the slide. The active policy can be changed at will by using this command. Alternatively, system administrators can use a job scheduling system or other management tools to automatically activate different policies based on different times of day or other circumstances.

When a new policy is activated, nodes are automatically reassigned to server pools, and relevant resources are automatically started or stopped in line with the new active policy.

You can examine the active policy by using the crsctl status commands shown in the slide. Examine the LAST\_ACTIVATED\_POLICY policy set attribute, and also check the server assignments in each server pool along with other server pool attributes to verify the policy settings.

## Quiz

Identify the correct statements regarding server categorization:

- a. Server categorization provides a mechanism to control which servers can be in a server pool.
- b. A server category must contain one or more servers.
- c. A server can belong to only one server category at a time.
- d. Servers can be categorized with user-defined expressions that combine various server attributes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: a, d

A server category can contain no servers if none meets the criteria for entry into the category. Servers can belong to multiple categories simultaneously; however, they can belong to only one server pool at a time.

## Quiz

Identify the correct statements regarding policy-based cluster management:

- a. Administrators can create multiple different policy sets to cater for different workloads and priorities.
- b. Administrators can create multiple different policies to cater for different workloads and priorities.
- c. The policy set automatically manages all server pools defined in the cluster.
- d. The policy set automatically manages the server pools identified in the SERVER\_POOL\_NAMES attribute, and policy changes have no effect on other server pools.
- e. The policy set automatically manages the server pools identified in the SERVER\_POOL\_NAMES attribute, and policy changes may indirectly affect other server pools.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b, e

There is always exactly one policy set defined in the cluster, and the policy set can contain zero or more policy definitions. The policy set automatically manages the server pools identified in the SERVER\_POOL\_NAMES attribute, and policy changes may indirectly affect other server pools.

## Quiz

Fill in the blank: The Current policy \_\_\_\_\_ matches the policy definition specified in the LAST\_ACTIVATED\_POLICY policy set attribute.

- a. sometimes
- b. always
- c. never



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: c

The Current policy never matches the policy definition specified in the LAST\_ACTIVATED\_POLICY policy set attribute because it includes the Generic built-in server pool, which is never under policy set control. In addition, the Current policy contains the current state of server pools not under policy set control. It is fair to say that the attributes of the LAST\_ACTIVATED\_POLICY may be reflected in the Current policy; however, those attributes may also vary over time because of changes made outside the policy set.

## Summary

In this lesson, you should have learned how to:

- Describe the architecture and components of policy-based cluster management
- Administer server categorization
- Administer a policy set
- Activate a policy



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 3 Overview: Configuring and Using Policy-Based Cluster Management

In this practice, you will configure server categories and the policy set. You will also examine the effect of various changes to verify the dynamic nature of policy-based cluster management. Finally, you will examine how easy it is to activate policies.



Copyright © 2013, 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

## What-If Command Evaluation

4

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the scope and capabilities of what-if command evaluation
- Perform different types of what-if command evaluation



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## What-If Command Evaluation

Oracle Clusterware 12c provides a set of commands to preview a cluster management operation.

- You can analyze the impact before performing an operation.
- This facilitates the smooth operation of the cluster, with no surprises.
- Supported events:
  - Resource Start
  - Resource Stop
  - Resource Relocate
  - Resource Modify
  - Resource Add
  - Resource Failure
  - Server Pool Addition
  - Server Pool Removal
  - Server Pool Modification
  - Server Addition
  - Server Relocate
  - Server Removal
  - Set Active Policy



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware 12c provides a set of commands to determine the impact of a cluster management operation before the operation is actually executed. This capability is known as what-if command evaluation. It helps administrators to smoothly maintain the cluster and minimizes the potential for surprises. The slide lists the events supported by what-if command evaluation.

What-if command evaluation is supported using the Clusterware C API, the `crsctl eval` command, and the `srvctl` command with the `-eval` option.

The following slides provide further details on performing what-if command evaluation by using the `crsctl` and `srvctl` commands. For further details regarding the Clusterware API, see the API header file at `Grid_home/crs/demo/clscrsx.h`.

# Performing What-If Command Evaluation on Application Resources with CRSCTL

- Commands for application administrators:

```
$ crsctl eval { start | stop | relocate | modify | add | fail } resource  
...
```

- Example:

```
$ crsctl eval start resource my_resource -n my_server  
Stage Group 1:  
-----  
Stage Number Required Action  
-----  
1 Y Resource 'my_dep_res1' (1/1) will be in state  
[ONLINE] on server [my_server]  
N Resource 'my_dep_res2' (1/1) will be in state  
[ONLINE|INTERMEDIATE] on server [my_server]  
2 Y Resource 'my_resource' (1/1) will be in state  
[ONLINE|INTERMEDIATE] on server [my_server]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using the `crsctl eval` command, application administrators can perform what-if command evaluation to test the effect of starting, stopping, relocating, modifying, or adding cluster resources. Administrators can also examine the effect of a potential resource failure. The slide outlines the available commands. For more information, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.

What-if command evaluation using the `crsctl eval ... resource` command, as shown in this slide, is recommended for use only in conjunction with user-defined application resources. For Oracle Clusterware resources (resources with the `ora.` name prefix), you should use the `srvctl predict` command or the `srvctl ... -eval` command.

The bottom of the slide contains example output for a what-if scenario showing the effect of starting `my_resource` on `my_server`. In this example, the application administrator can clearly see that starting `my_resource` on `my_server` requires `my_dep_res1` to be started first. In addition, before starting `my_resource`, Oracle Clusterware attempts to start `my_dep_res2`; however, the output notes that `my_dep_res2` is not a mandatory dependent resource.

# Performing What-If Command Evaluation on Oracle Clusterware Resources with CRSCTL

- Commands for cluster administrators:

```
$ crsctl eval { add | delete | modify } serverpool ...
$ crsctl eval { add | relocate | delete } server ...
$ crsctl eval activate policy ...
```

- Example:

```
$ crsctl eval delete server my_server -f
Stage Group 1:
-----
Stage Number    Required      Action
-----
1              Y             Server 'some_server' will be moved from pools
                           [Free] to pools [ora.my_pool]
                           Server 'my_server' will be removed from pools
                           [ora.my_pool]
...
6              Y             Resource 'my_resource' (1/1) will be in state
                           [ONLINE] on server [some_server]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Cluster administrators can use the `crsctl eval` command to perform what-if command evaluation that tests the effect of:

- Adding, deleting, and modifying server pools
- Adding servers to and deleting servers from a server pool
- Relocating a server from one server pool to another
- Removing a server from the cluster
- Enabling a specific management policy

The slide outlines the available commands. For more information, refer to the *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*.

The example in the slide contains partial output for a what-if scenario showing the effect of removing `my_server` from the cluster. In this example, removing `my_server` causes `some_server` to be moved into the `my_pool` server pool. After the server pool reallocation, the required resources, including `my_resource` and its dependent resources, are started on `some_server`.

# Formatting the Output for What-If Command Evaluation on Oracle Clusterware Resources

Commands for cluster administrators may contain additional parameters to govern the output format.

- Command syntax:

```
$ crsctl eval ... serverpool ... [-admin [-l <level>] [-x] [-a]]
```

```
$ crsctl eval ... server ... [-admin [-l <level>] [-x] [-a]]
```

```
$ crsctl eval activate policy ... [-admin [-l <level>] [-x] [-a]]
```

- Example:

```
$ crsctl eval delete server my_server -f -admin

NAME = Free
ACTIVE_SERVERS  =

NAME = Generic
ACTIVE_SERVERS  =

NAME = ora.my_pool
ACTIVE_SERVERS  = some_server
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `crsctl eval` commands for cluster administrators may contain additional parameters to govern the format of the command output. Their aim is to provide cluster administrators with the ability to control the amount of information returned by the commands.

The slide contains an example using the `-admin` option by itself. Rather than showing output describing all of the effects on servers and resources (like the example on the previous page), the `-admin` option modifies the output so that the administrator is provided with a summary of the server pool assignments resulting from the proposed action.

The following is a brief description of the additional formatting parameters that administrators can use in conjunction with the `-admin` option:

`-l <level>` specifies the output display level:

- l serverpools displays server pool information.

- l resources displays resource information.

- l all displays server pool and resource information.

`-x` shows differences only.

`-a` shows all resources.

# Performing What-If Command Evaluation with SRVCTL

- Commands:

```
$ srvctl { add | start | stop | modify | relocate } database ... -eval  
$ srvctl { add | start | stop | modify | relocate } service ... -eval  
$ srvctl { add | modify | remove } svrpool ... -eval  
$ srvctl relocate server ... -eval
```

- Example:

```
$ srvctl start database -db orcl -eval  
Resource ora.asm will be started on node c00n02  
Resource ora.DATA.dg will be started on node c00n02  
Resource ora.FRA.dg will be started on node c00n02  
Database orcl will be started on node c00n02
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In addition to the `crsctl eval` commands that perform what-if command evaluation, administrators can use the `srvctl` command with the `-eval` option to perform what-if command evaluation that tests the effect of:

- Adding, starting, stopping, modifying, and relocating databases
- Adding, starting, stopping, modifying, and relocating services
- Adding, modifying, and removing server pools
- Relocating a server from one server pool to another

The slide outlines the available commands. For more information, refer to the *Oracle Real Application Clusters Administration and Deployment Guide 12c Release 1 (12.1)*.

The example in the slide contains output for a what-if scenario showing the effect of starting a database. In this example, the database administrator can clearly see that starting the database also causes ASM and the required disk group resources to start.

# Evaluating Failure Consequences with SRVCTL

- Command:

```
$ srvctl predict { database | service | asm | diskgroup | filesystem |
    vip | network | listener | scan | scan_listener |
    oc4j } ... [-verbose]
```

- Examples:

```
$ srvctl predict asm -n c00n02
```

```
$ srvctl predict diskgroup -g DATA
```

```
$ srvctl predict filesystem -d /dev/asm/voll-261 -verbose
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `srvctl predict` command allows administrators to evaluate the consequences of a failure affecting any of the following types of resources:

- Database
- Service
- ASM
- Diskgroup
- Filesystem
- VIP
- Network
- Listener
- SCAN
- SCAN Listener
- OC4J

For more information regarding the `srvctl predict` command options, refer to the *Oracle Real Application Clusters Administration and Deployment Guide 12c Release 1 (12.1)*.

## Quiz

Which command is recommended to evaluate the effect of ASM failure on a server?

- a. crsctl eval fail resource "ora.<node>.ASM<n>.asm"
- b. srvctl predict asm -n <node>
- c. Either of the above



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b

What-if command evaluation using the `crsctl eval ... resource` command is recommended only for use in conjunction with user-defined application resources. For Oracle Clusterware resources (resources with the `ora.` name prefix), you should use the `srvctl predict` command or the `srvctl ... -eval` command.

## Quiz

Which command provides a concise summary of server assignments resulting from the removal of a server?

- a. crsctl eval delete server <server> -l serverpools
- b. crsctl eval delete server <server> -a
- c. crsctl eval delete server <server> -f -admin



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: c

The -admin option shows a summary of the server assignments resulting from the action specified in the crsctl eval commands for cluster administrators.

## Summary

In this lesson, you should have learned how to:

- Describe the scope and capabilities of what-if command evaluation
- Perform different types of what-if command evaluation



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 4 Overview: Using What-If Command Evaluation

In this practice, you will perform what-if command evaluation using the different commands for various different resources. You will also examine the different output formatting options available with what-if command evaluation.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Other Clusterware New Features



ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the other Clusterware new features in release 12.1
- Perform essential configuration and management associated with these features



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Shared GNS: Overview

In previous releases:

- GNS must be configured separately on each cluster that uses GNS.
- Each GNS requires a separate GNS VIP address, a separate domain (or subdomain), and a separate configuration for DNS forwarding.
- Multiple GNS environments consume more machine resources.

With Oracle Clusterware 12c, one GNS can support multiple Oracle clusters:

- Only one GNS VIP address, one domain, and one DNS configuration is required across the enterprise.
- GNS consumes fewer machine resources.

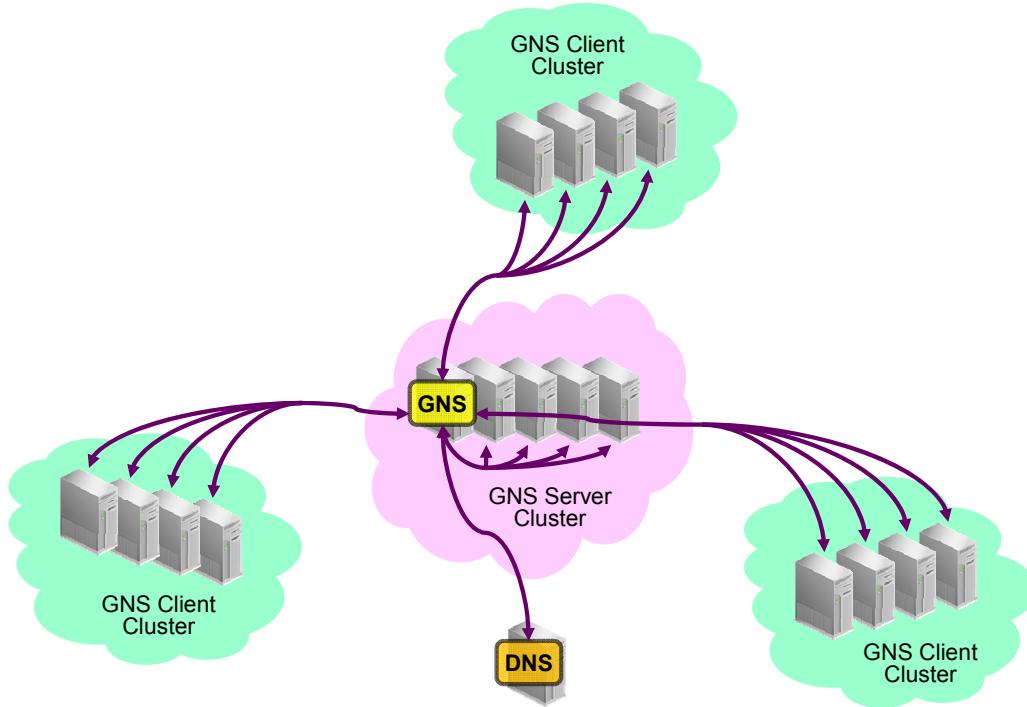


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Grid Naming Service (GNS) was introduced to simplify network management for Oracle clusters. The main aim was to make Oracle clusters self-contained from a network management perspective, so that changes in the cluster did not require network administrator involvement to manage hostnames and IP addresses. Previous releases of Oracle Clusterware required a separate GNS configuration for every Oracle cluster. For customers with many clusters, this meant many interactions with the network administrator and many copies of essentially the same service consuming machine resources throughout the enterprise.

With Oracle Clusterware 12c, one GNS can support multiple clusters. This feature is called shared GNS. With shared GNS, the network administrator is required only during the initial configuration of GNS and not when other clusters are added. In addition, GNS consumes fewer machine resources across the entire network.

# Shared GNS Architecture



ORACLE®

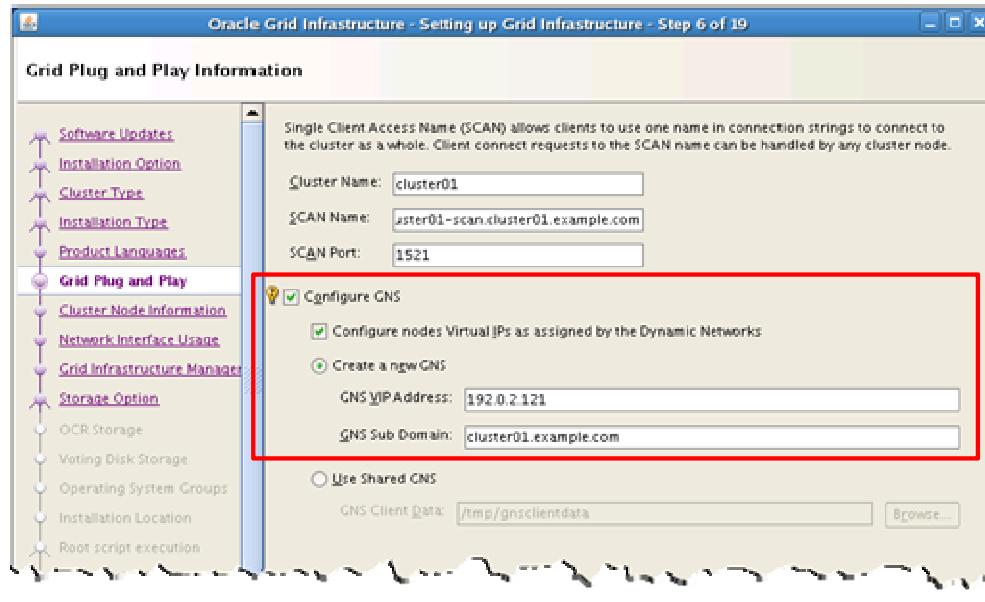
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Shared GNS is essentially the same as single-cluster GNS, except that the server maintains host information for remote (client) clusters and services queries from them. The configuration process for shared GNS on the GNS server cluster is the same as the configuration process for single-cluster GNS in previous versions. GNS still requires a GNS VIP address, a cluster domain (or subdomain), and the configuration of DNS forwarding so that DNS forwards all requests for resolving cluster hostnames to GNS.

Shared GNS relies on at least one node being available at all times in the GNS server cluster. If the entire GNS server cluster is down, shared GNS cannot function. If the GNS server cluster cannot be recovered, GNS can be moved to another cluster by exporting information about the GNS instance to a file and importing it on another cluster. If the OCR on the GNS server cluster is inaccessible, exporting the GNS data is not possible. In that case, a new GNS server can be configured on another cluster and the clients need to be reconfigured.

# Configuring a GNS Server Cluster

- The same process as for single-cluster GNS
- Typically performed during installation of the cluster



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

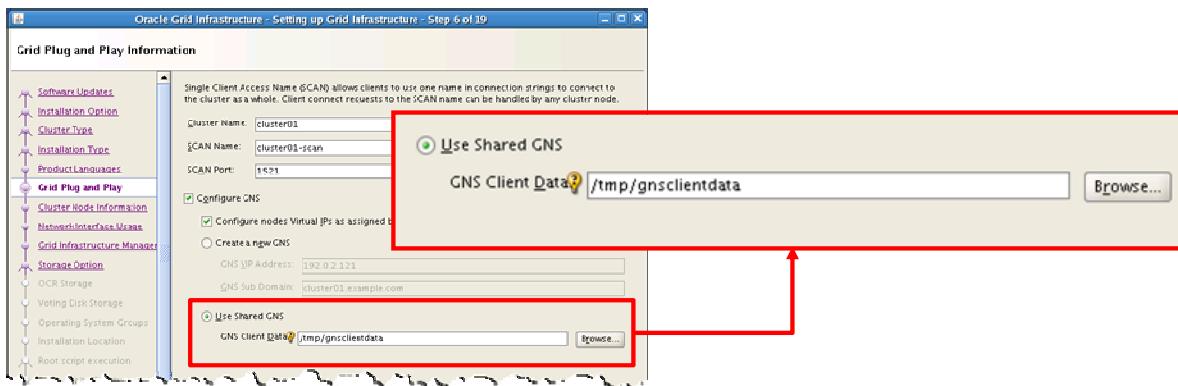
The procedure for configuring GNS on the server cluster remains unchanged from previous releases, and in most cases, GNS configuration occurs during the cluster installation and configuration process. The slide shows a GNS server cluster configuration example using Oracle Universal Installer (OUI).

# Configuring a GNS Client Cluster

1. Export GNS credentials on the GNS server cluster:

```
# srvctl export gns -clientdata /tmp/gnsclientdata
```

2. Copy the credentials file to the client cluster.
3. Select the Use Shared GNS option and reference the credentials file in the Oracle Universal Installer.



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The configuration process for a client cluster involves registering the client cluster with the GNS server. To perform the registration, a set of GNS server credentials is required. The overall procedure for configuring a GNS client cluster is as follows:

1. Export the GNS credentials on the server cluster by using the `srvctl export gns` command and specifying a file to store the GNS credentials.
2. Copy the credentials file to the client cluster by using a network file copy (`ftp` or `scp`, for example) or a physical file transfer (CD, DVD, or memory stick, for example).
3. On the Oracle Universal Installer “Grid Plug and Play Information” screen, select Use Shared GNS and specify the credentials file in the GNS Client Data field.

## Migrating to Shared GNS

- Configuring GNS on an existing non-GNS cluster:
  - Collaborate with the network administrator to configure the network.
  - Add GNS to the cluster:

```
# srvctl add gns -i <GNS VIP address> -d <cluster domain>
```
- Converting single-cluster GNS to a GNS server cluster:
  - No specific action is required.
  - GNS server can be shared after Clusterware is upgraded to 12c.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring GNS on an existing non-GNS cluster involves the same fundamental process as configuring GNS on a brand new cluster. In both cases, collaboration with the network administrator is required to configure the network to support GNS. This involves allocating a GNS VIP address, allocating a domain (or subdomain) in the network for the cluster, and configuring DNS forwarding so that DNS forwards all requests for resolving cluster hostnames to GNS. After the network requirements are met, GNS can be added to the cluster by using the `srvctl add gns` command and specifying the GNS VIP address and the cluster domain name.

You do not need to do anything to convert a single cluster running GNS to a server cluster. After a cluster is upgraded to 12c, the existing GNS is inherently capable of acting as a shared GNS server and is immediately ready to accept client cluster registrations.

## Migrating to Shared GNS

- Converting an existing cluster with GNS to a client cluster:
  1. Export GNS credentials on the GNS server cluster.
  2. Copy the credentials file to the client cluster.
  3. Stop and remove GNS on the client cluster:

```
# srvctl stop gns  
# srvctl remove gns
```

4. Register the client cluster:

```
# srvctl add gns -clientdata /tmp/gnsclientdata
```

5. Restart Clusterware on the client cluster.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To convert an existing GNS server cluster to a client cluster you must first nominate another cluster as the GNS server cluster. Then you export the GNS credentials for the server cluster as shown on the previous page. After transporting the credentials file to the client cluster, you must stop and remove the existing GNS on the client cluster. Then you can register the cluster as a client cluster. Finally, you must restart Clusterware on the client cluster nodes so that the Clusterware services can re-advertise themselves to the GNS server cluster.

Note that the cluster being converted into a client cluster must be a standard cluster running Oracle Clusterware release 12.1 or later at the time of conversion.

## Shared GNS Naming

- Separate GNS servers example:

	GNS Server Cluster	GNS Server Cluster
Cluster Name	cluster01	cluster02
Nodes	node01, node02	node01, node02
GNS Subdomain	gns1.example.com	gns2.example.com
SCAN	cluster01-scan.gns1.example.com	cluster02-scan.gns2.example.com
Node VIPs	node01-vip.gns1.example.com node02-vip.gns1.example.com	node01-vip.gns2.example.com node02-vip.gns2.example.com

- Shared GNS example:

	GNS Server Cluster	GNS Client Cluster
Cluster Name	cluster01	cluster02
Nodes	node01, node02	node01, node02
GNS Subdomain	gns1.example.com	cluster02.gns1.example.com
SCAN	cluster01-scan.gns1.example.com	cluster02-scan.cluster02.gns1.example.com
Node VIPs	node01-vip.gns1.example.com node02-vip.gns1.example.com	node01-vip.cluster02.gns1.example.com node02-vip.cluster02.gns1.example.com

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Shared GNS, the fully qualified hostnames associated with the SCAN and node VIPs change when compared with running separate GNS servers on each cluster.

With separate GNS clusters, each cluster is associated with a different GNS subdomain. Therefore, the fully qualified SCAN typically takes the following form:

<Cluster Name>-scan.<GNS Subdomain>

And each fully qualified node VIP typically takes the following form:

<hostname>-vip.<GNS Subdomain>

The table in the top half of the slide outlines an example showing two clusters with separate GNS servers. Note how the SCAN and node VIPs conform to the formats listed above.

The table in the bottom half of the slide outlines an example showing the same two clusters. However, this time cluster02 is configured as a GNS client cluster. In this case, note that the client cluster GNS subdomain is a concatenation of the cluster name and the server cluster GNS subdomain. This ensures that nodes in different clusters can be differentiated even if their hostnames are not unique.

Therefore, for GNS client clusters, the fully qualified SCAN typically takes the following form:

<Cluster Name>-scan.<Cluster Name>.<GNS Server Subdomain>

And each fully qualified node VIP typically takes the following form:

<hostname>-vip.<Cluster Name>.<GNS Server Subdomain>

# Moving GNS to Another Cluster

1. Stop GNS on the old (current) GNS server cluster:

```
# srvctl stop gns
```

2. Export the GNS instance data from the old server cluster:

```
# srvctl export gns -instance <filename>
```

3. Copy the GNS instance data file to the new server cluster.

4. Import the GNS instance data to the new server cluster:

```
# srvctl import gns -instance <filename>
```

5. Start GNS on the new server cluster:

```
# srvctl start gns
```

6. Remove GNS on the old server cluster:

```
# srvctl remove gns
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide outlines the procedure for moving GNS to another cluster. Using this procedure, GNS may be moved back and forth between different clusters with minimal disruption. This may be required under various circumstances, including situations where the GNS server cluster is being prepared for maintenance and might be unavailable for an extended period.

This procedure relies on the ability to export the existing GNS instance data. If this is impossible, due to cluster failure or irrecoverable loss of the OCR, two options exist:

1. If an up-to-date copy of the GNS instance data exists, it can be imported into a new GNS server cluster and GNS can be started on the new sever cluster.
2. If an up-to-date copy of the GNS instance data does not exist, a new GNS server can be configured on a new server cluster and the client clusters must be reconfigured to access the new server cluster.

## Quiz

Identify the correct statements regarding shared GNS:

- a. Shared GNS consumes fewer machine resources when implementing GNS on multiple clusters.
- b. Shared GNS requires less configuration involving the network administrator when implementing GNS on multiple clusters.
- c. A release 12.1 GNS server is inherently capable of servicing GNS client clusters.
- d. A GNS server cluster can easily be converted to a GNS client cluster.
- e. All of the above.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: e**

## Cluster Health Monitor Enhancements: Overview

With Grid Infrastructure 12c, CHM is enhanced in the following ways:

- The CHM architecture supports the Hub Node and Leaf Node architecture of Flex Clusters.
  - This enables CHM to scale as cluster size increases.
- The CHM repository resides inside an Oracle database known as the Grid Infrastructure Management Repository.
  - This enables:
    - The CHM repository to scale as the cluster size increases
    - Integration of CHM data into Oracle Enterprise Manager
    - Better analysis of CHM data



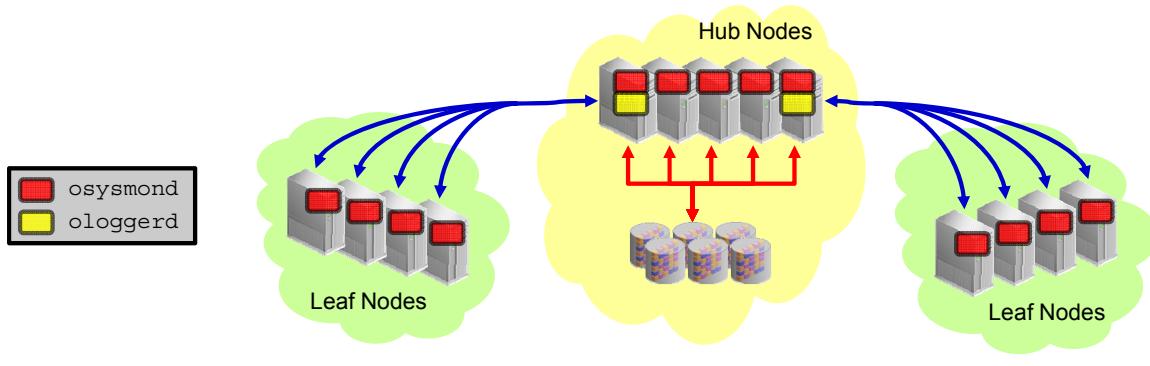
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Grid Infrastructure 12c, Cluster Health Monitor (CHM) is enhanced in two fundamental ways.

1. The CHM architecture is enhanced to support the Hub Node and Leaf Node architecture of Flex Clusters. This enables CHM to perform effectively with clusters containing hundreds, and perhaps thousands, of nodes.
2. The CHM repository, which previously resided in a file-based database, is implemented inside a dedicated Oracle database known as the Grid Infrastructure Management Repository. This enables the CHM repository to scale in size beyond previous limits and support longer retention periods for CHM data. In addition, moving the CHM repository into an Oracle database enables integration of CHM data with Oracle Enterprise Manager, which in turn allows CHM data to be correlated with other metrics inside Enterprise Manager.

# Cluster Health Monitor Services

- The system monitor service (`osysmond`) collects OS metrics and sends them to the cluster logger service.
- The cluster logger service (`ologgerd`) receives metrics from `osysmond` and persists them to the Grid Infrastructure Management Repository.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The CHM system monitor service (`osysmond`) resides on each cluster node and collects the OS metrics used by CHM. With 12c, `osysmond` is essentially the same as in previous versions.

The cluster logger service (`ologgerd`) receives metrics from `osysmond` and persists them to the Grid Infrastructure Management Repository. The communication between `ologgerd`, `osysmond`, and the Grid Infrastructure Management Repository is performed using the private network (cluster interconnect).

In previous versions, there was one master and one standby cluster logger service on each cluster. With Flex Clusters, `ologgerd` may be required to receive metrics from thousands of nodes. Therefore, when a Flex Cluster is configured, each Leaf Node communicates metric information to an `ologgerd` process running on the Hub Node that connects it to the cluster.

# Grid Infrastructure Management Repository

The Grid Infrastructure Management Repository is a special Oracle database that:

- Stores CHM metrics and underpins other features, such as QoS Management
- Can only be configured during installation of, or upgrade to, Oracle Clusterware, release 12.1
- Must run on one Hub Node
  - The instance fails over to another Hub Node in case of node or storage failure.
- Uses the same shared storage as OCR and the voting disk
  - Can reside on an NFS, a cluster file system, or an Oracle ASM disk group



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Grid Infrastructure Management Repository is a special Oracle database that:

- Stores real-time operating system metrics collected by CHM and also facilitates other features, such as QoS Management
- Is configured during an installation of, or upgrade to, Oracle Clusterware, release 12.1. If you elect not to configure the Grid Infrastructure Management Repository, the features that rely on it cannot be used, and the cluster must be reinstalled to add the repository and enable the associated features.
- Must run on one Hub Node in the cluster, and must support failover to another node in case of node or storage failure. You can locate the Grid Infrastructure Management Repository on the same node as `ologgerd` to improve performance and decrease private network traffic.
- Uses the same shared storage as OCR and the voting disk. It can be an NFS mount, a cluster file system, or an Oracle ASM disk group.

**Note:** If you are upgrading to release 12.1 (with Grid Infrastructure Management Repository), and the OCR and voting disk are stored on raw or block devices, you must move them to one of the supported shared storage options (NFS, CFS or ASM) before you upgrade your software and implement the Grid Infrastructure Management Repository.

# Managing Cluster Health Monitor: Routine Monitoring

- Check the Grid Infrastructure Management Repository:

```
$ srvctl status mgmtlsnr
Listener MGMTLSNR is enabled
Listener MGMTLSNR is running on node(s): c00n01
$ srvctl status mgmtdb
Database is enabled
Instance -MGMTDB is running on node c00n01
```

- osysmond logs data locally if the repository is unavailable.

- Check that the CHM services are running on each node:

```
$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

- The High Availability Services check fails if osysmond is down.
  - Because ologgerd is managed by osysmond, no additional checks are required.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Cluster Health Monitor is designed to be predominately automatic and self-contained, and does not require substantial management effort. As part of routine system monitoring, administrators can check that the Grid Infrastructure Management Repository and the CHM services are up and running.

If the CHM services detect that the Grid Infrastructure Management Repository is unavailable, the CHM system monitor service (osysmond) will record observations in a local file-based data store on each cluster node. When the Grid Infrastructure Management Repository becomes available again, entries in the local data store are uploaded to it. The local data store is located at `GRID_HOME/crf/db/<hostname>/<hostname>.1db` and is fixed to approximately 100 MB in size. The local data store is typically capable of storing between 12 hours and 24 hours of CHM data, depending on system load. If the local data store fills up, new observations overwrite the oldest ones in the store, resulting in an unavoidable loss of CHM data.

## Managing Cluster Health Monitor: New oclumon Commands

- List all nodes running the cluster logger service (ologgerd):

```
$ oclumon manage -get allogger  
Loggers = c00n01,
```

- List all nodes running the cluster logger service along with the nodes serviced by each logger:

```
$ oclumon manage -get allogger -details  
Logger = c00n01  
Nodes = c00n01,c00n02,c00n03,c00n04
```

- List the logger servicing the current node:

```
$ oclumon manage -get mylogger
```

- List the logger servicing the current node along with all the nodes serviced by the logger:

```
$ oclumon manage -get mylogger -details
```

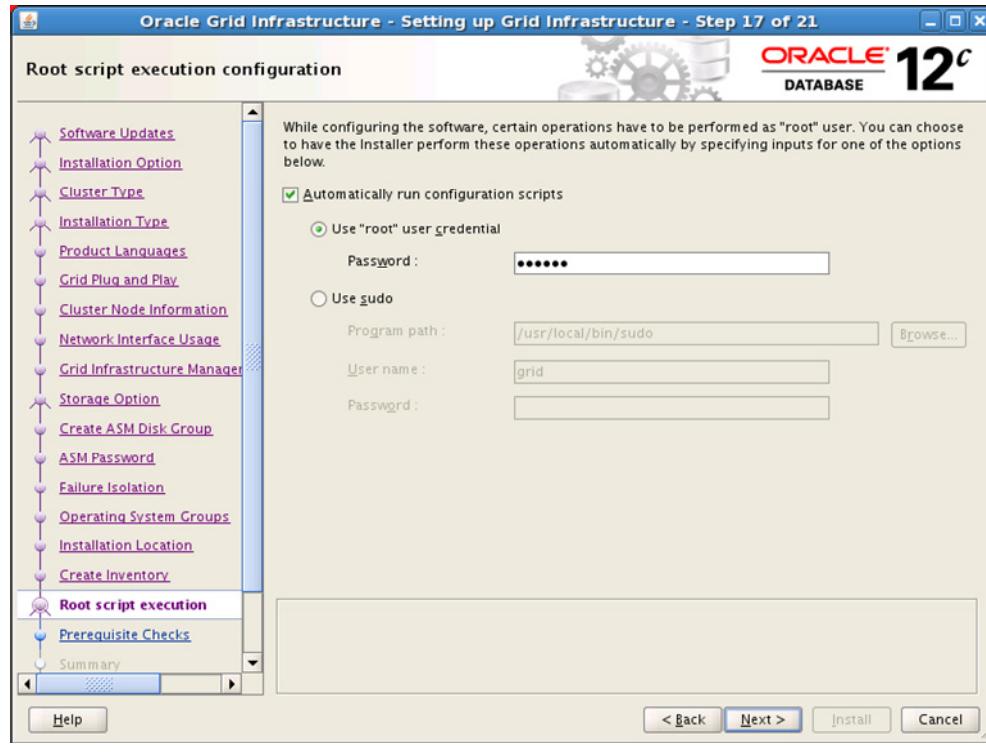


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Clusterware 12c, the oclumon command-line tool has been extended to provide administrators with more information about the cluster logger service. This enables administrators to understand the relationship between the system monitor service running on each node and the cluster logger instance that services it.

The new commands are listed in the slide.

# Grid Infrastructure Script Automation for Installation and Upgrade



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, Oracle Universal Installer paused and displayed a dialog box that instructed users to manually execute special root configuration scripts. Manual execution was required because the root configuration scripts required administrator privileges that were not available to the user who performed the installation.

With the 12c release, the Grid Infrastructure installation and upgrade processes support automatic root script execution by using the Oracle Universal Installer screen shown in the slide. Users have two options:

- They can supply the root password during the installer session. To use this option, the root user password must be the same across all nodes in the cluster. This option is illustrated in the slide.
- They can use a previously configured sudo configuration. This option requires the sudo program to be available on all nodes in the cluster. The user who runs the installer must have sudo privileges, and the password for the install user must be provided. The sudo program may be configured to require the root user password. In that case, the user needs to supply the root password.

## Bundled Agents

- Oracle Clusterware can be used to maintain high availability for business-critical applications.
- In previous releases, developers could use the available APIs and services to enable high availability using failover protection for any application.
- With Oracle Clusterware 12c, bundled agents provide a packaged solution to enable high availability:
  - Release 12.1 includes bundled agents for:
    - Apache
    - Oracle GoldenGate
    - Siebel CRM applications
  - Bundled agents can leverage database services to manage the dependencies between applications and databases.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Clusterware provides the necessary components to manage high availability for business-critical applications.

In previous releases, developers and application administrators could use customizable action scripts and resource attributes, along with the agent framework APIs to monitor application resources and provide automatic failover to ensure high availability.

With Oracle Clusterware 12c, bundled agents provide a packaged solution to enable high availability for selected applications. Release 12.1 includes bundled agents for Apache, Oracle GoldenGate, and Siebel CRM applications.

Using the bundled agents, each application entity is managed as a cluster resource. The bundled agents also include preconfigured resource dependencies, inherent placement policies, and fine-grained resource state reporting. Bundled agents can also reference database services. This enables end-to-end resource dependency modeling for applications and their associated databases.

Bundled agents are part of the standard software distribution for Oracle Clusterware 12c. Bundled agents are also available as an additional package for Oracle Clusterware release 11.2.0.3 and later.

## IPv6 Support for VIPs and Network Agents

In previous versions, support for IPv6 was incomplete:

- Database and Oracle Net Services supported IPv6.
- VIP and network agent supported only IPv4.

Oracle Clusterware12c provides complete support for IPv6:

- VIPs and network agents are enhanced to support IPv6.
  - No dependency on IPv4
- IPv6 and IPv4 can coexist on the same network:
  - A VIP supporting IPv4 and IPv6 is created.
  - The VIP agent monitors both addresses.
  - Oracle Net Listeners set up endpoints using both addresses.
  - Databases accept connections on both endpoints.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Before Oracle Database 12c, support for IPv6 was incomplete. Although support for IPv6 addressing existed in some components since release 11.2, the VIP addresses provided by Oracle Clusterware supported only IPv4, essentially rendering the IPv6 support in other components useless.

Oracle Clusterware 12c completes support for IPv6 addressing across the Oracle Database software stack. In this release, the VIPs and the network agents that monitor them are enhanced to support IPv6 addressing. There is no dependency on IPv4, and therefore, customers can implement Oracle Grid Infrastructure and Database software in a pure IPv6 network.

In most networks, however, IPv6 coexists with IPv4, and Oracle supports such scenarios. Specifically, support is provided for pure IPv4, pure IPv6, and a mixed mode of IPv4 and IPv6 on the same network. The addressing mode supported by the network is automatically detected when a new network resource is defined. If a mixed-mode network is detected, a VIP supporting IPv4 and IPv6 is created and the VIP agent monitors both addresses. Oracle Net Listeners can set up endpoints by using both addresses, and databases can accept connections by using either address.

Note that mixed-mode support is provided only on a single network interface. You cannot use IPv4 on one network interface and IPv6 on another network interface to imply mixed mode. In addition, all nodes must use the same addressing mode. You cannot use IPv4 on some nodes, IPv6 on other nodes, and mixed mode on the rest.

## Quiz

True or False: If you choose not to install the Grid Infrastructure Management Repository at cluster installation time, you can configure it later by using the `crsctl add repository` command.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b

If you elect not to configure the Grid Infrastructure Management Repository, the features that rely on it cannot be used. You must reinstall the cluster to add the repository and enable the associated features. The `crsctl add repository` command does not exist.

## Summary

In this lesson, you should have learned how to:

- Describe the other Clusterware new features in release 12.1
- Perform essential configuration and management associated with these features



Copyright © 2013, 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

## Flex ASM

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

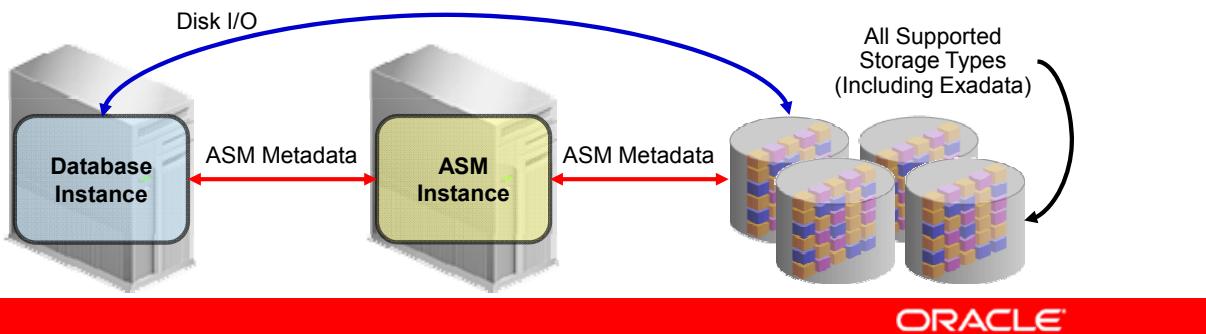
- Describe the architecture and components of Flex ASM
- Install and configure Flex ASM
- Administer Flex ASM



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Flex ASM: Overview

- In previous versions, ASM clients can only access ASM by using an ASM instance on the same host.
  - Resources are consumed by ASM on every database server.
  - If an ASM instance fails, its clients must fail.
- With Flex ASM, ASM clients can use a network connection to access ASM.
  - Resources are saved because ASM is not required on every database server.
  - If an ASM instance fails, its clients can connect to another instance.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Prior to Oracle Database 12c, an ASM client (database instance or ASM Cluster File System [ACFS]) can only connect to an ASM instance running on the same host. This requires every database server to dedicate system resources to ASM, which increases the overall system resource requirement to run Oracle Database in conjunction with ASM. This tightly coupled model also has availability concerns because if an ASM instance fails, all ASM clients on that host must also fail.

Oracle Database 12c introduces Flex ASM. Flex ASM allows ASM clients to connect to ASM over a network. By relaxing the hard dependency between ASM and its clients, the previous architectural limitations are overcome. With Flex ASM, a smaller pool of ASM instances can be used to serve a large pool of database servers. If an ASM instance fails, its clients can reconnect to another ASM instance.

Note that ASM continues to support the same architecture as previous releases where clients are coupled with ASM instances on the same host. This mode of deployment is called standard ASM.

## Flex ASM and Flex Clusters

- Flex Clusters require Flex ASM.
  - Standard ASM is not supported on a Flex Cluster.
- Flex ASM does not require a Flex Cluster.
  - Flex ASM can run on a standard cluster servicing clients across the cluster.
  - Flex ASM can run on the Hub Nodes of a Flex Cluster servicing clients across the Hub Nodes of the Flex Cluster.
- The benefits of Flex ASM apply regardless of cluster type:
  - Smaller ASM resource footprint
  - Protection from ASM failure



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Standard ASM is not supported on a Flex Cluster. Therefore, Flex Clusters require Flex ASM. However, Flex ASM does not require a Flex Cluster. Flex ASM can be configured on either a standard cluster or a Flex Cluster.

When Flex ASM runs on a standard cluster, ASM services can run on a subset of cluster nodes servicing clients across the cluster. When Flex ASM runs on a Flex Cluster, ASM services can run on a subset of Hub Nodes servicing clients across all of the Hub Nodes in the Flex Cluster.

The fundamental benefits of Flex ASM apply regardless of the type of cluster being used. That is:

- The overall resource footprint is smaller because a smaller pool of ASM instances can be used to serve a larger pool of database servers.
- Higher availability can be achieved because if an ASM instance fails, its clients can reconnect to another ASM instance.

## ASM Instance Changes

- ASM Instances are no longer required to run on every node in a cluster.
- Administrators specify the cardinality for ASM.
  - Cardinality sets the number of instances across the cluster.
  - The default is 3.
- All disk groups are mounted by all ASM instances.



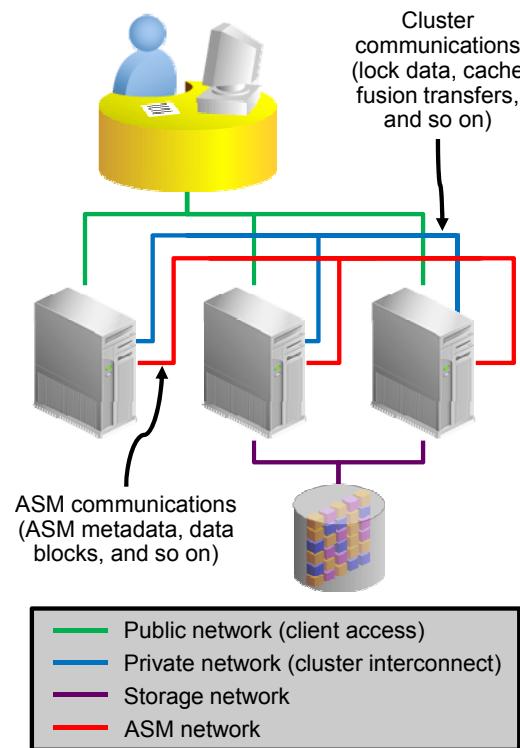
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flex ASM relaxes the requirement for an ASM instance to exist on every node in the cluster. Instead, the administrator specifies the cardinality for ASM. This number specifies the number of ASM instances that should be made available in the ASM cluster. The default cardinality setting for ASM instances is three.

Because an ASM instance can now service any database instance across the cluster, all disk groups are typically mounted on all the ASM instances. As is the case in previous versions, the ASM instance running on a node has its ORACLE\_SID set to +ASM<node number>.

# ASM Network

- In previous versions, a CSS cluster requires:
  - A public network for client application access
  - One or more private networks for inter-node communication within the cluster
- Flex ASM adds the ASM network, which is used for communication between ASM and ASM clients.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

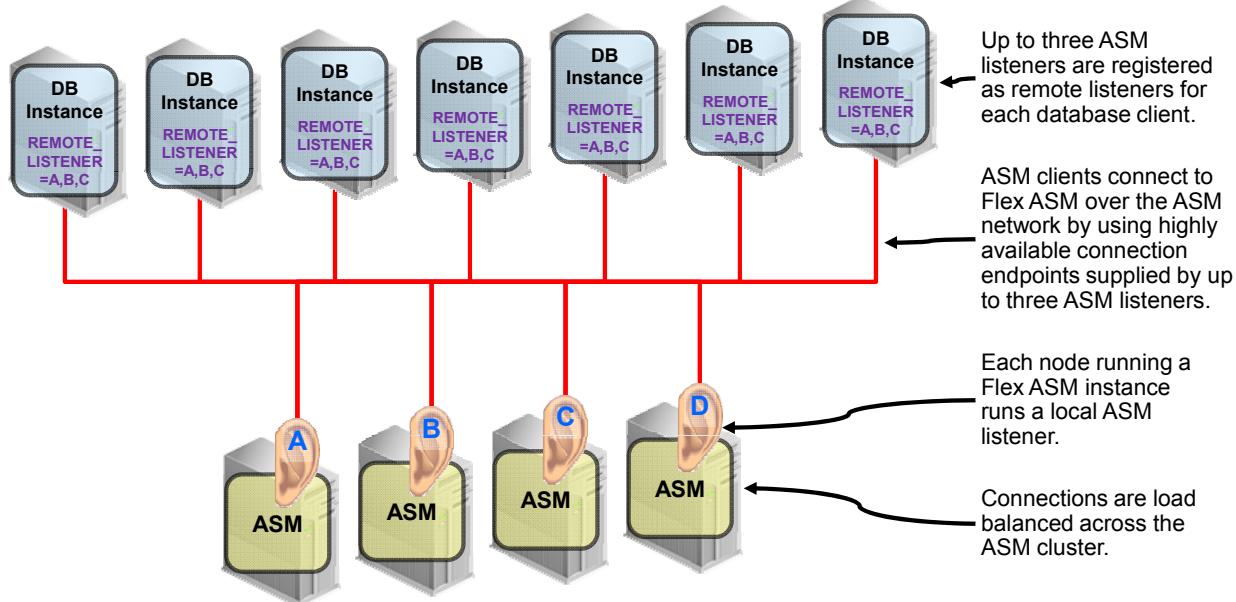
In previous versions, a CSS cluster requires access to a public network and one or more private networks. Clients outside the cluster use the public network to connect to servers in the cluster. The private networks are predominantly used for inter-node communication within the cluster. Sometimes, the private network also serves as the storage network. This is the case inside Oracle Exadata Database Machine.

Flex ASM introduces a new type of network called an *ASM network*. The ASM network is used for all communication between ASM and its clients. There can be more than one ASM network in a customer environment. ASM provides its services on all the ASM networks, and this requires all ASM networks to be accessible on all the nodes hosting ASM instances.

All ASM clients running within the ASM cluster can use any of the ASM networks to communicate with ASM.

It is possible to configure a network as both a private and an ASM network. That is, a single network can perform both functions.

## ASM Listeners



**ORACLE®**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To support Flex ASM connectivity, a set of ASM listeners is configured for every ASM network. The ASM listeners are in addition to other listeners such as the SCAN listeners and the local database listeners. The diagram in the slide illustrates the arrangement of ASM listeners.

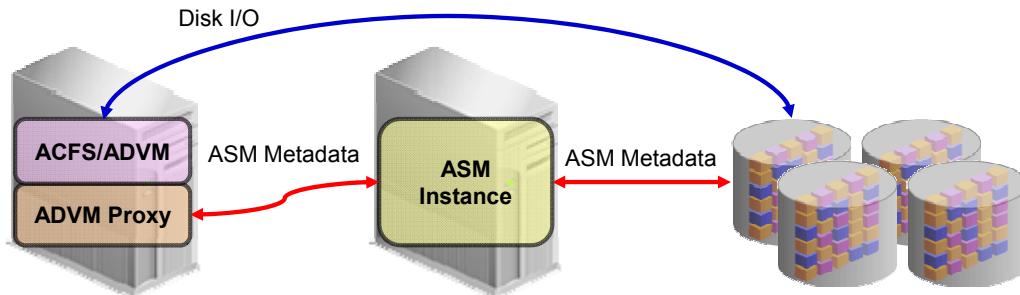
Each node hosting an ASM instance hosts one local ASM listener for each ASM network. Each ASM listener can service client connections over the corresponding ASM network. Up to three ASM listener addresses are registered as remote listeners in each client database instance. Using this arrangement, clients have a highly available connection endpoint to facilitate connection to ASM instances.

While connection is initiated by using one of the registered remote listeners, all client connections are load balanced across the entire set of available ASM instances. The load-balancing mechanism is connect-time load balancing.

## ADVM Proxy

The ADVM Proxy is a special Oracle instance.

- It enables ADVM to connect to Flex ASM.
- It is required to run on the same node as ADVM and ACFS.
- By default, it is configured on every node in a standard cluster or every Hub Node in a Flex Cluster.
- It can be shut down when ACFS is not running.



ORACLE

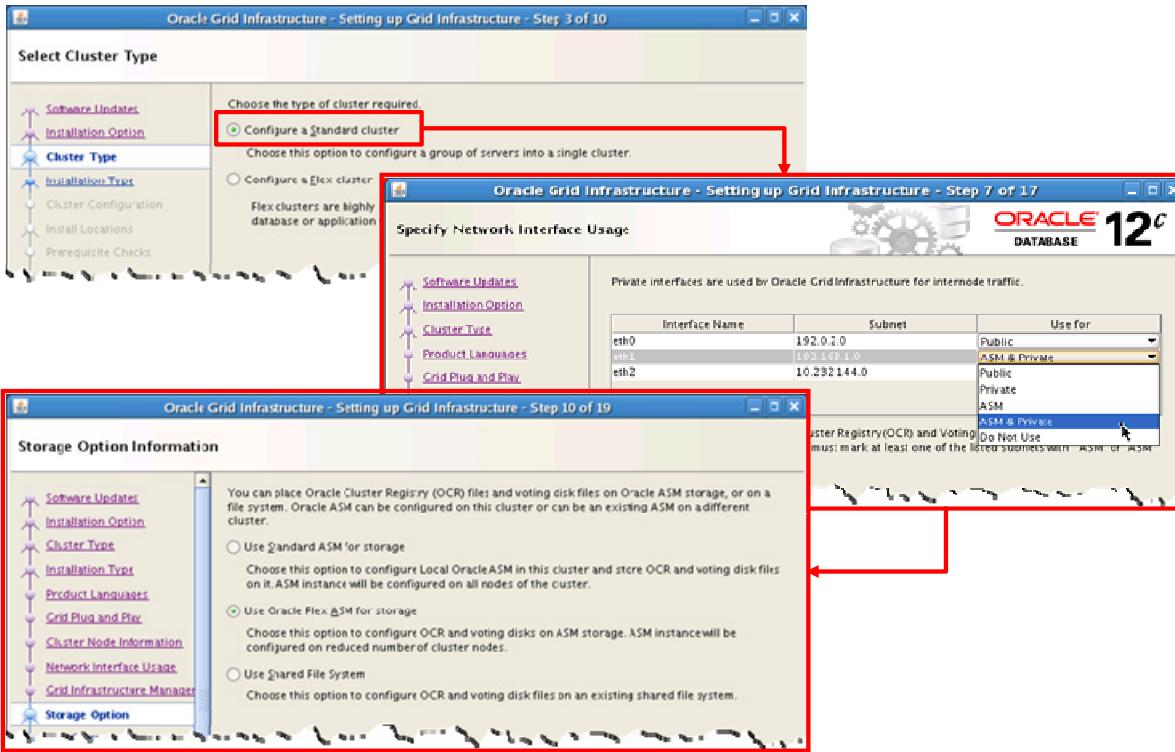
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The ADVM Proxy is a special Oracle instance. Its sole purpose is to enable ASM Dynamic Volume Manager (ADVM), and through it ASM Cluster File System (ACFS), to connect to Flex ASM.

In release 12.1, ACFS, ADVM and the ADVM proxy must reside on the same node. Therefore, by default, the ADVM proxy is configured to run on every node in a standard cluster or every Hub Node in a Flex Cluster. Administrators can shut down the ADVM proxy if ACFS is not running on the node.

The ADVM proxy instance has its ORACLE\_SID set to +APX<node number>.

# Configuring Flex ASM on a Standard Cluster



ORACLE

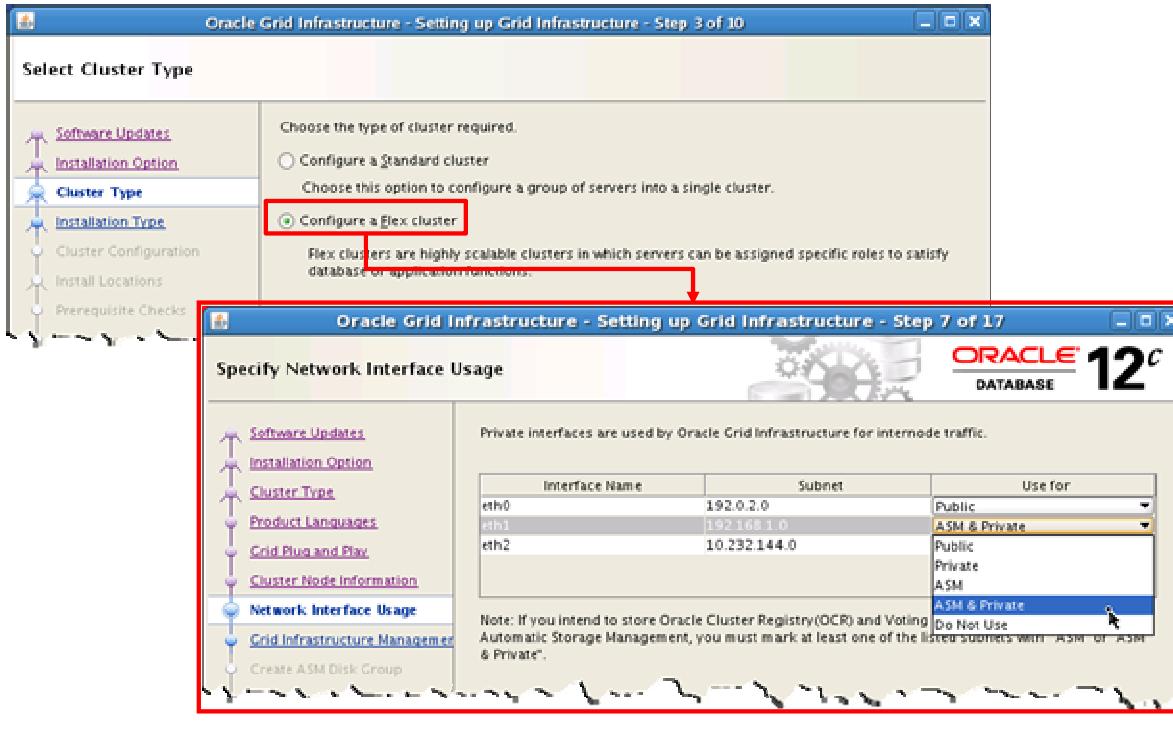
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Universal Installer (OUI) has been updated to facilitate configuration of Flex ASM. The procedure for configuring Flex ASM using OUI differs slightly, depending on whether or not the cluster is also being configured as a Flex Cluster.

To configure Flex ASM on a standard cluster, the following steps are required:

1. Select “Configure a Standard cluster” on the Select Cluster Type screen.
2. Specify an ASM network on the Specify Network Interface Usage screen.
3. Select “Use Oracle Flex ASM for storage” on the Storage Option Information screen.

# Configuring Flex ASM on a Flex Cluster



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

If the option to configure a Flex Cluster is selected on the Select Cluster Type screen, Flex ASM is implicitly configured and you must specify an ASM network on the Specify Network Interface Usage screen.

## Managing Flex ASM Instances

Flex ASM is designed to require minimal monitoring and ongoing management.

- The primary concern is that instances are up and running.

```
$ srvctl status asm -detail  
ASM is running on c00n03,c00n02,c00n01  
ASM is enabled.  
$ srvctl status asm -proxy -detail  
ADVM Proxy is running on c00n04,c00n03,c00n02,c00n01  
ADVM Proxy is enabled.
```

- No Flex ASM-specific instance parameters are required.
- Default settings will effectively support most situations.
- ASM and ADVM Proxy instances use automatic memory management.
  - Minimum default setting: MEMORY\_TARGET=1076M



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flex ASM is designed to require minimal monitoring and ongoing management after initial configuration. The primary concern for administrators is that the ASM instances are up and running. This can be verified by using the `srvctl status` commands shown in the slide.

In release 12.1, no new instance parameters are specific to Flex ASM. In addition, the default parameter settings have been adjusted to suit the Flex ASM architecture, making them sufficient to effectively support most situations.

Automatic memory management is used for ASM instances. In release 12.1, the default setting for `MEMORY_TARGET` is based on various attributes of the node hosting the instance, such as the physical memory size and the number of processor cores.

Note that the minimum default `MEMORY_TARGET` setting (1076M) is significantly larger than the default `MEMORY_TARGET` setting used by ASM instances in previous versions.

# Stopping, Starting, and Relocating Flex ASM Instances

- ASM Instances

```
$ srvctl status asm -detail  
ASM is running on c00n03,c00n02,c00n01  
ASM is enabled.  
$ srvctl stop asm -node c00n03 -f  
$ srvctl start asm -node c00n04  
$ srvctl status asm -detail  
ASM is running on c00n04,c00n02,c00n01  
ASM is enabled.  
$ srvctl relocate asm -currentnode c00n04 -targetnode c00n03  
$ srvctl status asm -detail  
ASM is running on c00n03,c00n02,c00n01  
ASM is enabled.
```

- ADVM Proxy Instances

```
$ srvctl stop asm -proxy -node c00n03  
$ srvctl start asm -proxy -node c00n04
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

At times it may be useful for administrators to control an individual ASM instance or ADVM Proxy instance. The slide shows examples of the `srvctl` commands to stop, start, and relocate individual Flex ASM instances.

# Setting the Cardinality for Flex ASM Instances

- ASM instances

```
$ crsctl status resource ora.asm -f | grep CARDINALITY=
CARDINALITY=3
$ srvctl modify asm -count 4
$ crsctl status resource ora.asm -f | grep CARDINALITY=
CARDINALITY=4
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows examples of the commands required to manage the cardinality setting for ASM instances. To view the current cardinality setting, use the `crsctl status resource` commands shown in the slide. To set the cardinality, use the `srvctl modify` command.

Note that an error is generated if you attempt to set the cardinality to a value smaller than the current number of running instances. In this case, you must first stop the extra instances and then modify the cardinality.

# Monitoring Flex ASM Connections

```
SQL> select distinct i.instance_name asm_instance_name,
  2   c.instance_name client_instance_name, c.db_name, c.status
  3   from gv$instance i, gv$asm_client c
  4   where i.inst_id=c.inst_id;
```

ASM_INSTANCE_NAME	CLIENT_INSTANCE_NAME	DB_NAME	STATUS
+ASM1	+APX1	+APX	CONNECTED
+ASM1	+ASM1	+ASM	CONNECTED
+ASM1	orcl_2	orcl	CONNECTED
+ASM1	orcl_5	orcl	CONNECTED
+ASM1	orcl_7	orcl	CONNECTED
+ASM2	+APX2	+APX	CONNECTED
+ASM2	+ASM2	+ASM	CONNECTED
+ASM2	orcl_1	orcl	CONNECTED
+ASM2	orcl_4	orcl	CONNECTED
+ASM3	+APX3	+APX	CONNECTED
+ASM3	+ASM3	+ASM	CONNECTED
+ASM3	orcl_3	orcl	CONNECTED
+ASM3	orcl_6	orcl	CONNECTED
+ASM3	orcl_8	orcl	CONNECTED



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

At times it may be useful for administrators to know which clients are connected to each ASM instance. This knowledge may be especially useful when considering the impact of shutting down a node for scheduled maintenance or if a change in the cardinality setting for ASM instances is being considered.

To determine the database instances that are connected to a specific ASM instance, ASM administrators can connect to an ASM instance and query the GV\$ASM\_CLIENT table. The example in the slide shows the distribution of eight database instances (orcl\_1 to orcl\_8) across three Flex ASM instances (+ASM1, +ASM2, +ASM3).

## Relocating an ASM Client

- Clients are automatically relocated to another instance if an ASM instance fails.
  - Clients reconnect and the connection is load balanced to an available instance.
- Clients can be manually relocated using the ALTER SYSTEM RELOCATE CLIENT command.
  - The command syntax is:

```
SQL> ALTER SYSTEM RELOCATE CLIENT '<instance_name>:<db_name>';'
```

    - Query GV\$ASM\_CLIENT to determine instance\_name and db\_name.
    - This is useful for manually adjusting the workload balance between instances.



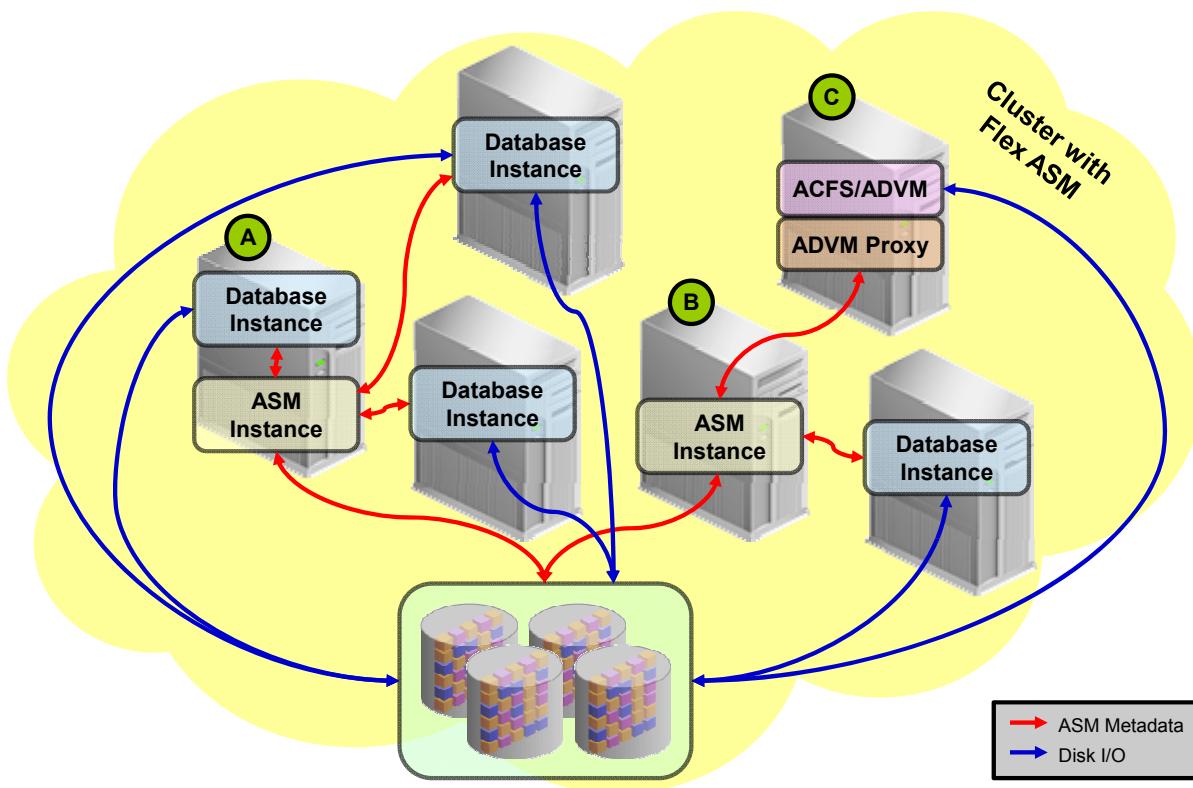
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Flex ASM, if an ASM instance fails, clients are automatically relocated to another instance. When the failure is detected by the client, it reconnects to an available instance. Like any other connection request, reconnection requests are subject to connection load balancing. Relocation due to failure is automatic and transparent to end users.

In addition to automatic relocation, the ALTER SYSTEM RELOCATE CLIENT command can be used to perform manual relocation. This command results in the server terminating the connection to the client, which forces the client to reconnect to the least loaded ASM instance. Manual relocation is useful for manually adjusting the workload balance between ASM instances when a significant imbalance is detected.

Note that if an ASM client is already connected to the least-loaded ASM instance, the ALTER SYSTEM RELOCATE CLIENT command will cause the client to disconnect; however, it will reconnect to the same ASM instance. In this case, to force an ASM client to relocate off the ASM instance, you would need to shut down the ASM instance.

## Flex ASM Deployment Example



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide shows a Flex ASM deployment example. The following notes provide additional detail and summarize some of the key points relating to Flex ASM:

- The diagram illustrates a standard cluster running Flex ASM. The diagram also illustrates the Hub Nodes of a Flex Cluster.
- ASM clients can run on any node in a standard cluster, or any Hub Node in a Flex Cluster. In release 12.1, Flex ASM does not support clients on Leaf Nodes.
- Flex ASM enables a smaller number of ASM instances (two in this example) to service a larger number of clients (four database instances and one ACFS in this example).
- Flex ASM enhances the availability of Oracle Database and ACFS by helping to protect against various ASM failures. If, for example, the ASM instance at node A failed, the three database instances it supports would transparently connect to the ASM instance at node B.
- The ASM cardinality setting specifies the number of ASM instances that should be made available in the cluster. In this example, the ASM cardinality is two. The default cardinality setting for ASM instances is three.

- Depending on the distribution of clients and ASM instances, an ASM client may reside on the same node as an ASM instance (as shown on node A in the diagram), or the ASM instance may reside on a node separate from the ASM clients (as shown on node B in the diagram).
- By default, the ADVM proxy runs on every node in a standard cluster or every Hub Node in a Flex Cluster. For the sake of simplicity, the ADVM proxy is only shown on node C in the diagram, which in this example is the only node running an ASM Cluster File System.

## Quiz

Identify the correct statements regarding server Flex ASM:

- a. Flex ASM requires an ASM instance on each cluster node running an Oracle Database instance.
- b. Flex ASM allows ASM clients to remotely connect to ASM over a network.
- c. With Flex ASM, a small pool of ASM instances can be used to serve a larger pool of database servers.
- d. If an ASM instance fails, the database clients and ASM cluster file systems can reconnect to another ASM instance.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b, c, d

Flex ASM relaxes the hard dependency between ASM and database clients, therefore Flex ASM does not require an ASM instance on each cluster node running an Oracle Database instance.

## Quiz

If OUI is used to install and configure a four node standard cluster with Flex ASM, which statement describes the resulting configuration?

- a. ASM instances run on two cluster nodes for high availability, and more ASM instances are started as the number of ASM clients increases.
- b. ASM instances run on the first three cluster nodes.
- c. Three ASM instances are spread across the cluster.
- d. Each of the four nodes runs an ASM instance.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: c

The default cardinality for ASM instances is three. Regardless of the cluster size, Clusterware attempts to start three ASM instances when a new cluster is configured with Flex ASM. Fewer than three instances may start only if an error prevents an ASM instance from starting, or if there are fewer than three nodes in a standard cluster, or fewer than three Hub Nodes in a Flex Cluster. The ASM instances may start on the first three nodes, as suggested in answer b; however, this will not always be the case.

## Quiz

Which statement best describes the relationship between Flex Clusters and Flex ASM?

- a. There is no relationship, except that both have “Flex” in their names.
- b. A Flex Cluster requires Flex ASM, but Flex ASM does not require a Flex Cluster.
- c. Flex ASM requires a Flex Cluster, but a Flex Clusters does not require Flex ASM.
- d. Flex Clusters and Flex ASM always require each other.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b

A Flex Cluster requires Flex ASM; however, Flex ASM can also run on a standard cluster providing I/O services on a subset of the cluster nodes.

## Summary

In this lesson, you should have learned how to:

- Describe the architecture and components of Flex ASM
- Install and configure Flex ASM
- Administer Flex ASM



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 6 Overview: Database Failover with Flex ASM

In this practice you'll crash an ASM instance and examine how the database client transparently fails over to another Flex ASM instance.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Other ASM New Features

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

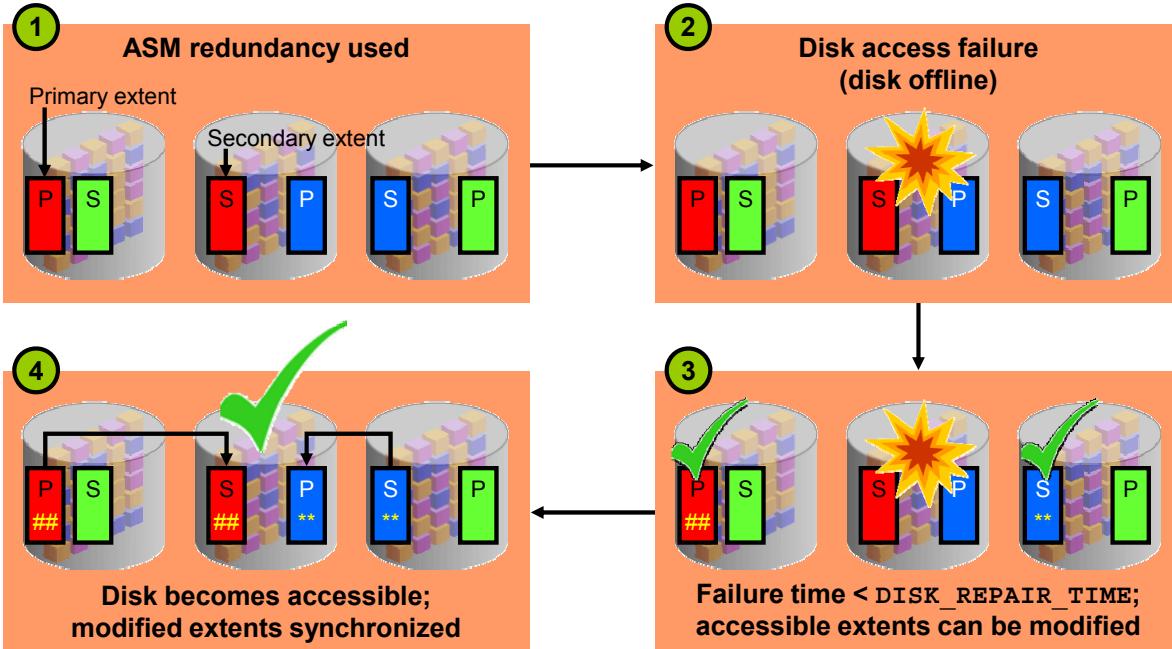
- Describe the other ASM new features in release 12.1
- Perform essential configuration and management associated with these features



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## ASM Fast Mirror Resync: Review

Enabled when `COMPATIBLE.RDBMS >= 11.1`



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Whenever ASM is unable to write an extent, ASM takes the associated disk offline. If the corresponding disk group uses ASM mirroring (NORMAL or HIGH redundancy), at least one mirror copy of the same extent exists on another disk in the disk group.

ASM fast mirror resync is used to efficiently deal with transient disk failures. When a disk goes offline following a transient failure, ASM tracks the extents that are modified during the outage. When the transient failure is repaired, ASM can quickly resynchronize only the ASM disk extents that have been modified during the outage. Note that the tracking mechanism uses one bit for each modified extent and is very efficient.

Using ASM fast mirror resync, the failed disk is taken offline but not dropped if you have set the `DISK_REPAIR_TIME` attribute for the corresponding disk group. The setting for this attribute determines the duration of disk outages that ASM will tolerate while still being able to resynchronize after the failed disk is repaired. The default setting for the `DISK_REPAIR_TIME` attribute is 3.6 hours. If a disk remains offline longer than the time specified by the `DISK_REPAIR_TIME` attribute, the disk is dropped from the disk group and the disk group is rebalanced.

Numerous enhancements have been made to ASM fast mirror resync in Oracle Database 12c. The following pages introduce those enhancements.

## Controlling the Resources Used by Resync

A power limit can be set for disk resync operations.

- This is conceptually similar to the power limit setting for disk group rebalance.
- The range is 1 (least resources) to 1024 (most resources).
- If not specified, the default setting is 1.
- Examples:

```
SQL> ALTER DISKGROUP DATA ONLINE DISK data_0000 POWER 100;  
ASMCMD> online -G DATA -D data_0000 --power 100  
SQL> ALTER DISKGROUP DATA ONLINE ALL POWER 500;  
ASMCMD> online -G DATA -a --power 500
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c, ASM allows administrators to control the amount of resources that are dedicated to disk resync operations. This is conceptually similar to the capability in previous versions, which allowed administrators to control the amount of resources that are dedicated to a disk group rebalance operation.

To control the amount of resources dedicated to disk resync operations, administrators can specify a power limit setting that is an integer value between 1 and 1024. Because lower numbers dedicate few resources, the operation takes longer but has minimal impact on other work. Higher values allow the operation to finish quicker at the cost of potentially impacting other work.

To specify the power limit setting for a resync operation, use the `ALTER DISKGROUP` SQL command or the `ASMCMD ONLINE` command. Examples of both commands are shown in the slide.

## More Efficient Disk Replacement

Administrators now have the option to replace a disk as a fast, efficient, atomic operation.

- No disk group reorganization is necessary.
  - Previously the original disk had to be dropped and the replacement disk added, requiring reorganization of the entire disk group.
- The replacement disk is populated with mirror copies of ASM extents from other disks.
  - The disk being replaced must be offline.
  - The replacement disk takes the same name as the original disk.
  - The replacement disk becomes part of the same failure group as the original disk.
- Example:

```
SQL> ALTER DISKGROUP DATA REPLACE DISK DATA_0001 WITH '/dev/disk2';
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If an ASM disk goes offline and cannot be repaired, administrators require the ability to replace the disk. In prior versions of ASM, there was no command to replace a disk. Rather, administrators had to drop the faulty disk and then add a new disk to the disk group. In the process, the entire disk group was rebalanced. Depending on various attributes of the disk group, the rebalance operation could be quite expensive and time-consuming.

Oracle Database 12c allows administrators to simply replace an offline disk by using one fast and efficient operation. Using this operation, the replacement disk is populated with mirror copies of the ASM extents from other disks, and there is no need for any additional reorganization or rebalancing across the rest of the disk group. Note that the replacement disk takes the same name as the original disk and becomes part of the same failure group as the original disk.

## Dealing with Transient Failure on a Failure Group

Administrators now have the option to specify a failure group repair time.

- This is similar to the existing disk repair time.
- Repair time is set by using a new disk group attribute:  
`failgroup_repair_time`
  - The default setting is 24 hours.
  - Configuration examples:

```
SQL> ALTER DISKGROUP DATA
      SET ATTRIBUTE 'failgroup_repair_time' = '48h';

SQL> CREATE DISKGROUP DATA2 HIGH REDUNDANCY
      FAILGROUP FG1 DISK '/dev/disk1*'
      FAILGROUP FG2 DISK '/dev/disk2*'
      FAILGROUP FG3 DISK '/dev/disk3*'
      ATTRIBUTE 'failgroup_repair_time' = '12h';
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When individual disks fail, the failure is often terminal and the disk must be replaced. When all the disks in a failure group fail simultaneously, it is unlikely that all the disks individually failed at the same time. Rather it is more likely that some transient issue caused the failure. For example, a failure group could fail because of a storage network outage, a network attached storage (NAS) device rebooting, or a software crash on a storage server.

Failure group outages are more likely to be transient in nature, and replacing all disks in a failure group is a much more expensive operation than replacing a single disk. Therefore, it would typically make sense to wait longer to allow a failure group to be repaired, rather than dropping all its disks from the associated disk groups.

With Oracle Database 12c, ASM provides administrators with the ability to specify a failure group repair time in addition to the disk repair time that was previously available. The failure group repair time can be set by using a new disk group attribute,  
`FAILGROUP_REPAIR_TIME`. The default failure group repair time is 24 hours.

Now by default, if all disks in a failure group fail simultaneously, ASM will not drop the disks when the disk repair time expires after 3.6 hours. Rather, the administrator will have 24 hours to correct the failure. If the failure is corrected before the time expires, the disks are updated using a fast mirror resync. If the failure is not corrected in time, ASM must automatically drop all disks in the failure group.

## Resync Time Estimate

Each resync operation shown in V\$ASM\_OPERATION now includes a time estimate.

For example:

```
SQL> SELECT PASS, STATE, EST_MINUTES FROM V$ASM_OPERATION;  
  
PASS      STAT EST_MINUTES  
-----  
RESYNC    RUN      1  
REBALANCE WAIT     1  
COMPACT   WAIT     1
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of the new time estimates that accompany resync operations displayed in the V\$ASM\_OPERATION view.

## Resync Checkpoint and Auto-Restart

- Resync operations can be interrupted. For example:
  - A disk group is dismounted by using the `FORCE` option.
  - An ASM instance fails.
- In previous versions:
  - Administrators had to manually reexecute the affected command
  - The entire resync operation had to be reexecuted
- With Oracle Database 12c ASM:
  - Interrupted resync operations are automatically restarted
  - Resync operations are broken into phases
    - A checkpoint marks the end of each resync phase, and stale extent metadata is cleared.
    - Interrupted resync operations restart from the last checkpoint prior to the interruption.



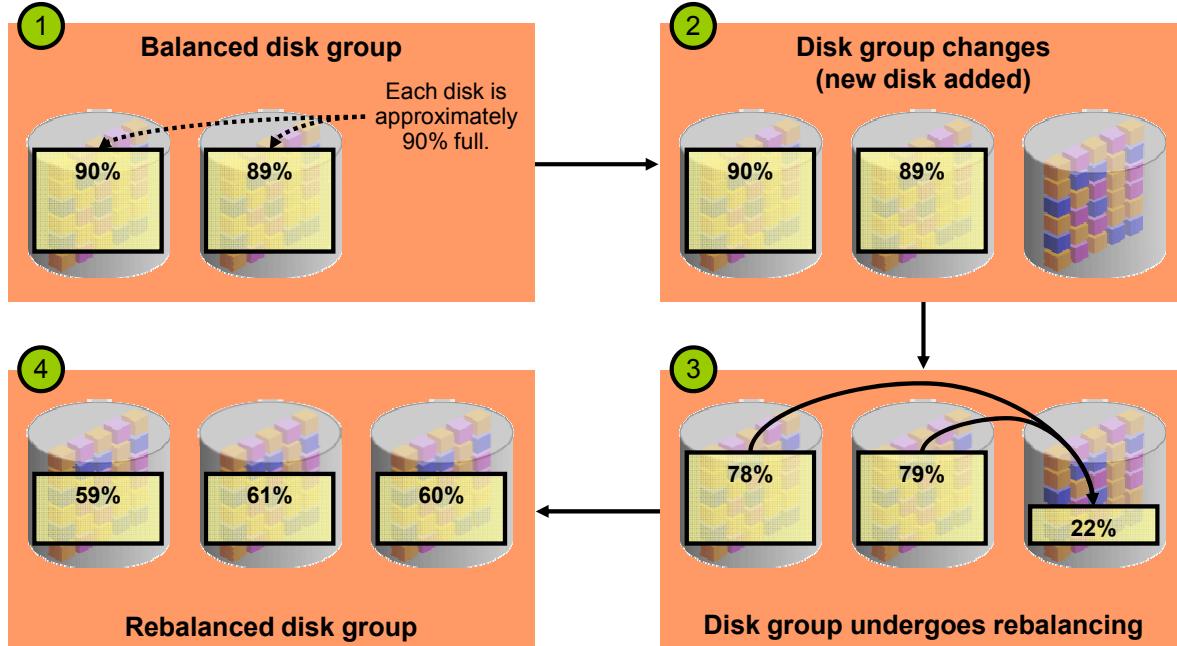
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A disk online operation, and associated disk resync, could be interrupted in various ways. For example, an interruption could occur because a disk group was dismounted with the force option or an entire ASM instance failed.

Previously, administrators had to manually reissue the interrupted command. In addition, the metadata used to identify the extents that require resyncing (the stale extents) was only cleared at the end of the resync operation. If the resync operation was interrupted for any reason, the entire operation must be reexecuted.

With Oracle Database 12c ASM, interrupted resync operations are automatically restarted. In addition, resync operations are internally broken into numerous phases, and the stale extent metadata is cleared at the end of each phase. Now, if a resync operation is interrupted and restarted, the completed phases can be skipped and processing can recommence at the beginning of the first remaining incomplete phase.

# ASM Disk Group Rebalance: Review



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An ASM disk group is balanced when there is an even distribution of data across all of the available disks. In the normal course of operations, ASM maintains balanced disk groups. A disk group can become unbalanced for various reasons, most commonly when disks are either added to or removed from the disk group. To restore balance, a rebalance operation is required, and it involves moving ASM extents from the most filled disks to the least filled disks.

Enhancements have been made to ASM disk group rebalance in Oracle Database 12c ASM. The following pages introduce those enhancements.

## Rebalance Work Estimates

Oracle Database 12c ASM provides more accurate rebalance work estimates.

- Work estimates are based on a detailed work plan.
- A work plan can be generated and viewed separately:
  - The EXPLAIN WORK command generates the work plan.
  - The work plan is visible in the V\$ASM\_ESTIMATE view.
  - Example:

```
SQL> EXPLAIN WORK SET STATEMENT_ID='Drop DATA_0000'
  2  FOR ALTER DISKGROUP DATA DROP DISK DATA_0000;

Explained.

SQL> SELECT EST_WORK FROM V$ASM_ESTIMATE
  2  WHERE STATEMENT_ID='Drop DATA_0000';

EST_WORK
-----
279
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Previous ASM releases provided work estimates for ASM rebalance operations. The accuracy of these estimates was highly variable.

With Oracle Database 12c ASM, a more detailed and more accurate work plan is created at the beginning of each rebalance operation.

In addition, administrators can separately generate and view the work plan before performing a rebalance operation. That way, they can better plan and execute various changes, such as adding storage, removing storage, or moving between different storage systems.

Administrators can generate the work plan by using the EXPLAIN WORK command. The work estimate can be viewed in the V\$ASM\_ESTIMATE view. The EST\_WORK column shows the estimated number of ASM allocation units that must be moved by the operation.

Note that the V\$ASM\_ESTIMATE view does not provide a time estimate, such as the one provided in V\$ASM\_OPERATION, because the time estimate in V\$ASM\_OPERATION is based on the current work rate observed during execution of the operation. Because the current work rate can vary considerably, due to variations in the overall system workload, administrators should use knowledge of their environment and workload patterns to convert the data in V\$ASM\_ESTIMATE into a time estimate, if required.

## Priority Ordered Rebalance

- Sometimes, rebalance operations are required to restore redundancy. For example, a disk fails and no replacement is available.
- In previous versions:
  - The rebalance occurs in file-number order
  - A secondary failure could result in the loss of a critical file
- With Oracle Database 12c ASM:
  - Critical files, such as control files and log files, are restored before data files
  - Secondary failure is less likely to result in critical file loss



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In some situations, rebalance operations are required to restore data redundancy within disk groups that use NORMAL or HIGH ASM redundancy. For example, if a disk fails and no replacement is available, a rebalance is required to redistribute the data across the remaining disks and to restore redundancy.

With Oracle Database 12c ASM, priority ordered rebalance is implemented. This capability concentrates on quickly restoring the redundancy of critical files, such as control files and online redo log files, to ensure that they are protected in case a secondary failure occurs soon afterwards.

## Proactively Validating Data Integrity

In previous versions, data is checked for logical consistency when it is read.

- If a logical corruption is detected:
  - Automatic recovery can be performed by using the mirror copies
  - Manual recovery can also be performed by using RMAN
- For seldom-accessed data, corrupted data could be present in the system for a long time between reads.
  - The possibility that all mirrors are corrupted increases over time.

With Oracle Database 12c, data can be proactively scrubbed.

- Areas can be scrubbed on demand.
- Scrubbing occurs automatically during rebalance operations.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous Oracle Database versions, when data was read, checks were performed on data to validate its logical consistency. If a logical corruption was detected, ASM could automatically recover by reading the mirror copies on NORMAL and HIGH redundancy disk groups.

One problem with this approach is that corruption of seldom-accessed data could go unnoticed in the system for a long time between reads. In addition, the possibility of multiple corruptions affecting all mirror copies of data increases over time; therefore, seldom-accessed data may simply be unavailable when it is required.

Oracle Database 12c introduces proactive scrubbing capabilities that check for logical corruptions and automatically repair them, where possible. In release 12.1, scrubbing can occur in two different ways:

- Scrubbing can also occur as part of a rebalance operation.
- On-demand scrubbing can be performed on specific areas by an administrator.

## Proactive Content Checking During Rebalance

Data read during rebalance is scrubbed.

- If enabled, checks are automatic with automatic error correction for mirrored data.
- Checks are enabled with the disk group attribute content.check.
  - Configuration example:

```
SQL> ALTER DISKGROUP DATA  
      SET ATTRIBUTE 'content.check' = 'TRUE';
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Previously, when an ASM extent was moved during a rebalance operation, it was read and written without any additional content checks. With Oracle Database 12c ASM, extents read during rebalance can be scrubbed to ensure their logical integrity.

Rebalance scrubbing can be enabled or disabled for a disk group by using the content.check disk group attribute. An example of this attribute's setting is shown in the slide. By default, rebalance scrubbing is disabled (content.check=FALSE).

## On-Demand Scrubbing

- Use the ALTER DISKGROUP . . . SCRUB command to perform on-demand scrubbing.

Examples:

```
SQL> ALTER DISKGROUP DATA SCRUB REPAIR;  
SQL> ALTER DISKGROUP DATA SCRUB FILE  
  2  '+DATA/ORCL/DATAFILE/SYSTEM.270.775354873'  
  3  REPAIR WAIT;  
SQL> ALTER DISKGROUP DATA SCRUB DISK DATA_0000  
  2  REPAIR POWER MAX FORCE;
```

- On-demand scrubbing operations can be monitored by using the V\$ASM\_OPERATION view.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On-demand scrubbing can be performed by using the ALTER DISKGROUP . . . SCRUB command. On-demand scrubbing can be performed on a disk group or on individual files or individual disks. An example of each command is shown in the slide. The progress of on-demand scrubbing operations can be monitored by using the V\$ASM\_OPERATION view.

Following are details regarding the various options associated with on-demand scrubbing:

- If the REPAIR option is not specified, the specified disk group, file, or disk is only checked and any logical corruptions are reported.
- The POWER option can be manually set to LOW, HIGH, or MAX. If the POWER option is not specified, the scrubbing power is automatically controlled based on the system I/O load.
- If the WAIT option is not specified, the operation is added into the scrubbing queue and the command returns immediately. If the WAIT option is specified, the command returns after the scrubbing operation is completed.
- If the FORCE option is specified, the command is processed immediately regardless of the system I/O load.

## Errors During Scrubbing

Error messages associated with ASM scrubbing:

```
ORA-15xxx: Logical corruption detected at [file, extent number,  
block#] [disk, au]
```

- The above error is accompanied by the following one if the REPAIR option is specified, but the repair is not successful.  

```
ORA-15xxx: Logical corruption cannot be repaired
```
- More details about the corruption and the reason of the failed repair attempt are written into the trace file.  

```
ORA-15xxx: Logical corruption checking request was denied
```
- An on-demand scrubbing request is denied because the I/O load of the system is high or scrubbing is disabled.



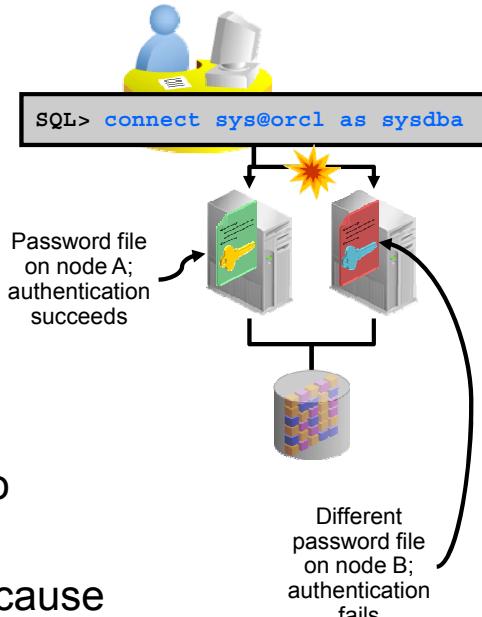
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Errors encountered during the scrubbing process are logged to trace files. The slide lists the new error messages associated with scrubbing.

# Managing Password Files: Background

Prior to Oracle Database 12c:

- Password files are located under \$ORACLE\_HOME/dbs
- Separate copies are typically maintained on each node
- Passwords could differ across the cluster resulting in unpredictable login behavior
- Diligent maintenance is required to keep password files consistent
- This is a critical issue for ASM, because the password file is required for all non-operating system authentication.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Prior to Oracle Database 12c, each instance in a cluster typically maintained its own separate copy of the password file. This applies to both database instances and ASM instances. The obvious drawback with this approach is that multiple copies of the password file can become unsynchronized across the cluster.

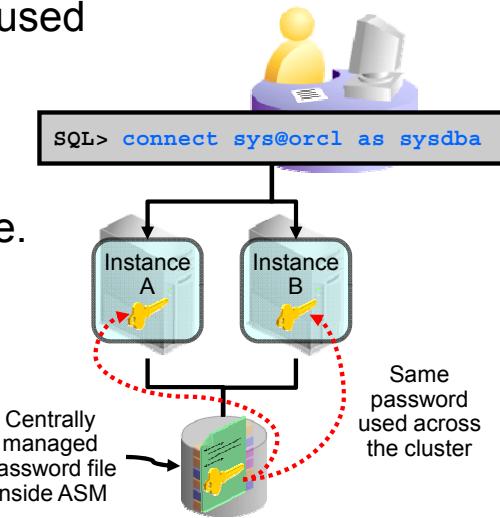
Oracle Database 11g, release 2 attempts to address this issue by synchronizing password file updates across the cluster, so that each available instance updates its own password file simultaneously. However, if an instance is down, the update to its password file is lost.

Clearly, having multiple copies of the password file creates a manageability issue, requiring administrators to synchronize the password files across the cluster. This issue is more severe for ASM. Because ASM lacks a persistent data dictionary, it therefore relies on the password file for authentication even when the ASM instance is up and running.

## Storing Password Files Inside ASM

Oracle Database 12c ASM provides a mechanism to maintain shared password files inside ASM.

- Password file consistency issues disappear.
- The ASM-based password file is used by default when ASM is used.
- The password file location is stored as an attribute of the database or ASM cluster resource.
- The password files stored inside ASM can be used only after ASM is started.
- OS authentication is used to start ASM.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c ASM provides a mechanism to maintain a shared password file on an ASM disk group to facilitate password file–based authentication. Oracle Database 12c also checks ASM first when searching for a password file.

For new installations, password files are no longer located in the `$ORACLE_HOME/dbs` directory. Instead, they are located on an ASM disk group, with the exact location stored as an attribute of the cluster resource that is associated with the corresponding database or ASM.

Authentication by the shared password file is available only after the ASM instance is up and running and the disk group with the password file is mounted. Operating system authentication is used to start the ASM instance. However, ASM is typically started automatically during the Clusterware startup sequence. Clusterware utilities, such as `crsctl` and `srvctl`, along with the Enterprise Manager agent will typically continue to use OS authentication.

Note that the `compatible.asm` disk group attribute must be set to 12.1 or later to enable storage of shared password files in an ASM disk group.

# Managing Password Files Inside ASM

- Creating a new password file inside ASM
  - Database password file examples:

```
$ orapwd file='+DATA' dbuniquename='orcl' password='welcome'
```

```
ASMCMD> pwcreate --dbuniquename orcl +DATA welcome
```

- ASM password file examples:

```
$ orapwd asm=y file='+DATA' password='welcome2asm'
```

```
ASMCMD> pwcreate --asm +DATA welcome2asm
```

- Locating the password file inside ASM:

```
$ crsctl status resource ora.orcl.db -f
```

```
...
```

```
PWFILE='+DATA/orcl/orapworcl
```

```
...
```

```
ASMCMD> pwget --dbuniquename orcl
```

- Moving an existing password file to ASM:

```
ASMCMD> pwmove --dbuniquename orcl /home/oracle/dbs/orapworcl +DATA
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

As with previous versions, the ORAPWD utility is used to manage the password file. Anyone with the SYSASM privilege on ASM can create the ASM password file inside ASM, whereas anyone who has SYSASM or SYSDBA privilege on ASM can create a database password file inside ASM.

In addition to ORAPWD, the ASM command utility, ASMCMD, is extended with a variety of commands that can be used to manage ASM-based password files.

The slide shows a series of examples for managing password files inside ASM.

## Even Read

In previous versions:

- By default, ASM always reads the primary copy of mirrored data if it is available.
- Alternatively, preferred read failure groups could be configured.

With Oracle Database 12c ASM:

- Even Read distributes data reads evenly across all disks.
  - Each read request is sent to the least-loaded available disk.
  - Even Read is transparent to applications and enabled by default in non-Exadata environments.
  - Users on I/O bound systems should notice a performance improvement.
- Preferred read failure groups can still be configured.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, the default behavior for ASM is to always read the primary copy of a mirrored extent unless a failure condition requires otherwise.

Alternatively, administrators can configure preferred read failure groups, by using the `ASM_PREFERRED_READ_FAILURE_GROUPS` instance parameter, to specify the failure group from which each ASM instance should read.

The Even Read feature of ASM, introduced in release 12.1, distributes data reads evenly across all disks in a disk group. For each I/O request presented to the system, one or more disks may contain the data. With Even Read enabled, each read request is sent to the least loaded of the available disks.

Even Read is enabled by default on all release 12.1 (and later) database and ASM instances in non-Exadata environments. Because Even Read is transparent to applications, users on I/O bound systems should notice a performance improvement after upgrading to release 12.1.

Note that Even Read only attempts to equalize the number of reads on each disk across each disk group. It does not measure the latency or performance of each read. Therefore, Even Read does not replace the functionality that is provided by preferred read failure groups.

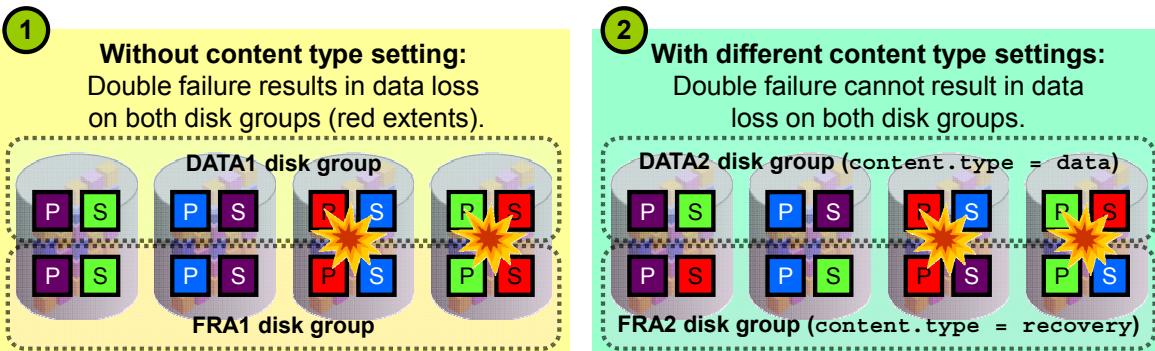
# Specifying the Content Type for a Disk Group

Administrators can specify the content type for each disk group.

- New disk group attribute: `content.type`
  - Possible values: data, recovery or system
  - Configuration example:

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'content.type'='data';
```

- Decreases the likelihood that multiple failures impact disk groups with different content type settings



ORACLE

Copyright © 2013, 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.

With Oracle Database 12c, ASM provides administrators with the option to specify the content type associated with each ASM disk group. This capability is provided by a new disk group attribute, `CONTENT . TYPE`. 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 FRA1, 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 FRA2, 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 FRA2 uses a different adjacency measure, and because of this, the contents of FRA2 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, FRA2 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.

Note also 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.

## Improved Error Reporting for Cluster Validation Failures

- Prior releases only reported that the operation failed:

```
ORA-15075: disk(s) are not visible cluster-wide
```

- With Oracle Database 12c ASM, the relevant disk and instance is identified:

```
ORA-15075: disk 'data_0000' is not visible on instance '+ASM2'
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In certain circumstances, a disk may not be accessible from one or more nodes in a cluster. For example, an `ALTER DISKGROUP ADD DISK` command specified a disk that could not be discovered by one or more nodes in the cluster. Previously, an error message reported the failure without specifying the disks or nodes involved. Now, a more detailed error message, including the failed disk and affected instance, is produced.

## Exadata Copy Offload

With Oracle Database 12c ASM, most rebalance tasks are offloaded to Exadata Storage Server.

- Each offload request can replace numerous I/O requests.
- Rebalance uses fewer I/O resources and executes quicker.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c ASM, most extent relocations performed by a rebalance operation can be offloaded to Exadata Storage Server. Using this capability, a single offload request can replace multiple read and write I/O requests. Offloading relocations avoids sending data to the ASM host, improving rebalance performance.

## Bulk File Ownership Changes

- In previous versions:
  - There was no simple way to remove an ASM user without deleting all their files
  - There was no simple way to transfer ownership of a specified user's files to another user
- With Oracle Database 12c ASM:
  - All files in a specified disk group that are owned by a specific user can be transferred to another user.

Examples:

```
SQL> ALTER DISKGROUP data REPLACE USER oldusr WITH newusr;  
ASMCMD> rpush data oldusr newusr
```

- This simplifies user removal without deleting files.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous ASM versions, there was no easy way to remove a user without deleting all the files that the user owned. File ownership could be changed; however, the administrator had to identify all relevant files and carefully transfer ownership to ensure that only the relevant files were affected.

With Oracle Database 12c ASM, members of the `ORA_ASMADMIN` operating system group can perform bulk file ownership changes by using the `ALTER DISKGROUP ... REPLACE USER` command. Using this command, ownership of all files in a disk group owned by one user can be transferred to another user.

This capability facilitates the quick transfer of file ownership between different administrators, and also makes it easier to remove an ASM user account.

# Changing ASM Privileges on Open Files

- In previous versions:
  - File ownership and permission modifications could be performed only on closed files
    - Modifications had to wait until an appropriate maintenance time.
- With Oracle Database 12c ASM:
  - File ownership and permission changes can be performed on open files
    - New settings take effect when reauthentication is required.
    - It is conceptually similar to permission changes on UNIX files.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, file ownership and permission modifications could be performed only when the file was closed. Often, administrators were forced to wait until databases were shut down before performing file ownership and permission modifications.

With Oracle Database 12c ASM, this constraint is removed and users are able to perform file ownership and permission modifications on open files. The new set of permissions takes effect when reauthentication is required. This behavior is similar to the way that permission modifications are performed on UNIX platforms.

This capability introduces no new commands or administrator tasks, but operations that previously failed now work.

## ASM File Access Control Available on Windows

- ASM file access control enables file ownership and permission settings on ASM files.
  - This is conceptually similar to UNIX file ownership and permissions.
- In previous versions:
  - ASM file access control was infeasible on Windows because Oracle had to run by using the LOCALSYSTEM account
- With Oracle Database 12c:
  - Any Windows user can install and run Oracle.
  - ASM file access control can be implemented.
    - This is functionally equivalent to UNIX and Linux platforms.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ASM file access control restricts file access to specific Oracle ASM clients that connect to ASM. An ASM client is typically a database, which is identified by the user that owns the database home directory. ASM file access control is particularly useful in environments used to consolidate many Oracle databases, and can be used to ensure that database administrators can access only the database files that are associated with the databases under their control.

In previous versions, ASM file access control could not be implemented on Windows because ASM and database processes (threads) had to run by using the LOCALSYSTEM account.

Oracle Database 12c removes this restriction and enables different Windows users to install and run Oracle Database and ASM. This, in turn, allows ASM file access control to be made available on Windows, with functionality equivalent to UNIX and Linux platforms.

## Quiz

Identify the correct statement regarding ASM support for storing password files:

- a. ASM can be used to store only the ASM password file.
- b. ASM can be used to store the ASM password file and database password files.
- c. ASM can be used to store only database password files.  
The ASM password file must be maintained outside ASM so that it can be used to authenticate ASM administrators prior to starting ASM.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b

ASM can be used to store the ASM password file and database password files. Operating system authentication is used to start ASM.

## Quiz

Identify the correct statements regarding ASM data scrubbing:

- a. Scrubbing ensures that data is clean by checking that data values are logically consistent with database constraints.
- b. Scrubbing proactively checks ASM extents to ensure that they are logically consistent.
- c. By detecting corruptions earlier, scrubbing reduces the possibility that multiple corruptions affect all mirror copies of data.
- d. Scrubbing can occur automatically as part of a rebalance operation, or specific areas can be scrubbed on demand.
- e. Scrubbing replaces the logical consistency checks that are performed when ASM performs a data read.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b, c, d

Scrubbing proactively checks ASM extents to ensure that they are logically consistent. Scrubbing does not perform any checks involving database constraints. Scrubbing complements the existing checks performed by ASM when data is read; it does not replace any checks.

## Summary

In this lesson, you should have learned how to:

- Describe the other ASM new features in release 12.1
- Perform essential configuration and management associated with these features



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 7 Overview:

- Practice 7-1: More Efficient Disk Replacement
  - In this practice, you will exercise a new command that enables an ASM disk to be efficiently replaced by another ASM disk. You will also see how the new disk replacement capability compares with the capability available in previous versions.
- Practice 7-2: Managing ASM-Based Password Files
  - In this practice, you will examine the default-created password files inside ASM. You'll then delete and re-create the ASM-based password file for a database by using the `orapwd` utility.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Cloud FS New Features

# 8



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe each of the Cloud FS new features
- Configure each new feature
- Administer each new feature



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Introducing Oracle Cloud File System

- Oracle Cloud File System incorporates:
  - ASM Cluster File System (ACFS)
  - ASM Dynamic Volume Manager (ADVM)
- New Features in Oracle Cloud File System, release 12.1:
  - High availability NFS
  - Snapshot enhancements
  - Support for all Oracle Database files
  - Advanced auditing
  - Metrics plug-in
  - Replication enhancements
  - Tagging API
  - Resource enhancements
  - Broader platform support
  - And more...



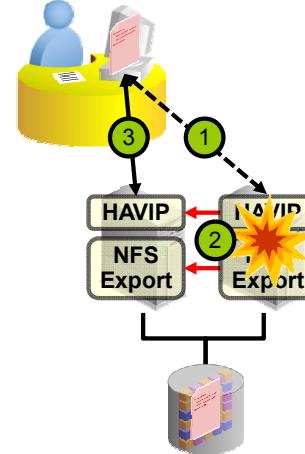
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In Oracle Database 12c, Oracle Cloud File System (Cloud FS) incorporates the capabilities provided by ASM Cluster File System (ACFS) and ASM Dynamic Volume Manager (ADVM). Along with the new name, Cloud FS introduces numerous enhancements and new features. The remainder of this lesson covers the Cloud FS enhancements and new features.

# High Availability NFS: Overview

High Availability NFS (HANFS) provides an uninterrupted NFS service.

- Exported file systems are exposed by using Highly Available Virtual IPs (HAVIPs).
- Oracle Clusterware manages the NFS exports and HAVIPs.
  - Services are automatically migrated if the current node fails.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c, Cloud FS includes High Availability NFS (HANFS). HANFS provides uninterrupted service of NFS exported paths by exposing NFS exports on Highly Available Virtual IPs (HAVIPs). Oracle Clusterware agents are used to ensure that the HAVIPs and NFS exports are always online. If a cluster node fails, the HAVIPs and NFS exports are automatically migrated to a surviving node.

HANFS works in conjunction with NFS version 2 and NFS version 3.

# Configuring High Availability NFS

- Ensure that NFS is running.
- After creating an ACFS file system:
  - Register the ACFS file system as a cluster resource:

```
# srvctl add filesystem -d /dev/asm/voll-201 \
> -m /mnt/acfsmounts/acfs1
```

- Mount the ACFS file system on all cluster nodes:

```
# srvctl start filesystem -device /dev/asm/voll-201
```

- Register a new HAVIP resource:

```
# srvctl add havip -address c01vip -id havip1
```

- Register the ACFS file system export:

```
# srvctl add exportfs -id havip1 -path /mnt/acfsmounts/acfs1 \
> -name export1 -options rw -clients *.example.com
```

- Export the file system:

```
# srvctl start exportfs -name export1
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Before configuring HANFS, ensure that an NFS server is running on the required cluster nodes and that you have created the ACFS file systems that you want to expose with HANFS.

HANFS requires the underlying ACFS file system to be registered as a cluster resource and mounted on multiple cluster nodes. This can be achieved by using the `srvctl add filesystem` and `srvctl start filesystem` commands, as shown in the examples in the slide. The key parameter for the `srvctl ... filesystem` commands is the device that is associated with the ACFS file system. Administrators can view the device that is associated with an ACFS file system by using the `volinfo --all` command in the ASMCMD command-line utility.

In addition, a HAVIP resource must be created. The key parameter for the HAVIP resource is the address, which is specified by using a hostname or IP address. The corresponding IP address must be a single static address (no dynamic host configuration protocol [DHCP] or round-robin domain name server [DNS] resolution), not currently in use, and on the same subnet as the existing node VIPs.

After the HAVIP is defined, an EXPORTFS resource must be created. To create the EXPORTFS resource, you must specify the HAVIP resource that will be used to export the file system, the path of the ACFS file system being exported, and a name that is used to identify the resource. You can also specify other NFS options and allowed clients.

After all the resources are in place, the file system can be exported by using the `srvctl start exportfs` command. Exporting the file system automatically starts the associated HAVIP.

# Cloud FS Snapshot Enhancements

- Snaps-of-Snaps
  - Existing snapshots can be used as the source for a new snapshot.
    - Any combination of read-only and read-write snapshots
  - An ACFS file system can have up to 63 snapshots, including Snaps-of-Snaps.
  - Command syntax:

```
$ acfsutil snap create [-w|-r] -p <parent_snap_name> <snap_name> <mountpoint>
```
- Conversion between read-only and read-write snapshots
  - Command syntax:

```
$ acfsutil snap convert -w|-r <snap_name> <mountpoint>
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c, Cloud FS supports the creation of snapshots based on an existing snapshot of the same ACFS file system (otherwise known as *Snaps-of-Snaps*). Any combination of read-only and read-write snapshots is supported. For example, a read-write snapshot can be based on an existing read-only snapshot, and a read-only snapshot can be based on an existing read-write snapshot. Each ACFS file system can support a total of 63 snapshots, including Snaps-of-Snaps.

In addition, snapshot conversions are enabled between read-only and read-write snapshots. Conversion in either direction is supported. For example, a read-only snapshot can be converted to a read-write snapshot, then modified, and finally converted back to a read-only snapshot.

## Cloud FS Support for All Oracle Database Files

- In previous versions, ACFS provided limited support for Oracle Database files.
- With Oracle Database 12c, Cloud FS is supported for storing all Oracle Database files.
  - Entire running databases can be stored inside Cloud FS.
  - Database administrators can leverage Cloud FS services:
    - Snapshots
    - Security
    - Tagging



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, ACFS was supported as a store for only a very limited set of Oracle Database files, including offline data file copies, archived log files, and various trace files and dump files.

With Oracle Database 12c, Cloud FS is supported as a store for all Oracle Database files. This means that entire running Oracle databases can be stored inside Cloud FS. As a result, database administrators will be able to leverage the power of Cloud FS to open up exciting new possibilities.

For example, Cloud FS snapshots could be used as a point-in-time image for backups while database processing continues uninterrupted. In addition, snapshots could be used to quickly reset databases to a known state for system testing and other activities.

Note that with the initial release of Oracle Database 12c, support will not be available for:

- Oracle Database files on file systems that use Cloud FS replication
- Cloud FS encryption on Oracle Database files or redo log files

## Configuration Settings for Database Files on Cloud FS

- Set the ASM and ADVM compatibility attributes to 12.1.
  - Required to enable new ASM and ADVM features
  - Command syntax:

```
SQL> [ CREATE | ALTER ] DISKGROUP ...
      ATTRIBUTE 'compatible.asm' = '12.1',
      'compatible.advm' = '12.1';
```
- Set stripe columns to 1 for the ADVM volume.
  - Disables ADVM volume striping
  - Command syntax:

```
ASMCMD> volcreate -G <diskgroup> -s <size> --column 1 <volume>
```
- Set FILESYSTEMIO\_OPTIONS=SETALL in the database initialization parameter file.
  - Enables direct I/O for the database, bypassing the OS file system cache



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide outlines the recommended configuration settings to achieve optimal performance with database data files on Cloud FS.

## Cloud FS Auditing: Overview

With Oracle Database 12c, Cloud FS introduces a general audit framework for file systems.

- A separate audit trail can be defined for each file system.
- It enables separation of duties to be enforced.
- A collector for Oracle Audit Vault is also available.
  - Audit Vault provides secure offline audit trail storage with built-in analysis and reporting tools.
- Consistent functionality is provided across all platforms that are supported by Cloud FS.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In previous versions, ACFS provided some support for auditing; however, there was no unified management framework and no separation of duties to help protect the integrity of the audit trail.

With Oracle Database 12c, Cloud FS introduces a general auditing framework for ACFS file systems. This auditing framework can produce a separate audit trail for each file system, and enforce separation of duties regarding the management and review of the audit trail.

Along with the generation of a file system audit trail, a collector for Oracle Audit Vault has been developed. This collector is separate from Cloud FS, but provides a way for Cloud FS audit data to be integrated into Oracle Audit Vault.

Although the concept of file system auditing is not new, Cloud FS auditing delivers the following set of advanced capabilities:

- Cloud FS auditing provides the same functionality and management interfaces across all of the platforms that Cloud FS supports.
- Cloud FS auditing enforces separation of duties for audit trail management and review.
- Cloud FS auditing integrates with Oracle Audit Vault, which provides a secure offline store for audit information, along with analysis and reporting capabilities.

# Cloud FS Audit Trail Files

- Audit files are located at:  
`<Mount Point>/ .Security/audit/acfs-audit-<FSID>-<Hostname>.log`
- Audit files are secured by using permissions that enforce separation of duties.
  - Audit managers can manage the audit trail.
  - Auditors can view but cannot manage.
  - Neither role can truncate, overwrite, or delete the audit trail.
- When audit files are full, they are automatically archived.
  - Files that are not full can be manually archived.
- Archive files are located at:  
`<Mount Point>/ .Security/audit/acfs-audit-<FSID>-<Hostname>.log.bak`
- Active audit files should not be interrogated.
  - Archive first and then interrogate the archive.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Cloud FS audit trail is contained in a set of files inside Cloud FS. Audit trail files generated by Cloud FS auditing are designed for:

- Manual review by a Cloud FS auditor using text-based viewing tools
- Integration into Oracle Audit Vault
- Integration into third-party products that can parse and import the audit data

After auditing is enabled, audit files are written under `<Mount Point>/ .Security/audit`. Each host in the Cloud FS cluster writes to a separate audit file. This avoids potential complications that are associated with multiple hosts attempting to use the same file.

The audit files are secured by using permissions that enforce separation of duties. That is, audit managers can perform management functions, such as archiving the audit files, but cannot mark the archived files as read. Auditors can view the contents of audit files and mark them as read, but they cannot perform management functions. Note that audit managers and auditors cannot truncate, overwrite, or delete the audit trail.

When audit files reach 10 MB, they are considered full and are automatically archived. When a file is archived, it is closed and `.bak` is appended to the file name. The next audit record is written to a new audit file, enabling auditing to continue without interruption. Files that are not full can be manually archived by using the `acfsutil audit archive` command.

Note that active audit files should not be interrogated because it could interrupt auditing or result in the loss of auditing data. Rather, an archive should be created and interrogated instead of the active audit file.

# Cloud FS Audit Trail Contents

Sample audit file:

Header
ACFS Audit Version: 1.0 Encoding: UTF-8
Event: ACFS_CMDRULE_WRITE Description: A user attempted to write to a realm protected file. Product: ACFS_SECURITY Timestamp: 2/21/2012 08:23:01 UTC User: 102 Group: 102 Process: 4567 Host: host1 File: /my_mount_point1/hr_data/payroll Evaluation Result: ACFS_REALM_AUTH Realm: myPayrollRealm Application: vi
File Access Event
Event: ACFS_SEC_PREPARE Description: A user prepared a device for ACFS Security. Product: ACFS SECURITY Timestamp: 2/23/2012 09:14:10 UTC User: 1042 Group: 1823 Process: 8901 Host: host1 Command Line: acfsutil sec prepare -m /my_mount_point2
Privilege Use Event

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Cloud FS audit trail consists of a set of audit records. Each audit record represents a single event.

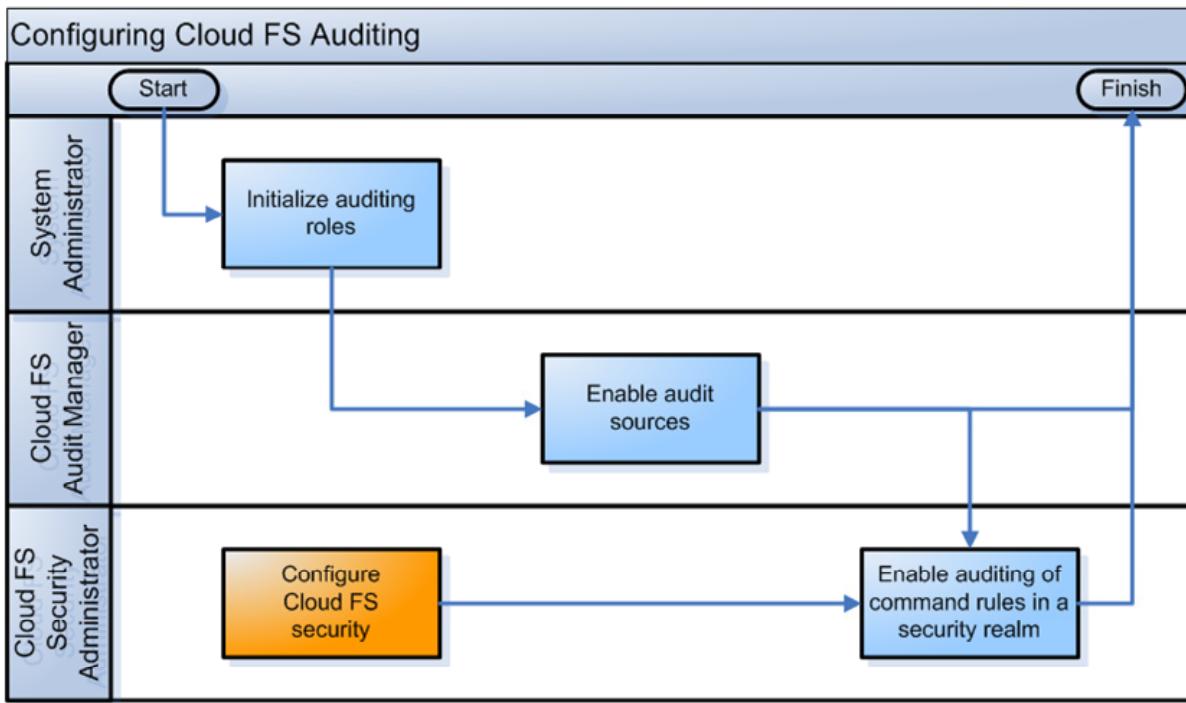
The audit trail has a brief header at the beginning of each file. The header identifies the version of the audit file format and the character encoding for the audit file.

Following the header, the audit file consists of audit records. There are several different types of audit records, each of which represents a unique type of event and contains different information that is relevant to the event. The types of events are:

- File access events
- Privilege use events
- Authentication failures or insufficient privileges events

Each record is written to the audit trail as a set of field names and values. Each field and value pair is separated by a colon and followed by an end-of-line character. The combination of audit record fields entered in the audit trail depends on the event type. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for a complete listing of all the audit events and audit record fields.

# Configuring Cloud FS Auditing



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide illustrates the process for configuring Cloud FS auditing. It outlines the key process tasks and who is required to perform them.

Note that the task of configuring Cloud FS security is separate from configuring Cloud FS auditing, but is mentioned here because it is an important prerequisite. Note also that the diagram does not mention other prerequisite tasks, such as initially configuring Cloud FS and creating ACFS file systems.

Further details regarding the steps for configuring Cloud FS auditing are provided on the following pages.

# Initializing Auditing Roles and Enabling Audit Sources

- Set up required roles for auditing.
  - Required task
  - Performed by the system administrator
  - Command syntax:

```
# acfsutil audit init -M <Audit Manager Group> -A <Auditor Group>
```
  - Groups cannot be changed after initialization.
- Enable auditing on a specified file system.
  - Required task
  - Performed by a Cloud FS audit manager
  - Command syntax:

```
$ acfsutil audit enable -m <Mount Point> -s [sec | encr]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The acfsutil audit init command must be run by the system administrator. The command sets up the required roles for auditing and must be run before any type of auditing can be enabled on a file system. After initialization, you cannot choose a different OS group for either the Cloud FS audit manager or Cloud FS auditor. Because of this, it is recommended that specific OS groups should be created for these roles.

The acfsutil audit enable command enables auditing on a specified file system. Only an audit manager can run this command. In addition to specifying the file system, the audit manager must specify whether to enable auditing for Cloud FS security (-s sec) or Cloud FS encryption (-s encr). Auditing can be enabled for both Cloud FS security and Cloud FS encryption by running the acfsutil audit enable command twice.

# Enabling Auditing of Command Rules in a Security Realm

Enable auditing of specific command rules in a Cloud FS security realm.

- Optional task
- Performed by a Cloud FS security administrator
- Command syntax:

```
$ acfsutil sec realm audit enable <Realm> -m <Mount Point>  
[-l <Command Rule>,<Command Rule>,....] [-a] [-v [-u]]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

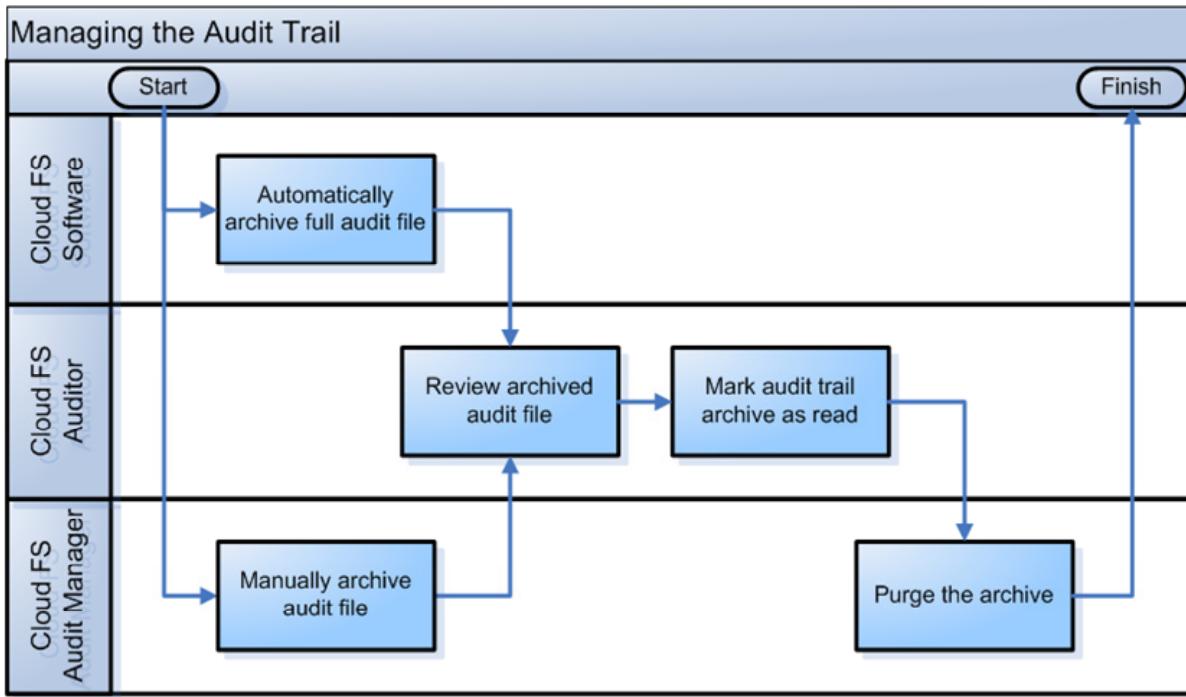
In addition to the core auditing that is enabled by the `acfsutil audit enable` command, auditing of specific command rules in a Cloud FS security realm can be enabled by using the `acfsutil sec realm audit enable` command. Note that only the security administrator, not the audit manager, can run this command. Following is a description of the options that are available with the `acfsutil sec realm audit enable` command:

<Realm>	Specifies the security realm name
-m <Mount Point>	Specifies the directory where the file system is mounted
-l <Command Rule>	Specifies the command rules that are audited. If it is not specified, all command rules associated with the realm are audited
-a	Specifies audit realm authorizations
-v [-u]	Specifies audit realm violations. If <code>-u</code> is specified, only realm violations by users who are members of a realm are audited

Auditing of command rules in a security realm builds on the realm-based security capabilities that are already present in earlier releases of ACFS. For more information regarding Cloud FS security, including realms and command rules, refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)*.

Note that Cloud FS security realms and command rules must be configured before auditing of command rules in a security realm can be enabled.

# Managing the Audit Trail



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide illustrates the process for managing the Cloud FS audit trail in the absence of Oracle Audit Vault. It outlines the key process tasks and who performs them.

The diagram represents one iteration of what is an endless loop as archive files fill up over and over again.

If Oracle Audit Vault is implemented, the Oracle Audit Vault collector automatically consumes archived files and marks them as read.

Further details regarding the steps for managing the audit trail are provided on the following pages.

## Archiving Audit Files

- Audit files are automatically archived when they reach the predefined maximum size of 10 MB.
- Audit files can also be manually archived by a Cloud FS audit manager.
  - Enables immediate review of recent audit data
  - Command syntax:

```
$ acfsutil audit archive -m <Mount Point>
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Audit files are automatically archived when they reach the predefined maximum size of 10 MB. In addition, Cloud FS audit managers can manually archive audit files by using the `acfsutil audit archive` command. This enables immediate review of recent audit data in the archive while the system continues to record audit data in the active file.

## Reviewing Audit Files

- Without Oracle Audit Vault:
  - Auditors can review archived audit files by using any tools.
    - Can back up the archive or copy audit data into another file, if necessary
  - Archived audit files should be marked as read when they are no longer required.
    - Indicates that it is safe to purge the archived files
    - Command syntax:

```
$ acfsutil audit read -m <Mount Point>
```
- With Oracle Audit Vault:
  - Archived audit files are automatically imported into Oracle Audit Vault.
    - Automatically marked as read after successful import
  - Auditors should use Audit Vault tools to review the audit trail.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, auditors can review the audit trail by using any tools to read or search against the archived audit files. Auditors are free to back up the archive or copy audit data into another file, if necessary. Remember that active audit files should not be interrogated because it could interrupt auditing or result in the loss of audit data.

When they are no longer required, archived audit files should be marked as read to indicate that it is safe to purge them. Use the `acfsutil audit read` command to mark the files as read.

If Oracle Audit Vault has been implemented, archived audit files are automatically imported into Oracle Audit Vault. After they are successfully imported, the archived audit files are automatically marked as read. In this case, auditors should use the Audit Vault tools to analyze and review the audit trail, rather than accessing the audit files directly.

## Purging Audit Files

Purging removes archived audit files that have been marked as reviewed.

- Must be performed by an audit manager
- Is important because you cannot archive the current audit file before the previous archive is purged
- Command syntax:

```
$ acfsutil audit purge -m <Mount Point> [-f]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Purging removes archived audit files that have been marked as reviewed.

Purging is important because you cannot archive the current audit file before the previous archive is purged.

Purging is performed by the audit manager by using the `acfsutil audit purge` command. This command has one optional argument (`-f`) that forces purging even if the auditor has not marked the archived audit files as read. Forced purging can result in the loss of audit data and should generally not be performed.

## Cloud FS Plug-in Infrastructure: Overview

The Cloud FS plug-in infrastructure enables user applications to access file system and volume metrics.

- Allows for customized monitoring solutions that include ACFS file system and volume data
- Can be enabled on individual file systems and hosts
- Referenced by applications using the Cloud FS plug-in API
  - The API supports message posting and polling delivery models.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c, Cloud FS provides infrastructure that enables a user space application to access just-in-time ACFS file system and ADVM volume metrics. Applications can use the Cloud FS plug-in infrastructure to create customized solutions that extend the general application file metric interfaces to include detailed file system and volume data.

The Cloud FS plug-in can be enabled on individual ACFS file systems mounted on a stand-alone host or on one or more nodes of a cluster where the Oracle ACFS file system is mounted. This enables message communication between an ACFS file system and an associated user space application module by using the Cloud FS plug-in application programming interface (API).

The plug-in API supports both polling and posting message delivery models and multiple message payload types.

Essential documentation for the API can be found in the header file at \$ORACLE\_HOME/usm/public/acfslib.h. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for detailed information regarding the plug-in infrastructure and API.

# Using the Cloud FS Plug-in

- Enabling the plug-in:

```
$ acfsutil plugin enable [<tag> ...] metricstype [<interval>[s|m]]<mount_point>  
$ acfsutil plugin enable HRDATA acfsmetric1 /hr  
$ acfsutil plugin enable acfsmetric1 5m /sales
```

- Referencing the metrics:

```
#include <usacfslib.h>  
  
/* allocate message buffers */  
ACFS_METRIC1 *metrics = malloc (sizeof(ACFS_METRIC1));  
  
/* poll for metric1 data */  
rc = acfsplugin_metrics(ACFS_METRIC_TYPE1, metrics,  
sizeof(metrics), mountp);  
  
/* print message data */  
printf("reads %8llu ", metrics->acfs_nreads);  
printf("writes %8llu ", metrics->acfs_nwrites);  
printf("avg read size %8u ", metrics->acfs_avgrsize);
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To use the Cloud FS plug-in API, the plug-in must first be enabled. The slide shows the general syntax for the `acfsutil plugin enable` command, followed by two examples.

In the first example, the plug-in is enabled for all files in the `/hr` file system that are tagged with the `HRDATA` tag. In this case, no interval is specified, which implies that the metrics will be referenced by using the polling delivery model.

In the second example, the plug-in is enabled for all files in the `/sales` file system, regardless of tagging. In this case, a five-minute interval is also specified, which implies that the metrics will be referenced by using message posting as the delivery model.

In both examples, `acfsmetric1` is specified as the metric type. This predefined metric type contains a collection of metrics, including number of reads, number of writes, average read size, average write size, minimum and maximum read size, minimum and maximum write size, read throughput (bytes per second) and write throughput.

After the plug-in is enabled, user applications can reference the exposed metrics. The example at the bottom of the slide shows some code fragments. The `acfsplugin_metrics` function reads the metrics associated with the file system specified in the `mountp` argument. If the plug-in was enabled without an interval, the function call polls for the latest metrics. If the plug-in was enabled with an interval, the call to `acfsplugin_metrics` is blocked, and the application pauses until the metrics are next posted.

## Using Cloud FS Replication in Conjunction with Cloud FS Security and Encryption

With Oracle Database 12c, Cloud FS replication can be used in conjunction with Cloud FS security and encryption.

- This capability enables:
  - Replication of realm-secured file systems
  - Replication of encrypted file systems
  - Realm security to be configured on an existing replicated file system
  - Encryption to be configured on an existing replicated file system
- The replicated file system inherits the security policies and encryption settings from the primary file system.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With Oracle Database 12c, Cloud FS replication can be used in conjunction with Cloud FS security and encryption. This new capability essentially lifts the previous restriction that disallowed replication of secured or encrypted file systems. With this enhancement, file systems configured with security or encryption (or both) can be replicated. Likewise, existing replicated file systems can be configured to implement security or encryption (or both).

When replication is used in conjunction with security or encryption, the replicated file system inherits the security policies or encryption settings from the primary file system. Otherwise, the configuration and management of replication, security, and encryption is not altered.

## Generic API for Cloud FS Tagging

Cloud FS provides an API for Cloud FS tagging.

- The API complements existing command line interfaces.
- The API is generic and platform independent.
  - Implemented as a C library
- The API provides the following operations:
  - `acfsgettag` Determines whether a file contains a tag
  - `acfslisttags` Lists tags that are assigned to a file
  - `acfsremovetag` Removes a tag from a file
  - `acfssettag` Adds a tag to a file



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c provides a generic, platform-independent API to support tagging operations on files. The API complements the existing `acfsutil tag` commands that are available in previous releases. The API is implemented as a C library and made available as part of the standard Oracle Database 12c software distribution.

The slide outlines the operations that are provided by the API. Essential documentation for the API can be found in the header file at `$ORACLE_HOME/usm/public/acfslib.h`. A demonstration application using the tagging API is also available at `$ORACLE_HOME/usm/demo`. Refer to *Oracle Automatic Storage Management Administrator's Guide, 12c Release 1 (12.1)* for further information.

# Cloud FS Resource Enhancements

- File system resource enhancements

```
$ srvctl add filesystem -device <vol_device> -mountpointpath <mount_path>
[-volume <vol_name>] [-diskgroup <dg_name>] [-user <user>]
[-nodes <node_list> | -serverpools <serverpool_list>]
[-fstype {ACFS|NTFS|ZFS|JFS|EXT3|EXT4}] [-fsoptions <options>]
[-description <description>] [-appid <application_id>]
[-autostart {ALWAYS|NEVER|RESTORE}]
```

- New ADVM resource
  - Completes the storage resource dependency tree
  - Ensures that resources are started and stopped in the right order
- ACFS mount registry resource removed
  - All file system attributes are in the file system resource.
- Consistent file system classification
  - No difference between general file systems and file systems that contain Oracle Database home directories



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c provides the following enhancements to cluster resources for Cloud FS:

- File System Resource Enhancements

The file system resource has been enhanced through the provision of extra attributes. The file system resource enhancements provide administrators with better control over where file systems are mounted, what mount options are used, and whether the file system should be mounted automatically. The full form of the `srvctl add filesystem` command is shown in the slide. The new attributes, highlighted in red, are:

-nodes	Comma-separated list of nodes on which the file system will be mounted (The default is all nodes.)
-serverpools	Comma-separated list of server pools on which the file system will be mounted (The default is all server pools.)
-fstype	File system type
-fsoptions	Comma-separated list of file system mount options
-description	File system description
-appid	File system application ID
-autostart	Policy for automatically starting the file system

- New ADVM Resource

A new resource type is included for ADVM resources. The new ADVM resource completes the storage resource dependency tree that includes ASM, ADVM, and ACFS. It allows more precise control over resource dependencies to ensure that resources are managed correctly, including the correct order for resource startup and shutdown. The ADVM resource is created automatically when a volume is created, and it contains no adjustable attributes settings. The current status of volume resources can be determined by using the `srvctl status volume` or `crsctl status resource` commands.

- ACFS Mount Registry Resource Removed

In previous releases, a Cluster Ready Services (CRS) resource was associated with the ACFS mount registry. This resource was primarily used to ensure that file systems were automatically mounted after a system restart. In addition, CRS resources were also associated with ACFS file systems designated as Oracle Database home file systems. Using the file system resource enhancements provided in Oracle Database 12c, all file system attributes previously maintained in the ACFS mount registry can be specified in the ACFS file system resource, and the ACFS mount registry resource is no longer required.

All of the ACFS registry interfaces and functions are preserved in Oracle Database 12c; however, the file system attributes are stored in the ACFS file system resource that is associated with each file system.

- Consistent File System Classification

In previous releases, an ACFS file system could be configured as a general file system or as an Oracle Database home file system. With Oracle Database 12c, there is no difference between general file systems and file systems that contain Oracle Database home directories. That is, any ACFS file system can house Oracle Database home directories and other data files, and no additional configuration is required to enable storage of Oracle Database home directories.

# Implementing Node-Specific File System Dependencies

- Use case: A clustered application needs to record log file information separately for each node.
- Implementation example:

```
$ srvctl add filesystem -device /dev/asm/log1-123  
  -mountpointpath /mnt/login01 -appid LOGFS -node c00n01  
$ srvctl add filesystem -device /dev/asm/log2-123  
  -mountpointpath /mnt/login02 -appid LOGFS -node c00n02  
  
$ crsctl status type | grep LOGFS  
TYPE_NAME=ora.LOGFS_fs.type  
  
$ crsctl modify resource my_application  
  -attr "START_DEPENDENCIES=hard(type:ora.LOGFS_fs.type)  
        pullup(type:ora.LOGFS_fs.type)"
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The new `appid` file system resource attribute can be used to define dependencies between a clustered application and separate node-specific file systems running on each cluster node. A common use case occurs when a clustered application needs to record log file information separately for the application instances running on each node.

The slide shows an implementation example based on a two-node cluster (`c00n01` and `c00n02`). The example assumes that two volumes (`/dev/asm/log1-123` and `/dev/asm/log2-123`) have already been formatted with ACFS.

The `srvctl add filesystem` commands create separate node-specific file system resources. Note that `-appid LOGFS` is specified in both commands.

Setting the `appid` file system resource attribute results in the creation of a type containing the `appid` in the type name. The `crsctl status type` command can be used to identify the complete type name, which is `ora.LOGFS_fs.type` in this example.

Finally, the type name can be used in a dependency definition associated with an application requiring the file systems. In this example, a cluster resource named `my_application` is modified to depend on the file systems associated with the `ora.LOGFS_fs.type` type.

As a result of this configuration, when `my_application` starts on either cluster node, the corresponding file system will also be mounted (`/dev/asm/log1-123` on `c00n01` and `/dev/asm/log2-123` on `c00n02`).

## Enhanced Platform Support for Cloud FS Data Services

	<b>Snapshot</b>	<b>Replication</b>	<b>Tagging</b>	<b>Security</b>	<b>Encryption</b>
<b>11.2.0.1</b>	Read-only on Linux and Windows	-	-	-	-
<b>11.2.0.2</b>	Read-only on Linux, Windows, Solaris, AIX	Linux	Linux	Linux	Linux
<b>11.2.0.3</b>	Read-write on Linux, Windows, Solaris, AIX	Linux, Windows	Linux, Windows	Linux, Windows	Linux, Windows
<b>12c</b>	Read-write on Linux, Windows, Solaris, AIX	Linux, Windows, Solaris, AIX	Linux, Windows, Solaris, AIX	Linux, Windows, Solaris	Linux, Windows, Solaris



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The table in the slide summarizes the platform support for the different Cloud FS data service capabilities. The platform-specific enhancements for Oracle Database 12c are highlighted. No new functionality is introduced; existing features are available on a wider selection of platforms.

## Miscellaneous Cloud FS Enhancements

- Support for cluster-wide, file-granular advisory file locking by using the POSIX `fcntl` interface
  - Byte-range locks continue to operate in node local mode.
- End-to-end storage visibility for files
  - `$ acfsutil info file -d <filename>`
  - Displays detailed file information, including file extent location on the Oracle ASM devices
- Support for ordered-write and deferred-write mount options on Linux
- Improved directory listing performance for newly created directories
- Unicode support for file names on Windows
  - Unicode file names are already supported on other platforms.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide lists other miscellaneous Cloud FS new features in Oracle Database 12c.

## Quiz

Identify the correct statements regarding Cloud FS support for storing Oracle Database files:

- a. With Oracle Database 12c, Cloud FS is supported as a store for all Oracle Database files.
- b. A Cloud FS snapshot can be used as a point-in-time image for backups while database processing continues uninterrupted.
- c. Cloud FS replication can be used to replicate database files for disaster recovery purposes.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: a, b

With Oracle Database 12c, support will not be available for Oracle Database files on file systems that use Cloud FS replication. Oracle Data Guard remains the recommended Oracle Database replication technology for disaster protection.

## Quiz

Identify the correct statement regarding the configuration process for Cloud FS auditing:

- a. The system administrator initializes auditing, the audit manager enables auditing, and the Cloud FS security administrator enables auditing of command rules in a security realm.
- b. The system administrator initializes auditing, and the audit manager enables auditing and auditing of command rules in a security realm.
- c. The audit manager initializes and enables auditing, and the Cloud FS security administrator enables auditing of command rules in a security realm.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: a**

## Quiz

Auditors should examine the active audit files to see the most current audit information.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

### Answer: b

Active audit files should not be interrogated, because this could interrupt auditing or result in the loss of audit data. To view recent audit information, the active file should first be archived by the audit manager. Then, the auditor can examine the archived audit file.

## Quiz

Identify the correct statements regarding Cloud FS file system resources:

- a. A file system resource can specify a list of nodes on which the file system will be mounted.
- b. A file system resource can specify a list of server pools on which the file system will be mounted.
- c. All file system attributes are recorded in the file system resource, removing the need for the ACFS mount registry.
- d. Any ACFS file system can house Oracle Database home directories and other data files, and no additional configuration is required to enable storage of Oracle Database home directories.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: a, b, c, d**

Cloud FS file system resources display all of the attributes listed in the question.

## Summary

In this lesson, you should have learned how to:

- Describe each of the Cloud FS new features
- Configure each new feature
- Administer each new feature



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 8 Overview:

- Practice 8-1: Configuring and Using HANFS
  - In this practice, you will configure and use HANFS. You will also crash the node running the HANFS service and watch it migrate to a surviving note.
- Practice 8-2: Using Cloud FS to Store Oracle Database Files
  - In this practice, you will use DBCA to create a new Oracle database that uses Cloud FS to store all its data files, control files, log files, etc. You will then use snapshots, including the new snaps-of-snaps feature, to perform a point-in-time backup of the database and quickly revert to the snapshot copy.
- Practice 8-3: Configuring and Using Cloud FS Auditing
  - In this practice, you will go through the process of configuring Cloud FS auditing. After configuration, you will interact with Cloud FS to generate audit records and exercise the audit trail management procedure.
- Practice 8-4: Implementing Node-Specific File System Dependencies
  - In this practice, you will use the enhanced file system resource definitions to implement node-specific file system dependencies so that application resources on different nodes have access to separate node-specific file systems.



Copyright © 2013, 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

# 9

## Application Continuity

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

- Describe the purpose of Transaction Guard and Application Continuity
- Describe the key concepts relating to Application Continuity
- Describe the side effects and restrictions relating to Application Continuity
- Describe the requirements for developing applications that leverage Application Continuity
- Configure Application Continuity



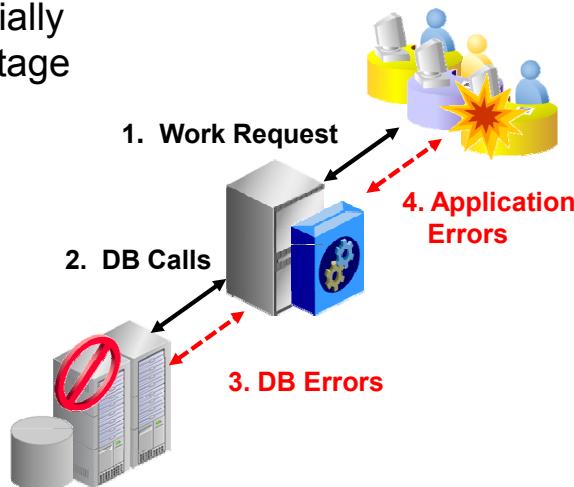
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## The Situation Prior to Application Continuity

- Database outages can cause:
  - In-flight work to be lost, leaving users in doubt
  - Users to restart applications and reenter data, leading to duplicate submission of requests
  - Additional failures, potentially prolonging the system outage
- Solving the problem inside the application is difficult and expensive.



**Did the last transaction commit?**



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For end users, database session outages can lead to undesirable results:

- Confusion: Users do not know what happened to their application's transactions, such as funds transfers, orders, payments, and bookings.
- Duplication: Doubt over a failed transaction may lead users to reenter it. This may lead to undesirable duplication resulting in overpayments, over ordering or overbooking, for example.
- Loss of productivity: Even if duplication is not an issue, user productivity is adversely affected by the need to restart applications and reenter data.
- Prolonged system outages: A database failure may cause related applications or other infrastructure, like sensors and communication equipment, to stall or fail. This may require applications and other components to be rebooted or reinitialized resulting in a prolonged system outage.

Developing a solution for these problems inside the application has traditionally been difficult and expensive for the following reasons:

- Every possible exception must be considered and handled.
- If a failure occurs during a commit, it is difficult to determine whether the commit occurred.
- To legitimately replay the work associated with a failed session, the application must ensure that the database is in the correct state; otherwise, the replay may be invalid.

# Introducing Transaction Guard and Application Continuity

- Transaction Guard is a new reliable protocol and API that returns the outcome of the last transaction after a recoverable error has occurred.
- Application Continuity is a feature that attempts to mask database session outages by recovering the in-flight work for requests submitted to the database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c introduces two fundamental capabilities for ensuring continuity of applications after database outages:

1. A foolproof way for applications to know the outcome of transactions
2. The ability to mask outages by reconnecting to the database and replaying the workload

These capabilities are provided by two new features: Transaction Guard and Application Continuity.

Transaction Guard is an API that applications use in error handling. It is a new and reliable way to return the outcome of the last transaction after a recoverable error has occurred. By itself, Transaction Guard can dramatically improve the end-user experience by erasing the doubt over whether the last transaction was committed or not.

Application Continuity is a feature that masks recoverable outages from end users and applications. Application Continuity attempts to replay the transactional and nontransactional work that constitutes a database request. When replay is successful, the outage appears to the end user as if the execution was slightly delayed. With Application Continuity, the end user experience is improved because users may never sense that an outage has occurred. Furthermore, Application Continuity can simplify application development by removing the burden of dealing with recoverable outages.

# Key Concepts for Application Continuity



**Database Request:** Unit of work submitted from the application, typically SQL, PL/SQL, RPC calls



**Recoverable Error:** Error that arises due to an external system failure, independent of the application



**Commit Outcome:** Outcome of last transaction; made durable by Transaction Guard



**Mutable Functions:** Functions that change their results each time they are executed



**Session State Consistency:** Describes how the application changes the nontransactional state during a database request



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide introduces some terms and concepts that are associated with Transaction Guard and Application Continuity. The following notes elaborate further:

## Database Request

A database request is a unit of work submitted from the application. A database request typically consists of the SQL statements, PL/SQL blocks, local procedure calls, and database RPCs (Remote Procedure Calls), in a single web request on a single database connection. Database requests are generally demarcated by the calls made to check out and check in a database connection from a connection pool. For recoverable errors, Application Continuity reestablishes the database session and repeats the database request safely.

## Recoverable Error

A recoverable error is an error that arises due to an external system failure, independent of the application session logic that is executing. Recoverable errors occur following planned and unplanned outages of foregrounds, networks, nodes, storage, and databases. The application receives an error code (such as ORA-03113 : end-of-file on communication channel) that can leave the application not knowing the status of the last operation submitted. Recoverable errors have been enhanced in Oracle Database 12c to include more errors, and now include a public API for OCI.

Application Continuity reestablishes database sessions and resubmits the pending work for recoverable errors. Application Continuity does not resubmit work following call failures due to nonrecoverable errors. An example of a nonrecoverable error is submission of an invalid data value, such as an invalid date or invalid numeric value.

### Commit Outcome

In Oracle Database, a transaction is committed by updating its entry in the internal transaction table. Oracle Database generates a redo log record corresponding to this update. After the redo log record is written out to the redo log on disk, the transaction is deemed committed at the database. From a client perspective, the transaction is committed when a commit outcome message, generated after that redo is written, is received by the client. However, the commit outcome message is not durable. For example, the commit outcome can be lost if the client is disconnected from the database after the commit outcome is generated by the database but before it is received by the client. Transaction Guard provides a reliable commit outcome because it can be used to reliably obtain the commit outcome when it has been lost following a recoverable error that results in the failure of a database session.

### Mutable Functions

Mutable functions are functions that can change their results each time that they are called. Mutable functions can cause replay to be rejected because the results visible to the application change at replay. Consider `sequence.NEXTVAL` that is often used in key values. If a primary key is built with a sequence value and it is later used in foreign keys or other binds, at replay the same function result must be returned if the application is using it.

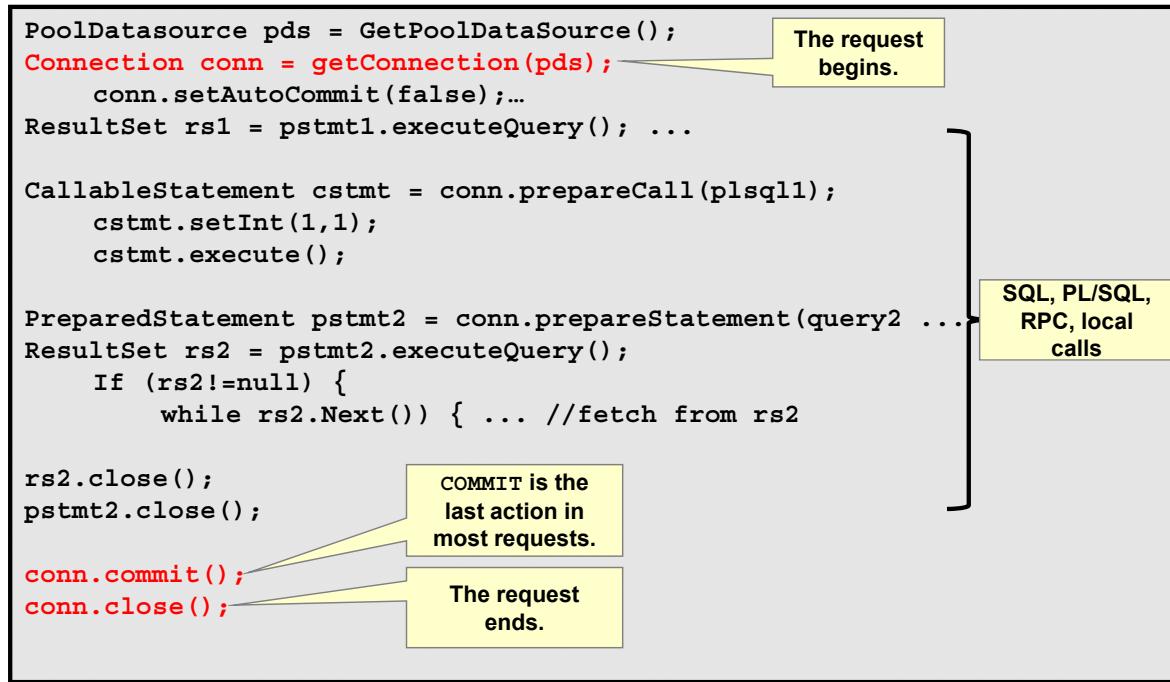
Application Continuity provides mutable object value replacement at replay for granted Oracle function calls to provide opaque bind-variable consistency. If the call uses database functions that are mutable, including `sequence.NEXTVAL`, `SYSDATE`, `SYSTIMESTAMP`, and `SYSGUID`, the original values returned from the function execution are saved and are reapplied at replay. If an application decides not to grant mutables, replay for these requests may be rejected.

### Session State Consistency

Nontransactional state includes national language support (NLS) settings, cursors, events, and global PL/SQL package states. After a `COMMIT` statement has been executed, if the nontransactional state was changed in that transaction, it is not possible to replay the transaction to reestablish that state if the session is lost. When configuring Application Continuity, the applications are categorized depending on whether the session state after the initial setup is static or dynamic, and thus whether it is correct to continue past a `COMMIT` operation within a request.

While configuring Application Continuity, almost all applications should use the default `DYNAMIC` mode. In the `DYNAMIC` mode, replay is disabled from `COMMIT` until the end of the request. This is not a problem for most applications, because almost all requests have zero or one commit, and commit is most often the last statement in a database request.

# Workflow of a Database Request



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A database request is a logical unit of work.

For example:

- Every action that the client driver receives from an Object Relational Manager (ORM)
- Every action that you submit at your automatic teller machine
- Every action that you submit at your browser while browsing, shopping, making bill payments, and so on

Typically, all database requests that use JDBC follow a standard pattern.

The slide contains a code segment that shows how typical JDBC applications are coded:

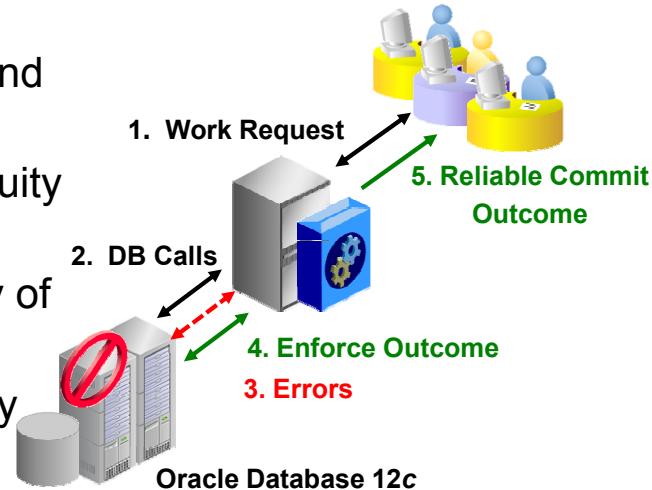
1. A database request begins with the call to `getConnection` to the `PoolDataSource`.
2. The application's logic is executed. This execution could include SQL, PL/SQL, RPC, or local procedure calls.
3. The transaction is committed.
4. The database request ends when the connection is returned to the connection pool.

# What Is Transaction Guard?

Transaction Guard is:

- A tool that provides a reliable commit outcome for the last transaction after errors
- An API available for JDBC Thin, C/C++ (OCI/OCCI), and ODP.NET
- Used by Application Continuity for at-most-once execution
- Can be used independently of Application Continuity

Without Transaction Guard, retry can cause logical corruption.



ORACLE

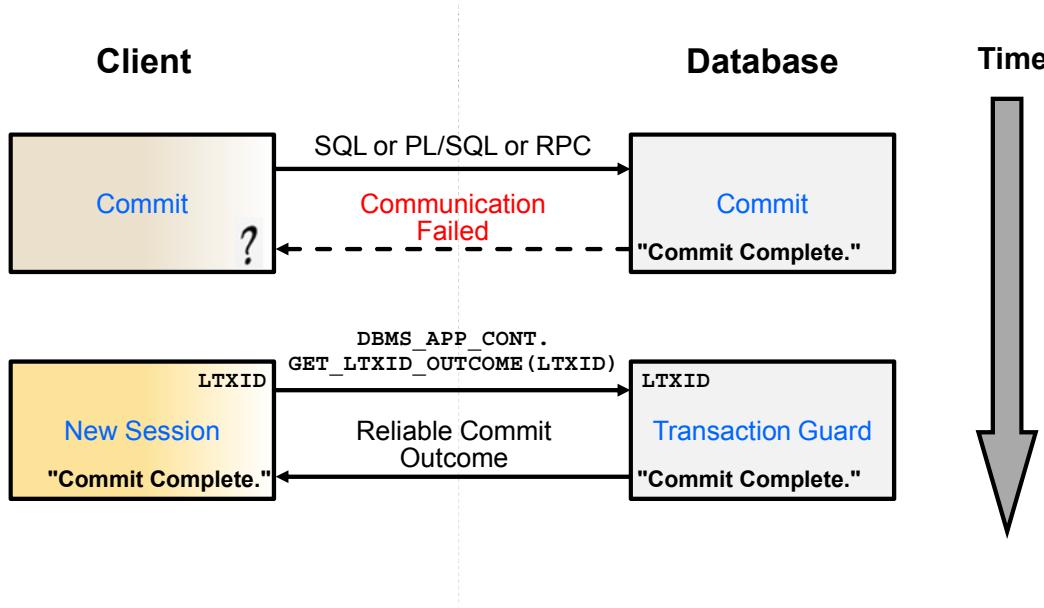
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Transaction Guard is a reliable protocol and API that applications use to provide a reliable commit outcome. The API is typically embedded in error handling and should be called following recoverable errors. The outcome indicates whether the last transaction was committed and completed. After the commit outcome is returned to the application, the outcome persists. Therefore, if Transaction Guard returns committed or uncommitted, the status stays this way. This enables the application or user to proceed with confidence.

Transaction Guard is used by Application Continuity and is automatically enabled by it, but it can also be enabled independently. Transaction Guard prevents the transaction being replayed by Application Continuity from being applied more than once. If the application has implemented application-level replay, integration with Transaction Guard can be used to ensure transaction idempotence; that is, executing the transaction multiple times has the same result as executing it only once.

# How Transaction Guard Works

Transaction Guard is a reliable protocol and API that enables applications to know the outcome of the last transaction.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the standard commit case, the database commits a transaction and returns a success message to the client. In the illustration shown in the slide, the client submits a commit statement and receives a message stating that communication failed. This type of failure can occur due to several reasons, including a database instance failure or network outage. In this scenario, without Transaction Guard, the client does not know the outcome of the transaction.

Oracle Database solves the problem by using a globally unique identifier called a logical transaction ID (LTXID). When the application is running, both the database and client hold the logical transaction ID. The database supplies the client with a logical transaction ID at authentication and at each round trip from the client driver that executes one or more commit operations.

When a recoverable outage occurs, the logical transaction ID uniquely identifies the last database transaction submitted on the session that failed. A new PL/SQL interface (DBMS\_APP\_CONT.GET\_LTXID\_OUTCOME) interface returns the reliable commit outcome. Further detail on using Transaction Guard APIs is provided later in the lesson.

# Using Transaction Guard

- Supported transaction types:
  - Local
  - Auto-commit and Commit on Success
  - Commit embedded in PL/SQL
  - DDL, DCL, and Parallel DDL
  - Remote, Distributed
- Not supported in release 12.1:
  - XA
  - Read-write database links from Active Data Guard
- Server configuration:
  - Set the `COMMIT_OUTCOME=TRUE` service attribute
  - Optionally, set the `RETENTION_TIMEOUT` service attribute
- Supported clients:
  - JDBC Thin, OCI, OCCI, and ODP.NET



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Transaction Guard supports all the transaction types listed in the slide. The primary exclusions in Oracle Database 12c release 12.1 are:

- Transaction Guard is not supported for applications developed by using Oracle XA.
- Transaction Guard is not supported if you are using Active Data Guard with read/write database links to another database.

To enable Transaction Guard, set the service attribute `COMMIT_OUTCOME=TRUE`. Optionally, change the `RETENTION_TIMEOUT` service attribute to specify the amount of time that the commit outcome is retained. The retention timeout value is specified in seconds; the default is 86400 (24 hours), and the maximum is 2592000 (30 days).

## Benefits of Transaction Guard

- After outages, users know what happened to their in-flight transactions, such as fund transfers, flight bookings, and bill payments.
- Transaction Guard provides better performance and reliability than home-built code for idempotence.



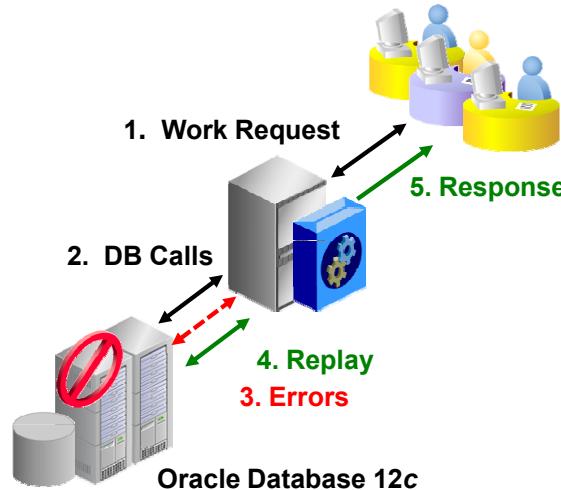
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Transaction Guard is a powerful feature. Some of the benefits are:

- For applications that integrate Transaction Guard, users can know what happened to their last submission and proceed with confidence and certainty. Without Transaction Guard, doubt following a failure can lead to resubmitting a database request, which can cause logical corruption.
- Because it is integrated into the database kernel, Transaction Guard provides better performance and reliability than home-built code for idempotence.

# What Is Application Continuity?

- Replays in-flight work on recoverable errors
- Masks many hardware, software, network, storage errors, and outages, when successful
- Improves the end-user experience



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Without Application Continuity, network outages, instance failures, hardware failures, repairs, configuration changes, patches, and so on can result in the failure of a user session followed by an error message of some sort.

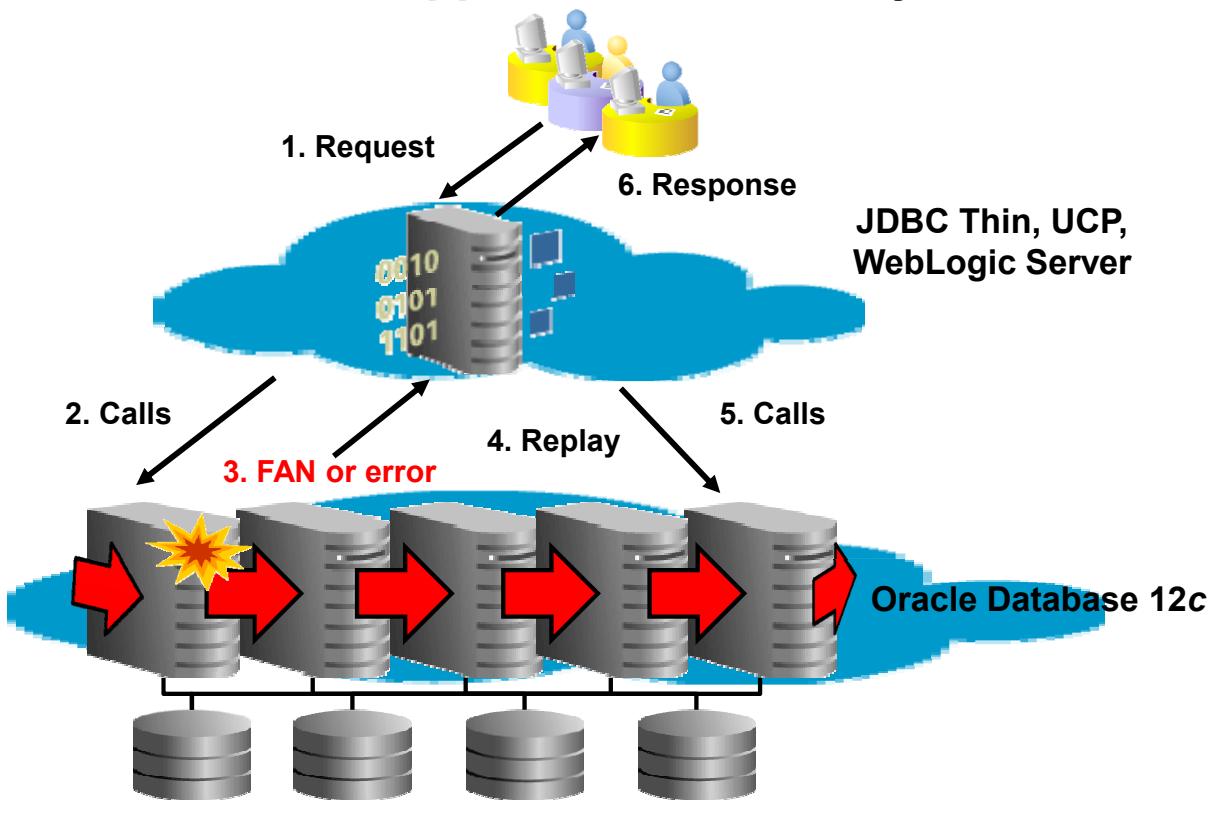
Application Continuity masks many recoverable database outages from applications and users. It achieves the masking by restoring the database session, the full session (including session state, cursors, and variables), and the last in-flight transaction (if there is one).

If the database session becomes unavailable due to a recoverable error, Application Continuity attempts to rebuild the session and any open transactions to the correct states. If the last transaction was successful and does not need to be reexecuted, the successful status is returned to the application. Otherwise, Application Continuity will replay the last in-flight transaction.

To be successful, the replay must return to the client exactly the same data that the client received previously in the original request. This ensures that any decisions based on previously queried data are honored during the replay. If the replay is successful, the application continues safely without duplication.

If the replay is not successful, the database rejects the replay and the application receives the original error. This ensures that the replay does not proceed if circumstances change between the original request and the replay.

## How Does Application Continuity Work?



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The graphic in the slide illustrates how Application Continuity works. Following is a description of a typical workflow involving Application Continuity:

1. The client sends a work request to the application.
2. The application sends the calls that make up the request to the database using the JDBC replay driver.
3. The JDBC replay driver receives a Fast Application Notification (FAN) notification or a recoverable error.
4. The replay driver performs the following actions:
  - It checks that the request has replay enabled and that the replay initiation timeout has not expired. If all is in order, the driver obtains a new database session. If a callback is registered, the callback is executed to initialize the session.
  - It checks with the database to determine whether the last transaction completed.
  - If replay is required, the JDBC replay driver resubmits calls, receiving directions for each call from the database. Each call must result in the same client-visible state.
5. When the last call is replayed, the replay driver ends the replay and returns to normal runtime mode.
6. If the replay succeeds, the application responds normally to the user.

# Using Application Continuity

- Supported database operations:
  - SQL, PL/SQL, and JDBC RPC:
    - SELECT, ALTER SESSION, DML, DDL, COMMIT, ROLLBACK, SAVEPOINT, and JDBC RPCs
  - Transaction models:
    - Local, Parallel, Remote, Distributed, and Embedded PL/SQL
  - Mutable functions
  - Transaction Guard
- Works in conjunction with:
  - Oracle RAC and RAC One
  - Oracle Active Data Guard
- Hardware acceleration on current Intel and SPARC chips
- Supported clients:
  - JDBC Thin driver, Universal Connection Pool, and WebLogic Server



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide lists key points relating to the use of Application Continuity. The following notes elaborate further:

- Application Continuity recovers the database request, including any in-flight transaction and the database session states. The requests may include most SQL and PL/SQL, RPCs, and local JDBC calls. Note that for remote and distributed transactions, all databases involved must be release 12.1 or later.
- Application Continuity offers the ability to keep the original values for some Oracle functions, such as `SEQUENCE.NEXTVAL`, that change their values each time that they are called. This improves the likelihood that the replay will succeed.
- Application Continuity uses Transaction Guard. Transaction Guard tags each database session with a logical transaction ID (LTXID), so that the database recognizes whether a request committed the transaction before the outage.
- Application Continuity works in conjunction with Oracle Real Application Clusters (Oracle RAC), RAC One, and Oracle Active Data Guard.
- On the database server, the validation performed by Application Continuity is accelerated using processor extensions built into current SPARC and Intel chips.
- Application Continuity provides client support for thin JDBC, Universal Connection Pool, and WebLogic Server.

# Application Continuity Processing Phases

Normal Run Time	Reconnect	Replay
<ul style="list-style-type: none"> <li>• Demarcates the database request</li> <li>• Builds proxy objects</li> <li>• Holds original calls with validation</li> <li>• Manages queues</li> </ul>	<ul style="list-style-type: none"> <li>• Ensures that the request has replay enabled</li> <li>• Handles timeouts</li> <li>• Creates a new connection</li> <li>• Validates the target database</li> <li>• Uses Transaction Guard to enforce last commit</li> </ul>	<ul style="list-style-type: none"> <li>• Replays held calls</li> <li>• Continues replay, if user-visible results match, based on validations</li> <li>• Continues the request</li> </ul>



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The operation of Application Continuity can be divided into three distinct phases:

- **Normal run time:** During normal run time, each new database request is tagged with a request beginning, either by checking out of the Universal Connection Pool or WebLogic Server Connection Pool, or by adding begin and end request markers to your own application or to your own Java connection pool at checkout and checkin. The JDBC replay driver and Oracle Database 12c collaborate so that calls in the database request are held in queues, together with validation received from the database. The JDBC Replay driver holds the calls until the end of the database request or until the replay is disabled. The JDBC replay driver is responsible for managing the queues and building proxy objects to maintain a record of nontransactional state, allowing objects to be replaced if replay is needed.

- **Reconnect:** The reconnect phase of Application Continuity is triggered when a recoverable outage occurs. In this phase, the request is checked to see whether replay is still enabled, and the replay initiation timeout is checked to ensure that it has not expired. If both checks are in order, a new connection to the database is obtained. Because the reconnection to the database can take some time, you may need to set FAILOVER\_DELAY and FAILOVER\_TIMEOUT to allow the service to be reestablished. After the driver has established a connection to the database, it checks whether the database is a valid target and whether the last transaction was committed successfully. Replay will not occur if the connection is to a logically different database or to the same database but transactions have been lost. For example, the database has been restored to a prior point in time. Application Continuity will not resubmit committed transactions. Idempotence is enforced using Transaction Guard.
- **Replay:** The replay phase starts when a new connection to the database is obtained. All calls held in the queues are replayed, and the request continues from the point where it failed. Replay is disabled if there are any user-visible changes in results for any request that is replayed.

# Restrictions

Global	Request	Target Database
<ul style="list-style-type: none"> <li>Do not use the default database service.</li> <li>Replay is not supported for an Oracle XA data source.</li> <li>Deprecated Java concrete classes are not supported.</li> </ul>	<ul style="list-style-type: none"> <li>For Java streams, replay is on “best effort” basis.</li> <li>Restricted calls: <ul style="list-style-type: none"> <li>ALTER SYSTEM</li> <li>ALTER DATABASE</li> </ul> </li> <li>Replay is not supported for Active Data Guard with read/write database links</li> </ul>	<ul style="list-style-type: none"> <li>Does not support: <ul style="list-style-type: none"> <li>Logical Standby</li> <li>GoldenGate</li> </ul> </li> </ul>



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide lists some restrictions and other considerations that apply to Application Continuity. There are basically three levels of restrictions for Application Continuity:

- Global:** The following restrictions prevent Application Continuity from being enabled or used on any request.
  - Do not use the default service corresponding to the DB\_NAME or DB\_UNIQUE\_NAME. The use of the database service is not recommended for high availability, because this service cannot be enabled or disabled, and cannot be relocated on Oracle RAC or switched over to Oracle Data Guard.
  - Replay is not supported for applications developed by using Oracle XA.
  - For applications using JDBC, there is no support for oracle.sql deprecated concrete classes like BLOB, CLOB, BFILE, OPAQUE, ARRAY, STRUCT, or ORADATA.
- Request:** The following restrictions disable Application Continuity for part of a request.
  - For JDBC stream arguments, replay is on a “best effort” basis. For example, if the application is using physical addresses, the address is no longer valid with the outage and cannot be repositioned.
  - Replay is disabled if the transaction executes the ALTER SYSTEM or ALTER DATABASE statement.
  - Replay is not supported if you are using Active Data Guard with read/write database links to another database.
- Target:** Application Continuity is not supported for logically different databases (Oracle Logical Standby and Oracle GoldenGate).

## Potential Side Effects

- Some applications may need to use the `disableReplay` API if there are any requests that should not be replayed.
- Examples of calls that create side effects are:
  - Autonomous transactions
  - `DBMS_ALERT` calls (email or other notifications)
  - `DBMS_FILE_TRANSFER` calls (copying files)
  - `DBMS_PIPE` and `RPC` calls (to external sources)
  - `UTL_FILE` calls (writing text files)
  - `UTL_HTTP` calls (making HTTP callouts)
  - `UTL_MAIL` calls (sending email)
  - `UTL_SMTP` calls (sending SMTP messages)
  - `UTL_TCP` calls (sending TCP messages)
  - `UTL_URL` calls (accessing URLs)



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Applications that use external actions should be reviewed to decide whether requests with side effects should be replayed or not.

Application Continuity replays SQL, PL/SQL, and RPCs. Application Continuity serves to rebuild the session as if the user submission were delayed. When a session is rebuilt, all states are rebuilt, including reexecuting statements that leave side effects. These side effects may be exactly what is required, such as writing a report, completing some auditing, or obtaining custom primary key ranges. However, the calls that are replayed might include some calls that should not be replayed. The application may want to take action to accommodate or mitigate the effects of the replay. Developers can elect to use the `disableReplay` API for requests that contain calls that they do not want to replay.

The slide shows some of the actions that create side effects, such as Autonomous transactions, email or other notifications using `DBMS_ALERT` calls, copying files using `DBMS_PIPE` and `RPC` calls, writing text files using `UTL_FILE` calls, making HTTP callouts using `UTL_HTTP` calls, sending email using `UTL_MAIL` calls, sending SMTP messages using `UTL_SMTP` calls, sending TCP messages using `UTL_TCP` calls, and accessing URLs using `UTL_URL` calls.

## Actions That Disable Application Continuity

Normal Run Time	Reconnect	Replay
Any calls in same request after: <ul style="list-style-type: none"> <li>• Successful commit in dynamic mode (the default)</li> <li>• A restricted call</li> <li>• disableReplay API call</li> </ul>	<ul style="list-style-type: none"> <li>• Error is not recoverable</li> <li>• Reconnection failure               <ul style="list-style-type: none"> <li>– Replay initiation timeout</li> <li>– Max connection tries</li> <li>– Max retries per incident</li> </ul> </li> <li>• Target database is not valid for replay</li> <li>• Last call committed, in dynamic mode</li> </ul>	<ul style="list-style-type: none"> <li>• Validation detects different results</li> </ul>



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If it is disabled, Application Continuity is ineffective until the next request. The following events will disable replay for the request:

- **Normal run time:** Capture happens on all original calls until a successful COMMIT (in dynamic mode) or unless it is disabled for the request.
- **Reconnect:** Replay does not occur if the error is not recoverable or if:
  - Timeouts have been exceeded
  - The target database is not the same or an ancestor
  - The request has committed
- **Replay:** The calls must return the same user-visible results that the application has previously seen and potentially used to make a decision. If the validation for the replayed call does not pass, replay is aborted and the original error is returned.

## When Is Application Continuity Transparent?

- Application Continuity is transparent for J2EE applications that:
  - Use standard JDBC and Oracle connection pools
  - Do not have external actions, or have external actions and correctness is preserved at replay
- For situations where Application Continuity is not transparent, the following changes are typically required:
  - Request boundaries are specified using the Application Continuity APIs.
  - The `disableReplay` API is used to selectively stop replay for calls that the application never wants to replay.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application Continuity is transparent (performed automatically) for J2EE applications that use standard JDBC and that use Oracle connection pools (UCP or WLS Active GridLink). For applications with external actions (for example, autonomous transactions or using UTL\_HTTP to issue a SOA call), Application Continuity is still transparent only if the application's correctness is preserved when these external actions are replayed after a failure.

For other scenarios in which Application Continuity is not transparent, the following infrastructure changes may be needed:

- If the connection pool or container does not use an Oracle connection pool, the application must use Application Continuity APIs to mark request boundaries. Request boundaries are needed to reclaim the memory used for holding calls, and to establish a point at which to resume recording following nonreplayable operations.
- If the application has requests that the application does not want repeated, the application can explicitly call an API to disable replay for those requests. Such calls are likely to be isolated to one or a few specialized pieces of application code.

## Benefits of Application Continuity

- Uninterrupted user service, when replay is successful
- Can help relocate database sessions to remaining servers for planned outages
- Improves developer productivity by masking outages that can be masked
- Few or no application changes
- Simple configuration



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here are some benefits of Application Continuity:

- User service is uninterrupted when the request replay is successful.
- Application Continuity can be used to migrate the database sessions to the remaining servers without the users perceiving an outage.
- Masking outages that can be masked improves developer productivity. Error handling code will be invoked less often and can potentially be simplified.
- Replay often requires few or no application changes.
- Application Continuity is simple to configure.

# Application Assessment for Using Application Continuity

Decide	What You Should Do
Request Boundaries	Mark request boundaries if you are not using Oracle Pools.
JDBC Concrete Classes	Replace deprecated concrete classes with Java interfaces.
Side Effects	Use the <code>disable</code> API if a database request has an external call that should not be replayed.
Callbacks	Ensure that a callback is registered if the state changes outside a request (WLS/UCP labeling included by default).
Mutable Functions	Grant keeping mutable values if they are compatible with your application.



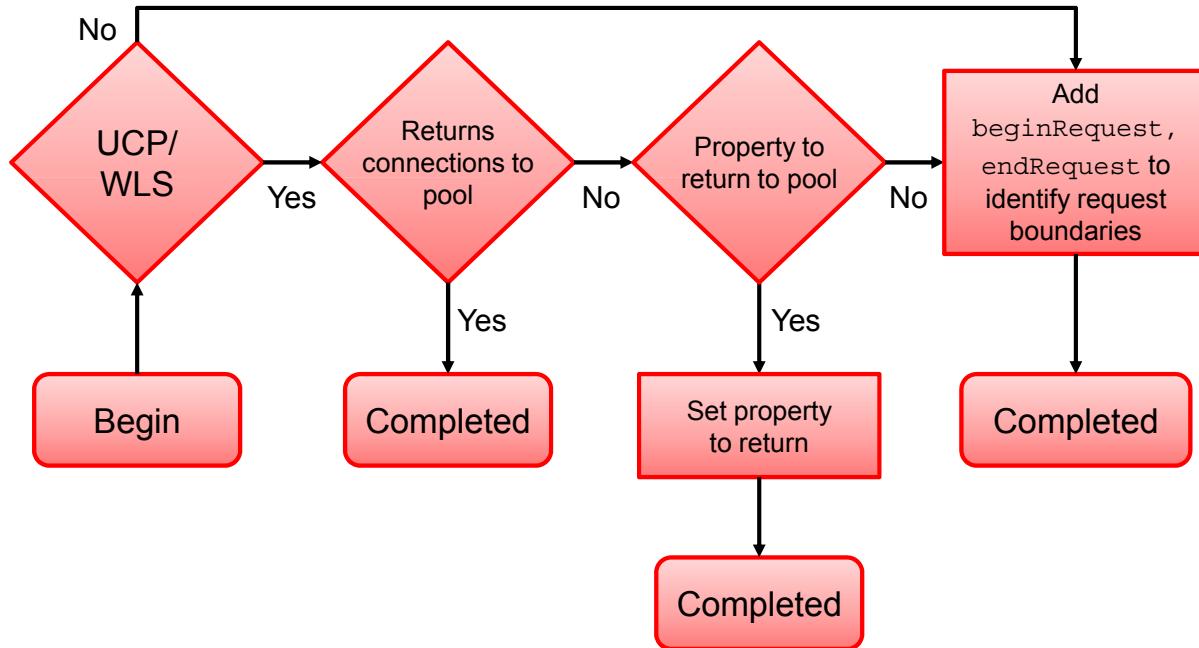
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Before using Application Continuity in conjunction with an application, the application owner must check the following:

1. Mark request boundaries if the application is not using Oracle Pools or is not releasing connections to Oracle Pools. Request boundaries tag the beginning and end of each request. Determine whether the application borrows and returns connections from the WebLogic Server Pool or Universal Connection Pool for each request, or whether `beginRequest` and `endRequest` APIs should be added to the application's own connection pool to identify request boundaries. If you are using Oracle Pools, but you are not releasing connections, consider your options to do so.
2. Determine whether the application uses Oracle JDBC concrete classes. To use Application Continuity, the deprecated concrete classes must be replaced. For information about the deprecation of concrete classes, including actions to take if an application uses them, see My Oracle Support Note 1364193.1.
3. Assess the impact of restrictions and side effects on your application. Use the `disableReplay` API for any request that should not be replayed. For example, if a request makes external calls by using one of the external PL/SQL messaging actions, decide whether the external call should be replayed or not.

4. Assess whether your application sets state outside the database request. If a state is set when a user starts a request and it is outside that request, replay needs to know about it in order to reexecute the calls.  
When using Oracle WebLogic Server or the Universal Connection Pool, connection labeling is recommended. The labeling is used for both run time and replay.  
If you use an application's own callback, register it at the WebLogic Admin Console, or at Universal Connection Pool, or JDBC replay driver levels.
5. Decide whether the application can keep mutables at replay. When a request is replayed, this function obtains a new value every time it is called. Keep mutable values when they are compatible with the application. Keep original values for seq.NEXTVAL, SYSDATE, SYSTIMESTAMP, and SYS\_GUID during failover.

# Handling Request Boundaries



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Request boundaries are used to tag the beginning and end of each request. They are required to resume recording following a nonreplayable operation, and also to reclaim the memory used for holding calls.

The flow chart shows how to decide whether request boundaries should be added:

1. Determine whether the application borrows and returns connections from the WebLogic Server Pool or Universal Connection Pool for each request, or whether `beginRequest` and `endRequest` APIs should be added to the application's own connection pool to identify request boundaries.
2. If you are using Oracle Pools, but you are not releasing connections to the pool, there is often a property to be set to release connections. Releasing connections scales much better and automatically embeds the request boundaries. This allows planned and unplanned failover support and better load balancing.

## Handling Request Boundaries: Example

```
doSQL(con, "CREATE SEQUENCE seq01 keep");

((oracle.jdbc.replay.ReplayableConnection)con).beginRequest();
System.out.println("\ncon.beginRequest()");

con.setAutoCommit(false);
System.out.println("\ncon.getAutoCommit()="+con.getAutoCommit());

String q = "select c1, seq01.NEXTVAL from test_tab";
System.out.println("\nExecute - "+q);

Statement s = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
    ResultSet.CONCUR_READ_ONLY);
ResultSet r= s.executeQuery(q);

while (r.next())
{
    System.out.println("c1="+r.getInt(1)+", seq="+r.getString(2));
}

r.close();
s.close();

((oracle.jdbc.replay.ReplayableConnection)con).endRequest();
System.out.println("\ncon.endRequest()");
```



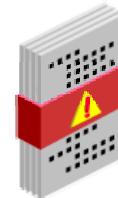
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows an example code fragment. In the example, the request boundary API calls are highlighted in red. If your application manages its own connection pool, these boundaries are best added at connection checkout (get) and connection release (close).

An application should use the Application Continuity APIs to mark request boundaries only if the connection pool or container does not use an Oracle connection pool or cannot release connections to the pool.

## Disabling Replay by Using the disableReplay API

- Use the disableReplay API for requests that should not be replayed.
- Make a conscious decision to replay external actions.
  - Autonomous transactions, UTL\_HTTP, UTL\_URL, UTL\_FILE, UTL\_FILE\_TRANSFER, UTL\_SMTP, UTL\_TCP, UTL\_MAIL, DBMS\_PIPE, and DBMS\_ALERT
- In addition, consider disabling replay when the application:
  - Synchronizes independent sessions
  - Uses time at the middle-tier in a logic execution
  - Assumes that ROWID values do not change
  - Assumes that location values do not change



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, the JDBC replay driver replays following a recoverable error. If the application has requests that the application does not want to be repeated, the application can explicitly call the disableReplay API to disable replay for those requests.

Application developers should make a conscious decision to allow replay for external actions, and especially for procedures that use the UTL\_HTTP package. If they should not be replayed, use the disableReplay API.

Additionally, it is recommended that you consider disabling replay when the application:

- Synchronizes independent sessions
- Uses time at the middle-tier in a logic execution
- Assumes that ROWID values do not change
- Assumes that location values do not change

This is because the assumptions contained in these scenarios may not be valid during replay, potentially yielding incorrect results. For further discussion of these scenarios refer to the chapter entitled Ensuring Application Continuity in the *Oracle Database Development Guide 12c Release 1 (12.1)*.

# Connection Initialization Options

- No callback
  - The application builds up its own state during each request.
- Connection labeling
  - Performance is improved by avoiding some initialization.
  - Available with UCP or WebLogic Server
  - The replay driver uses Connection Labeling when present.
- Connection initialization callback
  - The replay driver uses callback to set the initial state of the session at run time and replay.
  - Register with WebLogic, UCP, and JDBC Thin.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Nontransactional session state (NTSS) is state of a database session that exists outside transactions and is not protected by recovery. For applications that use stateful requests, the nontransactional state is reestablished as the session is rebuilt by Application Continuity.

For applications that set state only at the beginning of a request, or for stateful applications that gain performance benefits from using connections with a preset state, choose one of the following callback options:

- **No callback:** In this scenario, the application builds up its own state during each request.
- **Connection labeling:** Connection Labeling is a generic pool feature that is recommended for its excellent performance. When Connection Labeling is present, Application Continuity uses it. Connection Labeling is a feature of Universal Connection Pool (UCP) and Oracle WebLogic server. The application can be modified to take advantage of the preset state on connections. Connection Labeling APIs determine how well a connection matches, and use the labeling callback to populate the gap when a connection is borrowed. Not all applications can use Connection Labeling, because it requires some application recoding.

- **Connection initialization callback:** In this scenario, the replay driver uses an application callback to set the initial state of the session during run time and replay. The JDBC replay driver provides an optional connection initialization callback interface and methods to register and unregister connection initialization callbacks in the `oracle.jdbc.replay.OracleDataSource` interface.

When registered, the initialization callback is executed at each successful checkout at run time and each reconnection following a recoverable error. If the callback invocation fails, replay is disabled on that connection. Use the connection initialization callback only when the application has not implemented Connection Labeling, and needs state information that is not established at the request.

Callback registration is performed using the WebLogic Admin Console, in UCP, or in the JDBC Thin Replay driver.

## Mutable Objects and Application Continuity

- Requests using mutable functions can fail to replay if the function result changes.
- When a mutable privilege is granted, replay applies the original function result.

Mutable Function	Application A	Application B	Application C
<code>SYSDATE,</code> <code>SYSTIMESTAMP</code>	Original	Current	Not Applicable
<code>SEQUENCE.NEXTVAL,</code> <code>SEQUENCE.CURRVAL</code>	Original	Original	Current
<code>SYS_GUID</code>	Original	Not Applicable	Not Applicable



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A mutable object is a function that obtains a new value every time it is called. An example of a mutable is a call to the `SYSTIMESTAMP` function. When a request is replayed, the default and desired treatment of mutable objects can vary. Applications using Application Continuity can determine whether to keep the original values from mutable functions if the request is replayed.

Support for keeping mutable object values is currently provided for `SYSDATE`, `SYSTIMESTAMP`, `SYS_GUID`, and `SEQUENCE.NEXTVAL`. If the original values are not kept and if different values for these mutable objects are returned to the client, replay will be rejected because the client will see different results.

The table in the slide shows examples of the different approaches that different applications might use for mutable objects. No approach is recommended over another. They are just presented to illustrate some of the possibilities that might reasonably exist.

# Keeping Mutable Objects for Replay

- SQL commands:

```
SQL> GRANT [KEEP DATE TIME | KEEP SYSGUID] TO <user>;
SQL> REVOKE [KEEP DATE TIME | KEEP SYSGUID] FROM <user>;
SQL> GRANT KEEP SEQUENCE ON <sequence_name> TO <user>;
SQL> REVOKE KEEP SEQUENCE ON <sequence_name> FROM <user>;
SQL> CREATE SEQUENCE <sequence_name> [KEEP|NOKEEP];
SQL> ALTER SEQUENCE <sequence_name> [KEEP|NOKEEP];
```

- Examples:

```
SQL> CREATE SEQUENCE new_seq KEEP;
SQL> ALTER SEQUENCE existing_seq KEEP;
SQL> GRANT KEEP SEQUENCE on sales.seq1 TO user2;
SQL> GRANT KEEP DATE TIME TO user2;
SQL> GRANT KEEP SYSGUID TO user2;
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows the SQL commands that you can use to configure mutable objects for replay. The following notes elaborate further:

- The database user running the application may have the KEEP DATE TIME and KEEP SYSGUID privileges granted, and the KEEP SEQUENCE object privilege on each sequence whose value is to be kept. Note that these privileges cannot be granted using GRANT ALL ON <object>. In addition, it is not recommended to grant DBA privileges to database users running applications for which you want replay to be enabled. Grant only privileges that are necessary for such users.
- Sequences may be defined with the KEEP attribute, which keeps the original values of SEQUENCE.NEXTVAL for the sequence owner. Note that specifying KEEP or NOKEEP with CREATE SEQUENCE or ALTER SEQUENCE applies to the owner of the sequence.
- If granted privileges are revoked between run time and failover, the mutable values that are collected are not applied for replay.
- If new privileges are granted between run time and failover, mutable values are not collected and, therefore, cannot be applied for replay

## Configuring the JDBC Replay Data Source

Use the `oracle.jdbc.replay.OracleDataSourceImpl` data source to obtain JDBC connections.

- Configure the data source in the property file for UCP or WebLogic Server.
- Reference the data source in JDBC Thin applications.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To use Application Continuity for Java, you must use the `oracle.jdbc.replay.OracleDataSourceImpl` data source to obtain JDBC connections. This data source supports all the properties and configuration parameters of all the Oracle JDBC data sources (for example, `oracle.jdbc.pool.OracleDataSource`).

You can configure the replay data source by changing the data source property for Universal Connection Pool or by setting it using the WebLogic Console. You can also set the replay data source for JDBC Thin applications in the property file or directly in the application.

# Configuring Database Services for Application Continuity

Example:

```
$ srvctl add service -db orcl -service acservice  
  -serverpool ora.orcldb  
  -cardinality singleton  
  -failovertype TRANSACTION  
  -commit_outcome TRUE  
  
  -failoverretry 50  
  -failoverdelay 5  
  -retention 86400  
  -replay_init_time 1800  
  -notification TRUE
```

**Mandatory Settings for Application Continuity**

**Optional Settings for Application Continuity**



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To use Application Continuity, set the following mandatory service attributes:

- FAILOVER\_TYPE=TRANSACTION to enable Application Continuity
- COMMIT\_OUTCOME=TRUE to enable Transaction Guard

You can also set the following attributes:

- RETENTION\_TIMEOUT: This setting specifies the number of seconds that the commit outcome is retained. The default is 86400 (24 hours), and the maximum is 2592000 (30 days).
- REPLAY\_INITIATION\_TIMEOUT: This setting specifies the number of seconds within which replay must start. If replay does not start within the specified time then it is abandoned.
- FAILOVER\_RETRIES: This setting specifies the number of connection retries for each replay attempt. If replay does not start within the specified number of retries then it is abandoned.
- FAILOVER\_DELAY: This setting specifies the delay in seconds between connection retries.
- AQ\_HA\_NOTIFICATIONS: Set this to TRUE for FAN.

# Resource Requirements for Application Continuity

- Application Continuity
  - Java client
    - Increase memory to maintain replay queues.
    - Additional CPU is consumed for garbage collection and to build proxies.
  - Database server
    - Additional CPU is required for validation.
    - CPU overhead is minimal on current Intel and SPARC CPUs where validation is hardware-assisted.
- Transaction Guard
  - Built into the Database kernel
  - Minimal overhead
  - Excellent scalability



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application Continuity has the following resource requirements:

- **CPU:** Application Continuity operates on both the database server and application client and needs additional CPU resources to operate. Application Continuity consumes CPU resources on the database server to perform validation during replay. Note that CPU overhead is reduced on platforms with current Intel and SPARC CPUs where validation is assisted in hardware (using Cyclic Redundancy Checks [CRCs]). On the client side, Application Continuity uses some CPU resources above the base driver. Additional CPU costs are incurred for building proxy objects and garbage collection.
- **Memory:** The JDBC replay driver requires more memory than the base driver because the calls are retained until the end of a request. At the end of the request, the calls are released to the garbage collector. This action differs from the base driver, which releases closed calls.

The memory consumption of the replay driver depends on the number of calls per request. If this number is small, the additional memory consumption of the replay driver is less and is comparable to the base driver.

To obtain the best performance, set the same value for both the `-Xmx` and `-Xms` parameters on the client. This sets the initial heap size and maximum heap size to the same size ensuring that JVM performance is not impacted by attempts to use lower settings.

## Quiz

Application Continuity attempts to mask recoverable database outages from applications and users by restoring database sessions and replaying database calls.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: a**

## Quiz

Which of these is not one of the phases in Application Continuity?

- a. Run time
- b. Replay
- c. Reconnect
- d. Restore



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: d**

## Quiz

Select the statements that apply to Application Continuity:

- a. Application Continuity is supported with thin JDBC clients.
- b. The replay target database must have the same database ID, ancestors, and descendants as the source database.
- c. Replay is not supported if the transaction executes the ALTER SYSTEM or ALTER DATABASE statement.
- d. Replay is supported for applications developed using Oracle XA.
- e. Replay is supported if you are using Active Data Guard with read/write database links to another database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: a, b, c**

## Summary

In this lesson, you should have learned how to:

- Describe the purpose of Transaction Guard and Application Continuity
- Describe the key concepts relating to Application Continuity
- Describe the side effects and restrictions relating to Application Continuity
- Describe the requirements for developing applications that leverage Application Continuity
- Configure Application Continuity



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Practice 9 Overview: Using Application Continuity

In this practice, you will use Application Continuity against a RAC database to demonstrate how Application Continuity helps an application to seamlessly recover after the failure of a RAC instance.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# 10

## RAC New Features

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Objectives

After completing this lesson, you should be able to describe the Oracle Database 12c new features that enhance Oracle RAC.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Lesson Overview

- Oracle Database 12c new features that enhance RAC:
  - Application continuity
  - Flex ASM
  - Oracle Multitenant
  - Policy-based cluster management
  - What-if command evaluation
  - Restricting service registration with valid node checking
  - Role-separated installation support for Windows



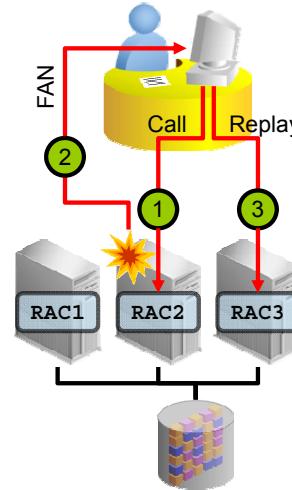
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Database 12c introduces many new features that enhance Oracle RAC. The slide lists the key new features associated with Oracle RAC that are discussed in this lesson.

Many of the new features also have applications outside of Oracle RAC and are covered in detail elsewhere in this course and in other courses. For features that have already been covered elsewhere in this course, an overview of the feature is provided along with an outline of how the feature specifically relates to Oracle RAC. For features with detailed coverage in another course, a similar RAC-oriented outline is provided along with a reference to other relevant training courses.

# RAC and Application Continuity

- Application Continuity transparently replays database requests after a failed session.
  - Users are shielded from many types of problems.
- With RAC, Application Continuity replays requests on another RAC instance.
- Using Application Continuity with RAC provides:
  - Protection against a wider variety of failure scenarios
  - Faster reconnect and replay



ORACLE

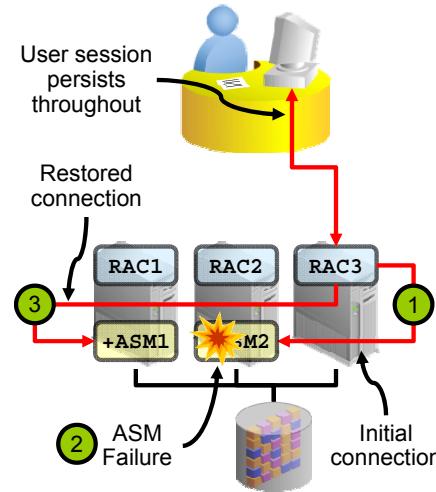
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Application Continuity is a feature that rapidly and transparently replays a request against the database after a recoverable error that makes the database session unavailable. The request can contain transactional and nontransactional work. With Application Continuity, the end user experience is improved by masking many system, communication, hardware, and storage problems from the end user.

When Application Continuity is used in conjunction with Oracle RAC, failed sessions can be quickly restarted and replayed on another RAC database instance. Using Application Continuity in conjunction with Oracle RAC provides protection against a wider variety of possible failures compared with using Application Continuity against a single instance database. In addition, using Application Continuity in conjunction with Oracle RAC enables quicker replay compared with using Application Continuity in conjunction with Data Guard, because reconnecting to another already running database instance can be completed in a few seconds, while a Data Guard failover operation may take a few minutes.

## RAC and Flex ASM

- Flex ASM allows Oracle Database instances to connect to ASM on another node in the cluster.
  - If an ASM instance fails, its clients connect to another instance.
  - Resources are saved because ASM is not required on every database server.
- Therefore:
  - Flex ASM improves RAC availability.
  - Flex ASM frees resources which can be used by RAC.



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

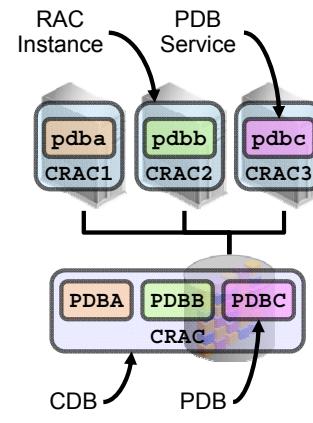
Flex ASM allows Oracle Database instances to connect to ASM on another node in the cluster. As a result, if an ASM instance fails, its clients can reconnect to another ASM instance. Furthermore, Flex ASM enables a smaller pool of ASM instances to serve a larger pool of database servers, which reduces the overall resource footprint of ASM across the cluster.

Flex ASM increases the availability of Oracle RAC because the failure of an ASM instance no longer results in the failure of the corresponding database instance. With Flex ASM, if an ASM instance fails, any RAC database instances connected to it automatically and transparently reconnect to another ASM instance. Database clients and end users are not affected by the failure except that there may be a slight pause in processing.

In addition, the resource savings associated with Flex ASM can be applied to your RAC database instances allowing them to benefit accordingly.

# RAC and Oracle Multitenant

- Oracle Multitenant enables multiple PDBs to share the resources of a single CDB.
  - Enables more efficient consolidation
  - Maintains isolation between PDBs
    - Separate data files, users, schemas, privileges, and so on
- Each PDB is exposed as a service.
- Using Oracle Multitenant in conjunction with RAC:
  - Each PDB service can be exposed on some or all of the RAC instances.
  - Each PDB service can be associated with a server pool.
    - PDB services can be managed using policy-based Clusterware management



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A multitenant container database (CDB) is an Oracle database that includes one or more pluggable databases (PDBs). A PDB is a portable collection of schemas, schema objects, and nonschema objects that appears to an application client as a self-contained database. The name of this new Oracle Database 12c capability is *Oracle Multitenant*.

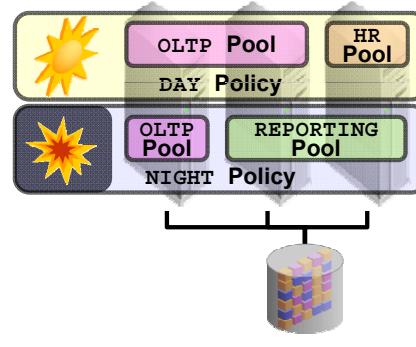
Oracle Multitenant enables multiple PDBs to share the resources of a single CDB instance while maintaining database-level isolation. This enables more efficient consolidation of databases while maintaining separate storage (data files) and security definitions (users, schemas, privileges, and so on).

Using the multitenant architecture, each PDB is exposed as a service. In an environment where Oracle RAC is used for the CDB, each PDB service can be exposed across all the RAC instances or across a subset of instances or on a single instance (as shown in the diagram in the slide). Furthermore, PDB services can be associated with server pools and managed in association with policy-based cluster management.

For more information on Oracle Multitenant, refer to the Oracle Database 12c documentation library.

# RAC and Policy-Based Cluster Management

- Policy-based cluster management is enhanced to provide:
  - Server categorization with extended server attributes to govern node placement
  - A library of policy definitions with an easy way of switching between policies
- Benefits for RAC users:
  - Precise control for placement of RAC instances and services
  - Quicker and easier policy changes
    - One-step policy changes automatically start and stop RAC database instances and services.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

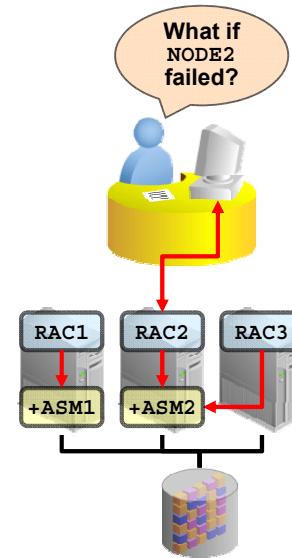
Oracle Clusterware 11g, release 2 introduced policy-based cluster management. With this capability, a cluster could be logically divided into groups of servers known as server pools. With Oracle Clusterware 12c, policy-based cluster management is enhanced in three important ways. First, numerous server attributes allow for greater flexibility and control over node assignments to different server pools. Second, an extended policy framework allows administrators to maintain a library of policies and easily switch between them as required. Finally, policy-based cluster management has been unified with QoS Management.

In release 12.1, policy-based cluster management enhances Oracle RAC in the following ways:

- Using server categorization, RAC instances and database services can be placed on servers with specific characteristics rather than being arbitrarily spread across the cluster. This allows available resources to be allocated more precisely. It also enables clusters of heterogeneous servers to be used effectively by Oracle RAC databases.
- Using the new policyset, administrators can define a library of policy definitions to cater for different workloads and priorities. These policies can then be used to quickly and easily reconfigure the cluster for different situations, which makes it easier to manage a RAC database environment, especially where multiple RAC databases share a cluster.

## RAC and What-If Command Evaluation

- What-if command evaluation previews the effect of a cluster management operation.
  - Allows administrators to analyze the impact before performing the operation
  - Facilitates smooth operation of the cluster; no surprises
- RAC database administrators can analyze:
  - Adding, starting, stopping, modifying, and relocating databases
  - Adding, starting, stopping, modifying, and relocating services
  - Adding, modifying, and removing server pools
  - Relocating a server from one server pool to another
  - Various failure scenarios



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

What-if command evaluation provides a set of commands to determine the impact of a cluster management operation before the operation is actually executed. It helps administrators to smoothly maintain the cluster and minimizes the potential for surprises.

What-if command evaluation using the `crsctl eval` command is predominantly performed by cluster administrators.

RAC database administrators who do not have access to the `crsctl` utility can use the `srvctl` command with the `-eval` option to analyze the effects of:

- Adding, starting, stopping, modifying, and relocating databases
- Adding, starting, stopping, modifying, and relocating services
- Adding, modifying, and removing server pools
- Relocating a server from one server pool to another

In addition, using the `srvctl predict` command, RAC database administrators can evaluate the consequences of a failure affecting different types of resources, including databases, services, ASM instances, ASM disk groups, networks, VIPs, listeners, and so on.

## Restricting Service Registration with Valid Node Checking

This feature provides the ability to manage a set of addresses from which registration requests are allowed by the listener.

- The network administrator can specify a list of valid nodes, excluded nodes, or disable valid node checking.
  - The list of valid nodes explicitly lists the nodes and subnets that can register with the listener.
  - The list of excluded nodes explicitly lists the nodes that cannot register with the listener.
- Instance registration with a listener succeeds only when the request originates from a valid node.
- The control of dynamic registration results in increased manageability and security of Oracle RAC deployments.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Valid node checking provides the ability to configure and dynamically update a set of IP addresses or subnets from which registration requests are allowed by the listener. Database instance registration with a listener succeeds only when the request originates from a valid node. The network administrator can specify a list of valid nodes, excluded nodes, or disable valid node checking. The list of valid nodes explicitly lists the nodes and subnets that can register with the listener. The list of excluded nodes explicitly lists the nodes that cannot register with the listener. The control of dynamic registration results in increased manageability and security of Oracle RAC deployments.

## Restricting Service Registration with Valid Node Checking

Default configuration:

- Valid node checking is enabled.
- Registration requests from all nodes within the subnet of SCAN listener can register with the SCAN listener.
- Non-SCAN listeners only accept registration from instances on the local node.
- Remote nodes or nodes outside the subnet of the SCAN listener can be included on the list of valid nodes by:
  - Setting the REGISTRATION\_INVITED\_NODES\_ALIAS parameter in the `listener.ora` file
  - Modifying the SCAN listener using SRVCTL:

```
$ srvctl modify scan_listener [-invitednodes <node_list>]  
[-invitedsubnets <subnet_list>]
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, valid node checking for registration is enabled. In the default configuration, all nodes within the subnet of SCAN listener can register with the SCAN listener. Non-SCAN listeners only accept registration from instances on the local node. Remote nodes or nodes outside the subnet of the SCAN listener must be included on the list of valid nodes by using the REGISTRATION\_INVITED\_NODES\_ALIAS parameter in the `listener.ora` file or by modifying the SCAN listener using SRVCTL.

## Role-Separated Installation Support for RAC on Windows

Oracle Database 12c on Windows supports the use of an Oracle Home User, specified at installation time.

- The Oracle Home User is associated with a Windows domain user.
- The Windows user should be a low-privileged account to ensure that the Oracle Home user has a limited set of privileges.
  - The Oracle Database services have only those privileges required to run Oracle products.
- Windows Administrator user privileges are still required to perform Oracle software maintenance tasks.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Starting with Oracle Database 12c, Oracle Database in Windows supports the use of an Oracle Home User, which can be specified at installation time. In turn, this enables role-separated installation support for Oracle Database on Windows.

The Oracle Home User is associated with a Windows domain user. The Windows User Account should be a low-privileged (non-Administrator) account to ensure that the Oracle Home User has a limited set of privileges, thus ensuring that the Oracle Database services have only those privileges required to run Oracle products.

Windows Administrator user privileges are still required to perform Oracle software maintenance tasks including installation, upgrading, patching, and so on. Oracle Database administrative tools have been enhanced to ask for the password of the Oracle Home User, if needed. In Oracle RAC environments, you can store the password for the Oracle Home User in a secure wallet. If such a wallet exists, the Oracle Database administrative tools automatically use the password from the wallet and do not require the user to enter the password for the Oracle Home User.

Release 12.1 also introduces a new Windows utility called Oracle Home User Control. This is a command-line tool that displays the Oracle Home User name associated with the current Oracle Home. This utility also enables you to modify the Windows Services used by Oracle to use a new password when the password for Oracle Home User is changed.

## Summary

In this lesson, you should have learned how to describe the Oracle Database 12c new features that enhance Oracle RAC.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# 11

## Oracle Data Guard New Features

ORACLE®

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Objectives

After completing this lesson, you should be able to describe the Data Guard 12c new features designed to improve the following areas:

- Far Sync
- Data Guard Transport
- Active Data Guard
- Database Rolling Upgrades
- Data Guard Broker



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## Road Map

### Oracle Database 12c: Data Guard New Features

- **Far Sync and Data Guard Transport Enhancements**
- Active Data Guard
- Database Rolling Upgrades
- Other Data Guard Enhancements
- Data Guard Broker



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are many enhancements to Data Guard with Oracle Database 12c, release 1. They can be broken into the following broad categories: far sync and Data Guard transport enhancements; Oracle Active Data Guard enhancements; improvements to database rolling upgrades; miscellaneous Data Guard enhancements; and changes to the Data Guard broker. We will begin the lesson with far sync and Data Guard transport enhancements.

## Data Guard 12c: Far Sync



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

The diagram in the slide displays a flat projection of a world map showing North America, South America, Europe, and Africa. Somewhere in North America, an icon in the slide represents the primary database system. Somewhere in Europe, an icon in the slide represents a standby database system. A short distance from the primary database system is a new type of an instance called a far sync that is introduced as a new feature with Data Guard in Oracle Database 12c.

To fully support High Availability (HA) and allow switchover to the standby database system, you should create a second far sync instance a short distance from the standby database system. This second far sync is not shown in the slide.

## Data Guard 12c: Far Sync

What is a far sync?

- A lightweight Oracle database instance
  - Only a standby control file, password file, standby redo logs, and archive logs
  - No Oracle data files, no database to open for access, not running redo apply
- Deployed within a distance that the primary can tolerate the impact of network latency on synchronous transport
  - Looks like any other Data Guard destination to the primary database
  - With Data Guard 12c Fast Sync, further extends the practical distance between the primary and a far sync

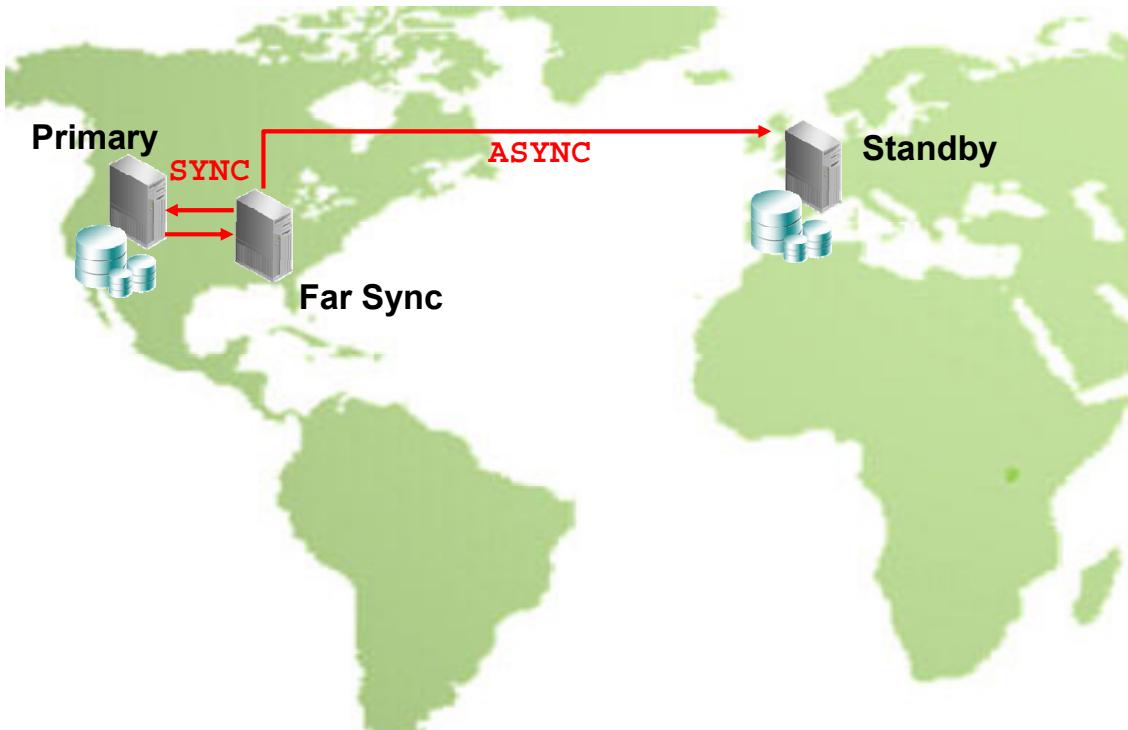


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A Data Guard far sync is a remote Data Guard destination that accepts redo from the primary database and redistributes that redo throughout the Data Guard configuration. It is similar to a physical standby database in that it manages a control file, receives redo into Standby Redo Logs (SRLs), and archives those SRLs to local Archived Redo Logs (ARLs). However, unlike a standby database, a far sync does not manage data files, cannot be opened for access, and cannot run redo apply. These limitations yield the benefit of using fewer disk and processing resources. More importantly, a far sync provides the ability to fail over to a terminal database with no data loss.

Many variables, such as the redo write size, available network bandwidth, round-trip network latency, and standby I/O performance while writing to the standby redo logs, impose practical limitations on the distance that a standby can reside from a primary database. This includes a far sync. A Data Guard 12c new feature called Fast Sync removes the standby I/O performance from the limitations list and helps to extend the distance that a far sync can reside from the primary database. Data Guard 12c Fast Sync will be discussed later in the lesson.

## Far Sync: Redo Transport



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide displays a flat projection of a world map showing North America, South America, Europe, and Africa. Somewhere in North America, an icon in the slide represents the primary database system. Somewhere in Europe, an icon in the slide represents a standby database system. A short distance from the primary database system is a new type of instance called a far sync that is introduced as a new feature with Data Guard in Oracle Database release 12c.

The redo transport uses synchronous (SYNC) transmission between the primary database system and a far sync. This imposes a practical limit on the distance between the primary database system and the far sync instance because it impacts the performance on the primary database. The log writer (LGWR) process of the primary database system has to wait for confirmation from the Network Server SYNC (NSS) process that the redo has been transmitted over the network before it can proceed with the next transaction. Redo transport from the far sync to the standby database system uses asynchronous communication. This eliminates the requirement of waiting for acknowledgment from the Network Server ASYNC (NSA) process on the far sync instance, creating near zero performance impact because of network transmission even if intercontinental distances are involved.

## Far Sync: Redo Transport

How does redo transport work with a far sync configuration?

- The far sync instance receives redo synchronously from the primary database.
- The far sync instance forwards redo asynchronously in real time to its final destination.
- Redo transport supports one local and up to 30 additional local or remote destinations.
- Oracle Recovery Manager (Oracle RMAN) deletion policies are used to automate archive log management.
- A far sync instance can also compress redo transport.
- An alternate far sync can be used for HA.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In maximum availability mode, the far sync instance is relatively close to the primary database to minimize network latency, and the primary database services the far sync by using `SYNC` transport.

When a primary database services a far sync by using `SYNC` transport, all committed redo resides on disk at the far sync. That way, the far sync can use one of the terminal standby destinations for a no-data-loss failover if the primary database is lost.

The far sync uses `ASYNC` transport to redistribute the incoming redo to terminal standbys that can be much farther away. This extends no-data-loss protection to destinations that are too far away for a primary database to feasibly service directly with `SYNC` transport because of the resulting degradation in transaction throughput. This is a case where a far sync is beneficial even if there is only one standby destination in the configuration.

The redo transport architecture for a far sync is configured like any other type of standby database: Use the `LOG_ARCHIVE_DEST_n` parameter, which supports up to 30 destinations. Other parameters, such as `DB_UNIQUE_NAME`, `LOG_ARCHIVE_CONFIG`, `FAL_SERVER`, `FAL_CLIENT`, `LOG_FILE_NAME_CONVERT`, and `DB_FILE_NAME_CONVERT`, are also configured in the same way as any other type of standby database. With the Oracle Database 12c Advanced Security option, redo transport compression is also available.

## Far Sync: Alternate Redo Transport Routes



```
LOG_ARCHIVE_DEST_STATE_2='ENABLE'

LOG_ARCHIVE_DEST_2='SERVICE=chicagoFS SYNC AFFIRM MAX_FAILURE=1
ALTERNATE=LOG_ARCHIVE_DEST_3 VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=chicagoFS'

LOG_ARCHIVE_DEST_STATE_3='ALTERNATE'

LOG_ARCHIVE_DEST_3='SERVICE=boston ASYNC ALTERNATE=LOG_ARCHIVE_DEST_2
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=boston'
```

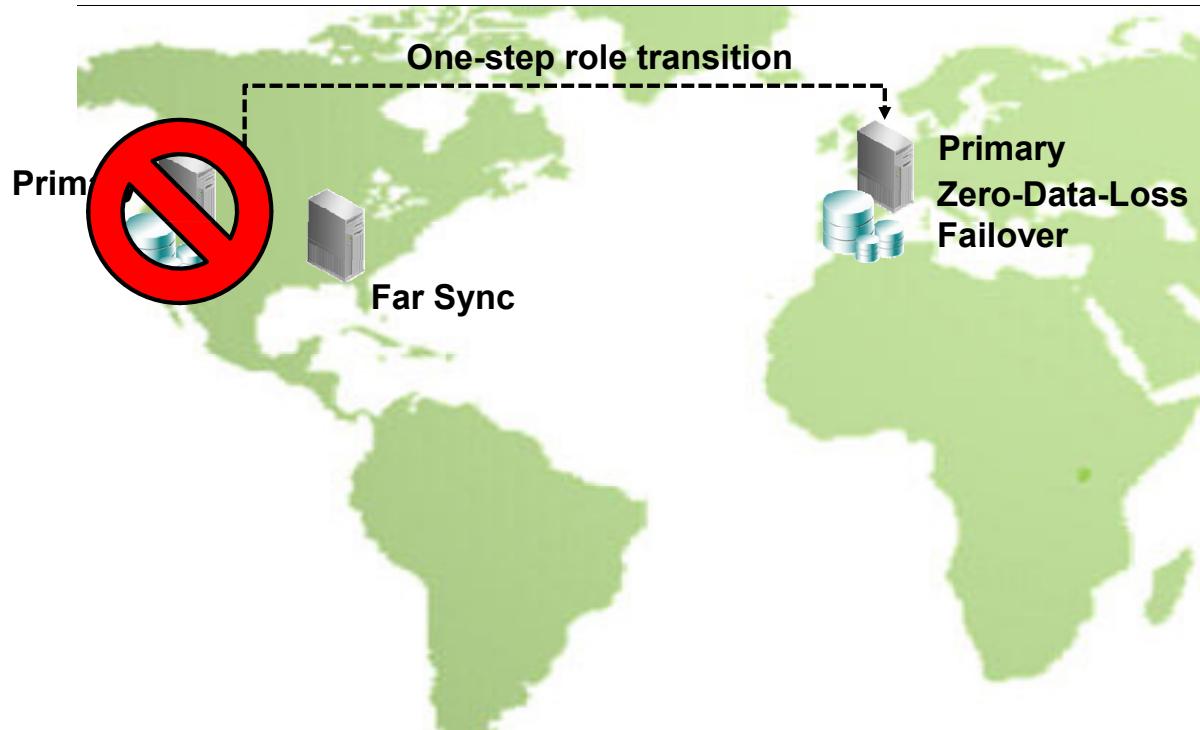
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the event that communication with the far sync instance is lost, you can optionally configure the terminal standby to automatically become the alternate destination. This will reduce the amount of data loss by allowing Oracle Data Guard to ship redo asynchronously directly from the primary to the terminal standby, temporarily bypassing the far sync instance.

This enables Oracle Data Guard to continue sending redo, asynchronously, to the terminal standby `boston` when it can no longer send the redo directly to the far sync instance `chicagoFS`. When the far sync instance becomes available again, Oracle Data Guard automatically resynchronizes the far sync instance `chicagoFS` and returns to the original configuration in which the primary sends redo to the far sync instance and the far sync instance forwards that redo to the terminal standby. When the synchronization is complete, the alternate destination (`LOG_ARCHIVE_DEST_3` in the example) will again become dormant as the alternate.

## Far Sync: Role Transitions



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Real-time cascading, a new feature introduced with Data Guard 12c, enables either a physical standby database or a far sync to cascade the primary database redo immediately as it is being written to the standby redo log file. Therefore, the terminal destination has minimal redo transport lag with respect to the primary database and provides nearly the same level of data protection as any standby database that receives redo directly from a primary database.

The diagram in the slide displays a flat projection of a world map showing North America, South America, Europe, and Africa. Somewhere in North America, an icon in the slide represents the primary database system. A red circle indicates a site failure with the primary database. A short distance from the primary database system is a new type of instance called a far sync that has received all committed redo information prior to the failure of the primary database site. Somewhere in Europe, an icon in the slide represents a standby database system. A new one-step role transition or failover is performed to the database site in Europe, making it the new primary database system with zero data loss.

## Far Sync: Role Transitions

Performing role transitions with a far sync in the configuration:

- One-step, zero-data-loss failover
  - Same failover/switchover commands used for any Data Guard configuration, whether or not far sync is used
  - Also supports Fast-Start Failover
- Data Guard transparently “drains the pipe” so that the remote failover target has all committed redo.
- A second far sync can be preconfigured to transmit in reverse direction after failover/switchover.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Including a far sync in the Data Guard configuration does not alter the ability or syntax needed to perform a role transition, such as switchover or failover. When your configuration includes a far sync instance, each destination should define a `LOG_ARCHIVE_DEST_n` parameter that points back to its source for use during a role transition.

For example, a far sync should have one that points back to the primary (the source of the far sync), and the terminal destination should have one that points back to the far sync (the source of the terminal destination). The terminal standby should also have a `LOG_ARCHIVE_DEST_n` parameter that points back to the primary, for use in situations involving automatic block media recovery or global sequence support.

Upon failure of the primary database, all committed redo has been sent to the far sync instance and is available to be forwarded to a standby database asynchronously. This allows a primary database and standby database configuration to fully support a zero-data-loss failover when a far sync instance is between the two database sites. However, although redo is forwarded in real time, a second network hop creates the potential for additional data loss if an outage prevents all redo from reaching the terminal destination. In other words, data loss would require a failure at the primary site, followed by a second network failure that would prevent transmission of the redo from the far sync to the terminal destination.

## Oracle Data Guard 12c: Far Sync Creation

- Create a control file using the mounted primary database:

```
SQL> ALTER DATABASE CREATE FAR SYNC INSTANCE CONTROLFILE  
AS '/tmp/boston1.ctl';
```

- Copy the parameter file (PFILE) and password file used by the primary database. (Several initialization parameters must also be modified for the far sync instance.)
- Copy the control file to the far sync system.
- Create standby redo log files as you would for any standby.
- Start the far sync instance and mount the control file by using standard syntax.
- The **DATABASE\_ROLE** column in V\$DATABASE will contain the value **FAR SYNC**.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You start creating a far sync by issuing a command on the primary database to create a new far sync instance control file. The primary database must be mounted or open to issue the command. The syntax for creating a far sync instance control file is:

```
SQL> ALTER DATABASE CREATE FAR SYNC INSTANCE CONTROLFILE AS  
'/PATH/FILENAME' ;
```

Next, copy the resulting control file, along with a copy of the primary database parameter file and the password file, to the machine that will host the far sync instance. It is assumed that the database software has already been installed on the machine that will host the far sync instance and a database listener has been configured. If the primary database uses a server parameter file (SPFILE), create a parameter file (PFILE) before copying the file. The following list of initialization parameters may need changing for the far sync instance:

DB\_UNIQUE\_NAME; CONTROL\_FILES, FAL\_SERVER, LOG\_ARCHIVE\_CONFIG, and LOG\_ARCHIVE\_DEST\_n.

On both the primary and far sync systems, use Oracle Net Manager to create a network service name for the primary and standby databases that will be used by redo transport services. This name will be supplied in the LOG\_ARCHIVE\_DEST\_n parameter. Use standard syntax to start and mount the far sync. After you mount the far sync, the DATABASE\_ROLE column in the view V\$DATABASE shows 'FAR SYNC'.

## Benefits: Far Sync

Provides no-compromise data protection and primary offload

- Ideal combination of attributes
  - Best data protection, least performance impact
  - Lower cost and complexity compared to previous multisite architecture
  - No change in applications required
  - Ideal for a combined near HA plus far DR model adopted by a growing number of users
- Relevant to existing Data Guard ASYNC configurations
  - Now practical to implement SYNC zero data loss where only ASYNC was previously possible
  - Offloads overhead for redo transport compression and for servicing multiple destinations

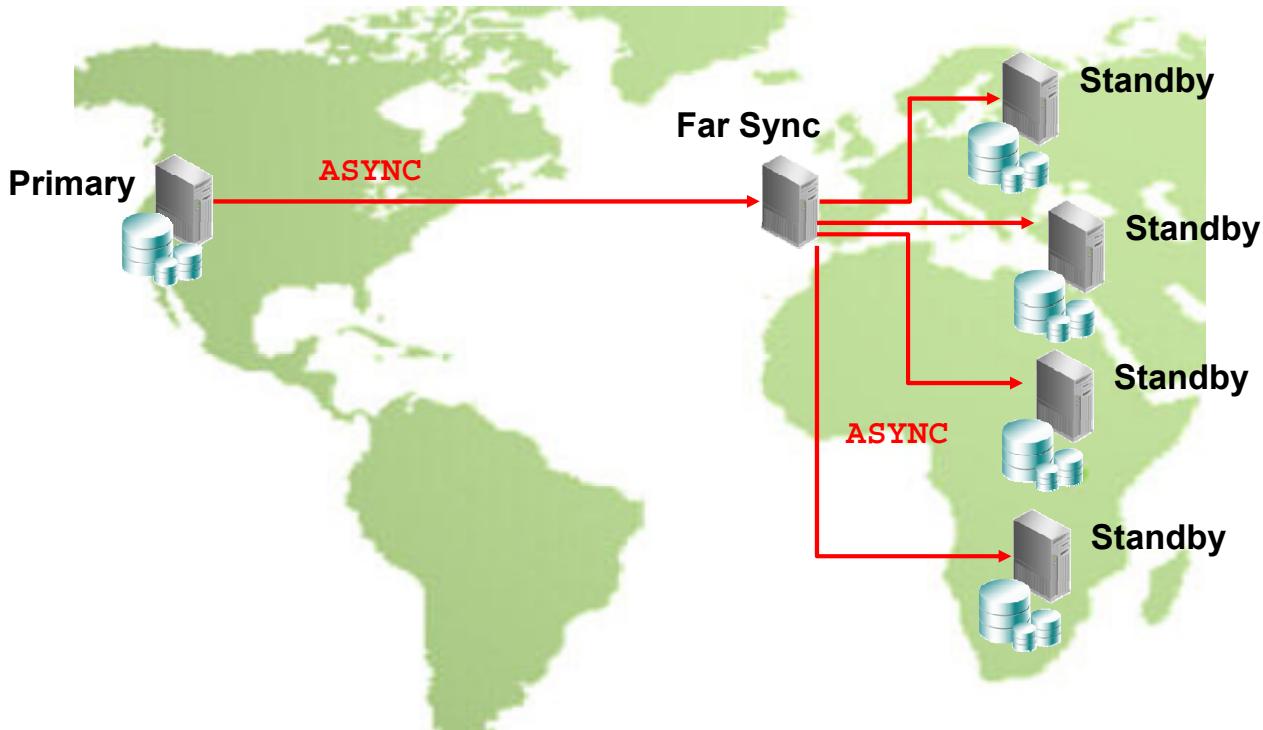


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are many benefits of a far sync. A far sync transparently integrates in a maximum availability mode configuration for no-compromise data protection. It requires no changes in applications that currently use a primary database and standby database architecture. A far sync has the least performance impact because it does not manage data files, cannot be opened for access, and cannot run redo apply. Features such as redo transport compression that would have a performance impact on a primary database can be offloaded to the far sync host.

It is ideal for an HA model serving as a nearby archived log repository, yet still allows the benefits of a far-away disaster recovery model by extending no-data-loss protection to destinations that are too far away for a primary database to feasibly service directly by using SYNC transport.

## Far Sync: Alternate Design



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Up to this point in this lesson, the discussion of a redo transport mechanism from a primary database to a far sync has used synchronous transmission. It is possible to also use asynchronous redo transport in a maximum performance Data Guard configuration.

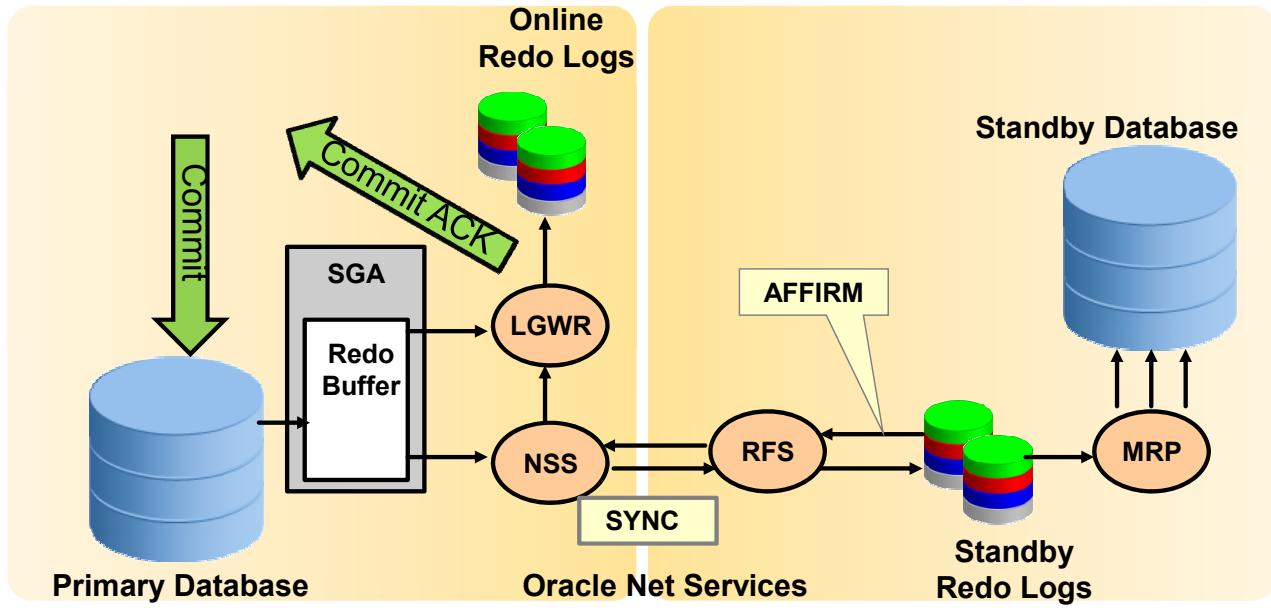
In maximum performance mode, the primary database services the far sync destination by using ASYNC redo transport, regardless of the physical distance between the primary and the far sync. High network latencies do not affect transaction throughput when a destination is serviced with ASYNC transport.

In maximum performance mode, a far sync can benefit Data Guard configurations that manage more than one remote destination. Each destination that a primary directly services takes computing resources away from applications running on the primary. When a far sync is used, the primary only has to service the far sync, which then services the rest of the configuration. The performance benefit increases as the number of destinations increases.

The slide shows a world map with a primary database in North America using ASYNC redo transport to a far sync located in Europe. The far sync is then used to cascade the primary redo to multiple standby sites located throughout Europe and Africa. Up to 30 terminal destinations are supported.

## Data Guard 11g, Release 2: Maximum Availability Protection Mode

The maximum availability protection mode required the SYNC redo transport with the AFFIRM settings.



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Data Guard protection modes are a set of rules that the primary database must follow when running in a Data Guard configuration. Data Guard protection modes define how the redo must be transported to the standby by the primary database and what the primary database will do when a standby or the network fails. There are three protection modes: maximum performance, maximum protection, and maximum availability.

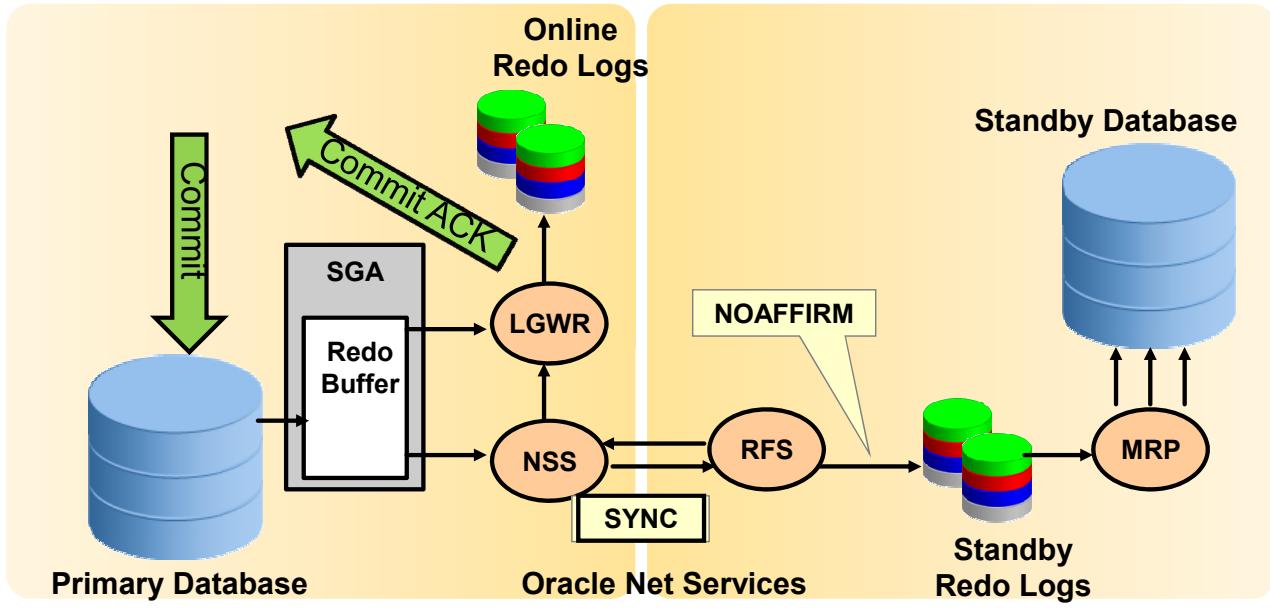
The maximum performance mode is the default mode and provides the highest level of data protection without affecting primary database performance. No specific redo transport settings are required, and standby redo logs are not required for this protection mode.

The maximum protection mode ensures that no data loss will occur, even at the expense of the availability of the primary database. It requires a restart of the primary database and at least one reachable standby database using synchronous transport (SYNC) with affirmation of the standby I/O (AFFIRM) and standby redo logs. Any error that prevents the redo from being written to at least one synchronized standby database causes the primary database to shut down.

The maximum availability mode provides the highest level of data protection without compromising the availability of the primary database. It has the same redo transport setting requirements as the maximum protection mode, but it does not shut down the primary database in the event of an error in transporting redo.

## Data Guard 12c: Fast SYNC

The maximum availability protection mode can now use the SYNC redo transport with the NOAFFIRM settings in a zero-data-loss configuration.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The maximum availability protection mode defines a zero-data-loss configuration as long as at least one synchronized standby database can be maintained. If the primary database cannot write its redo stream to at least one synchronized standby database, it operates as if it were in maximum protection mode to preserve the primary database availability until it is able to resume the write of its redo stream back to a synchronized standby database. This mode ensures that no data loss occurs if the primary database fails, but only if a second fault does not prevent a complete set of redo data from being sent from the primary database to at least one standby database. In Oracle Database 12c, the maximum availability mode can now operate with both SYNC/AFFIRM and SYNC/NOAFFIRM transport settings.

In the default transport mode (SYNC/AFFIRM), the primary is blocked until redo is transmitted synchronously to the physical standby and written to disk. In this case, there is no data loss after a single point failure because the data is recorded persistently on both the primary and the standby. Even if both the primary and standby fail, the data is still available on the standby when the system is recovered.

When a transport is performed by using SYNC/NOAFFIRM, the primary is blocked only until the data is received on the standby, not until it is written to disk. Therefore, it is possible for a small amount of data to be lost if multiple failures affect both the primary and the standby. If the primary fails and the standby is unable to write the in-memory data to disk, that data can be lost.

## Data Guard 12c: Configuring Fast SYNC

Configuring Oracle Data Guard 12c Fast Sync:

- Configure redo transport at the primary database:

```
LOG_ARCHIVE_DEST_2="service=db2_tns db_unique_name=db2 sync noaffirm"
```

- Enable the maximum availability protection mode:

```
ALTER DATABASE SET STANDBY DATABASE TO MAXIMIZE AVAILABILITY;
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There is no change in the syntax or initialization parameters to use the Data Guard 12c Fast Sync feature. The feature is enabled when the `LOG_ARCHIVE_DEST_n` parameter uses the `SYNC` and `NOAFFIRM` attributes and the maximum availability mode is enabled. This combination resulted in an error message in Oracle Database 11g, release 2 if at least one `LOG_ARCHIVE_DEST_n` parameter was not set to use the `SYNC` and `AFFIRM` attributes.

## Benefits: Fast Sync

- Reduces SYNC performance impact on the primary database
  - Removes standby redo log I/O from sync round-trip time
  - Eliminates the impact of synchronous transport on the primary database if network round-trip latency (RTT) is less than the time for a local redo log write
- Provides more predictable primary database performance
  - Standby I/O issues do not impact primary database.
- Enhances data protection
- Enables increased distance between primary and synchronous destinations
  - Primary and standby databases
  - Primary and far sync



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

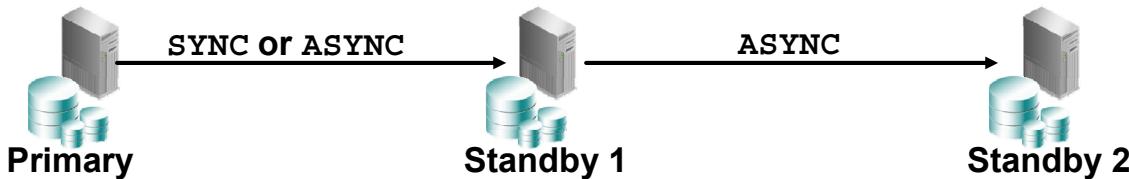
When you perform a transport by using SYNC/AFFIRM, the primary performs write operations and waits for acknowledgment that the redo was transmitted synchronously to the physical standby and written to disk. A SYNC/AFFIRM transport provides a small protection benefit at the expense of a slight loss of performance.

When you perform a transport by using SYNC/NOAFFIRM, the primary performs write operations and waits only for acknowledgment that the data was received on the standby, not that it was written to disk. A SYNC/NOAFFIRM transport provides a small performance benefit at the expense of a slight loss of protection.

The performance benefit of SYNC/NOAFFIRM comes from removing the time needed to write to the standby redo logs from the synchronous round-trip time. Any disk performance issues that service the standby redo logs are also eliminated and allow for a more predictable behavior in redo transport times. With the reduction in time for synchronous transport from a primary database, it may now be possible to use synchronous redo transport in cases where only asynchronous redo transport was available due to performance impact. The time savings in waiting on redo log writes could be applied to the synchronous redo transport network round-trip latency, providing increased distances between a primary database and a far sync or other standby type.

## Real-Time Cascade

- Traditional cascaded standby database
  - Primary redo is shipped to Standby 1 and then forwarded to Standby 2.
  - The redo is forwarded at the log switch, making Standby 2 always in the past of Standby 1.
- Real-time cascade, new for Data Guard 12c
  - Standby 1 forwards the redo to Standby 2 in real time, as it is received.
  - There is no propagation delay of Standby 2 waiting for a primary log switch.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With traditional cascaded standby database configurations prior to Oracle Database 12c, primary database redo is transported to a standby database and that standby database cascades or forwards the primary redo to additional standby databases. However, traditionally the redo is not immediately cascaded. Primary database redo is written to the standby redo log as it is received at a cascading standby database. This redo was cascaded only after the standby redo log file that it was being written to had been archived locally.

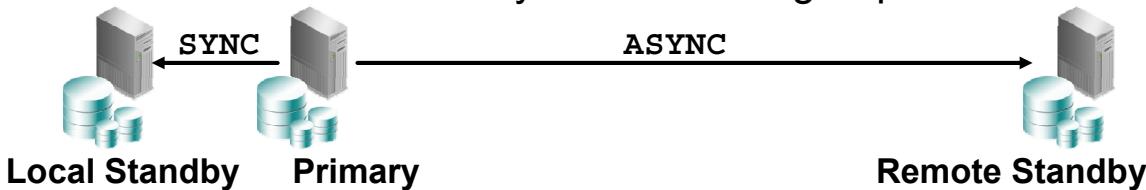
Because standby redo logs are usually configured with the same size as the primary database redo log files, the rate that a standby redo log switches and is archived locally should be about the same time frequency as the primary database log switch rate. Depending on the size of the redo logs and the rate of redo generation, this log switch could take hours. It is also common that the time interval between switches varies. Even when the redo logs were not full, forced log switches on the primary database were typically used to standardize the time interval of log switches. However, the time meant that the final cascaded standby database was guaranteed to lag behind the primary database.

As of Oracle Database 12c, release 1 (12.1), primary database redo is cascaded immediately, as it is being written to the standby redo log file. The terminal destination, therefore, has minimal redo transport lag with respect to the primary database. Only physical standbys and far syncs can cascade redo.

## Benefits: Real-Time Cascade

Multiple standby database configurations, the gold standard of WAN zero-data-loss architectures, consist of:

- A local **SYNC** standby database for zero-data-loss protection
  - Fast, local HA failover with zero data loss
  - Offload RMAN backups, read-only workload, data extracts
  - Real Application Testing using Data Guard Snapshot Standby
  - Database rolling maintenance and standby-first patching
- A remote **ASYNC** standby database for geo-protection



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Multiple standby database configurations are very common in zero-data-loss architectures, where the primary database services one local standby database and one remote standby database. The local standby database may be located in the same data center with high-speed LAN connections using synchronous redo transport. This provides very fast HA failover in the event of a problem with the primary database system. It is also often used for offloading backups, read-only reporting, data extracts, rolling maintenance, and patching.

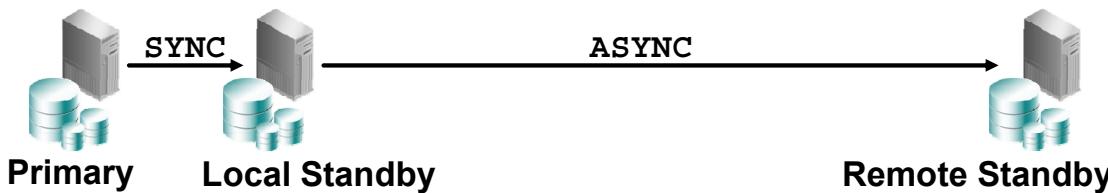
When the standby database is local to the primary database system, one disadvantage is site-wide disasters (for example, hurricanes or tornadoes could potentially impact entire data centers). To achieve zero data loss for all disasters, a second remote standby database is often used. The remote distance often necessitates asynchronous redo transport.

This multiple standby database configuration places additional performance overhead and burden on the primary database system because of the need to ship redo to two different locations. Traditional cascaded standby database configuration could not provide zero data loss for the remote standby database system because of redo transport lag.

## Benefits: Real-Time Cascade

Multiple standby database configurations can now use real-time cascade.

- Less performance overhead
  - The primary database ships to a single destination.
- Less network volume at the primary database



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With real-time cascade introduced in Oracle Database 12c, the terminal destination has minimal redo transport lag for the primary database. Therefore, the local standby can be used to cascade the redo to the remote standby. This reduces the volume of redo sent from the primary by half, which results in less performance impact on the primary database.

## Road Map

### Oracle Database 12c: Data Guard New Features

- Far Sync and Data Guard Transport Enhancements
- **Active Data Guard**
- Database Rolling Upgrades
- Other Data Guard Enhancements
- Data Guard Broker



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next category of enhancements that we will examine is those features that relate to Active Data Guard. Active Data Guard allows redo apply to be active while a physical standby database is open, thus allowing queries to return results that are identical to what would be returned from the primary database. The capability is also known as the real-time query feature.

## Active Data Guard: DML on Temporary Tables

Supported by a new initialization parameter in Oracle Database 12c for all databases:

```
TEMP_UNDO_ENABLED = false | true
```

- Separates undo for temporary tables from undo for the persistent table
  - Temporary undo is not logged in redo.
  - Temporary undo uses the temporary tablespace.
- Enables DML on temporary tables when using Active Data Guard
  - It is set by default on an Active Data Guard standby database.
  - Data definition language (DDL) to create temporary tables must be issued on the primary database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Redo generation on a read-only database is not allowed. When a data manipulation language (DML) operation makes a change to a global temporary table, the change itself does not generate redo because it is only a temporary table. However, the undo generated for the change does, in turn, generate redo. Prior to Oracle Database 12c, release 1 (12.1), global temporary tables could not be used on Active Data Guard standbys, which are read-only.

However, in Oracle Database 12c, release 1 (12.1), the temporary undo feature allows the undo for changes to a global temporary table to be stored in the temporary tablespace instead of in the undo tablespace. Undo stored in the temporary tablespace does not generate redo, thus enabling redo-less changes to global temporary tables. This allows DML operations on global temporary tables on Oracle Active Data Guard standbys.

To enable temporary undo on the primary database, use the TEMP\_UNDO\_ENABLED initialization parameter. On an Active Data Guard standby, temporary undo is always enabled by default, so the TEMP\_UNDO\_ENABLED parameter has no effect.

The temporary undo feature requires that the database initialization parameter COMPATIBLE be set to 12.0.0 or higher. The temporary undo feature on Active Data Guard instances does not support temporary binary and character large objects.

## Active Data Guard: Support for Global Sequences

- Sequences created using the default CACHE and NOORDER options can be accessed from an Active Data Guard standby database.
- When first accessed by the standby, the primary allocates a unique range of sequence numbers.
- When all sequences within a range have been used, the standby requests another range of numbers.
- Because each range assigned to a standby is unique, there is a unique stream of sequences across the entire Data Guard configuration.
- Sequences created with the ORDER and NOCACHE options cannot be accessed on an Active Data Guard standby database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To support read-mostly applications by using Active Data Guard, you might have to use sequences with global temporary tables. In an Active Data Guard environment, sequences created by the primary database with the default CACHE and NOORDER options can be accessed from standby databases as well. When a standby database accesses such a sequence for the first time, it requests that the primary database allocate a range of sequence numbers. The range is based on the cache size and other sequence properties specified when the sequence was created. Then the primary database allocates those sequence numbers to the requesting standby database by adjusting the corresponding sequence entry in the data dictionary. When the standby has used all numbers in the range, it requests another range of numbers.

Because the standby's requests for a range of sequences involve a round-trip to the primary, be sure to specify a large enough value for the CACHE keyword when you create a sequence for an Active Data Guard standby. Otherwise, performance could suffer.

In addition, the terminal standby should have a defined LOG\_ARCHIVE\_DEST\_n parameter that points back to the primary.

## Active Data Guard: Support for Session Sequences

- Session sequences are specifically designed for use with global temporary tables that have session visibility
  - They return a unique range of sequence numbers within a session.
  - Session sequences are not persistent. The state of the session sequences accessed during a session is lost when the session terminates.
- Session sequences are created by the primary database and are accessed on any read-write or read-only database.
- To create a session sequence:

```
SQL> CREATE SEQUENCE ... SESSION;
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A session sequence is a special type of sequence that is specifically designed to be used with global temporary tables that have session visibility. Unlike existing regular sequences (referred to as “global” sequences for the sake of comparison), a session sequence returns a unique range of sequence numbers only within a session, not across sessions. Session sequences are not persistent. If a session goes away, so does the state of the session sequences that were accessed during the session.

Session sequences support most of the sequence properties that are specified when the sequence is defined. However, the CACHE/NOCACHE and ORDER/NOORDER options are not relevant to them and are ignored.

Session sequences must be created by a read-write database, but they can be accessed on any read-write or read-only database (either a regular database temporarily open as read-only or a standby database).

## Benefits: Temporary Undo and Sequences

- Reporting and other applications that are generally read-only but require nonpersistent write access to the database can be run on an Active Data Guard standby by using temporary tables.
- Temporary undo reduces the redo volume if it is also enabled on the primary database. Temporary undo:
  - Is not logged in redo
  - Improves primary database performance
  - Reduces network bandwidth consumption
  - Reduces standby I/O
- Applications that are read-only except for the requirement to generate unique sequences can be offloaded to an Active Data Guard standby database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Prior to Oracle Database 12c, release 1 (12.1), the inability to use global temporary tables and sequences for Active Data Guard limited some read-mostly applications from being offloaded from the primary database to a standby database.

The new temporary undo and sequences support for Active Data Guard provides many benefits. Additional reporting workload can now be migrated to a standby system to reduce the overhead of system resources in general. Because temporary undo reduces the amount of redo generated, the amount of redo needed to be shipped to standby database systems is reduced and results in network performance improvements. The reduction in redo also implies less redo being written to standby redo logs and local archiving of standby redo logs. Therefore, performance improvements are realized on the primary and standby database systems and on the network between the two systems.

## Road Map

### Oracle Database 12c: Data Guard New Features

- Far Sync and Data Guard Transport Enhancements
- Active Data Guard
- **Database Rolling Upgrades**
- Other Data Guard Enhancements
- Data Guard Broker



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next category of enhancements that we will examine is improvements made to database rolling upgrades.

# Data Guard Simple Rolling Upgrades

Goal: Simple, Reliable, Repeatable

- Early problem detection
- Dedicated interface: DBMS\_ROLLING PL/SQL package
- Centralized, simplified, and uniform execution
- Fault tolerance
- Configuration rollback
- Centralized monitoring: DBA\_ROLLING\_\* views



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Guard has supported rolling database upgrades in versions prior to Oracle Database 12c, release 1 (12.1), but the process involved many steps. A simple two-site configuration required jumping at least 6 times between the primary and standby databases when executing instructions. At least 17 DDL commands were needed to progress through the upgrade, and it was necessary to issue 12 queries against 6 different views to validate completion of the steps. In all, a traditional Data Guard rolling upgrade required at least 42 steps to perform.

Using Data Guard to perform a rolling upgrade has been simplified and improved with Oracle Database 12c, release 1 (12.1). The Data Guard simple rolling upgrade process is managed through a new package called DBMS\_ROLLING by creating a centralized plan with the DBMS\_ROLLING.INIT\_PLAN procedure that contains default and user-customized parameters for the rolling upgrade. Additional standby databases can be employed to protect both the database at the previous version and the database at the newer target version for full fault tolerance during the rolling upgrade. A rolling upgrade can be aborted using the DBMS\_ROLLING.ABORT procedure, which can restore the transient logical standby back to the original physical standby role by performing a database flashback.

Six new views are provided to monitor the progress of a rolling upgrade:

DBA\_ROLLING\_DATABASES, DBA\_ROLLING\_EVENTS, DBA\_ROLLING\_PARAMETERS, DBA\_ROLLING\_PLAN, DBA\_ROLLING\_STATISTICS, and DBA\_ROLLING\_STATUS.

## Rolling Upgrades of Database Software

- Database software upgrades using the DBMS\_ROLLING PL/SQL package will be usable starting with the first patchset of Oracle Database 12c.
- Database upgrades from Oracle Database 11g to Oracle Database 12c will need to use the classic Transient Logical Standby upgrade procedure.
- The DBMS\_ROLLING PL/SQL package can be used for other database maintenance tasks with the first release of Oracle Database 12c, such as:
  - Adding partitioning to nonpartitioned tables
  - Changing BasicFiles LOBs to SecureFiles LOBs
  - Changing XML-CLOB to binary XML
  - Altering a table to be OLTP-compressed



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Rolling Upgrade Using Active Data Guard feature, new as of Oracle Database 12c, release 1 (12.1), provides a streamlined method of performing rolling upgrades. It is implemented using the new DBMS\_ROLLING PL/SQL package, which allows you to upgrade the database software in a Data Guard configuration in a rolling fashion. The Rolling Upgrade Using Active Data Guard feature requires the Oracle Active Data Guard option.

You can use this feature for database version upgrades starting with the first patchset of Oracle Database 12c. This means that the manual Transient Logical Standby upgrade procedure must still be used when upgrading from Oracle Database 11g to Oracle Database 12c, or when upgrading from the initial Oracle Database 12c release to the first patchset of Oracle Database 12c.

Additionally, you can use this feature immediately for other database maintenance tasks beginning with Oracle Database 12c, release 1 (12.1). The database where maintenance is performed must be operating at a minimum of Oracle 12.1. Such maintenance tasks include adding partitioning to nonpartitioned tables, changing BasicFiles LOBs to SecureFiles LOBs, changing XMLType stored as CLOB to XMLType stored as binary XML, and altering tables to be OLTP-compressed.

## DBMS\_ROLLING: Concepts

- Rolling changes can be applied on the whole Data Guard configuration.
  - Three stages: Specification, Compilation, and Execution
  - Execution has three phases: Start, Switchover, and Finish
- Two key groups:
  - Leading group (LG)
    - These databases are upgraded first (before switchover).
    - The LG has a master database (the future primary database).
  - Trailing group (TG)
    - These databases are upgraded last (after switchover).
    - The TG has a master database (the original primary database).



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Rolling changes can be applied to the whole Data Guard configuration using all standby databases identified by the `LOG_ARCHIVE_CONFIG` parameter. There are three stages to the rolling upgrade process using the `DBMS_ROLLING` PL/SQL Package: Specification, Compilation, and Execution. In the Specification stage, you identify how you want to implement the rolling upgrade process and designate which standby will become the future primary database. The Compilation stage builds an upgrade plan by performing validation checks and reports any errors that require corrective actions before the Execution stage begins. The Execution stage performs the actual upgrade and has three phases to it: Start, Switchover, and Finish.

Conceptually, the rolling upgrade process splits the Data Guard configuration into two groups: the leading group (LG) and the trailing group (TG). The leading group contains a master database that is designated to be the future primary database. The leading group can contain additional standby databases designed to protect the master database during the upgrade process. Databases in the leading group are upgraded first before switchover. The trailing group contains the original primary database along with any additional databases designed to protect it during the upgrade process. Databases in the trailing group are upgraded last after the switchover is performed.

## DBMS\_ROLLING: Concepts

- Leading group
  - The Leading Group Master database (LGM) must be identified during Specification.
  - The LGM starts as a physical standby, converted into a logical standby (START), and then becomes the primary database (SWITCHOVER).
  - Other databases in the leading group protect the LGM.
  - LGM responsibility is transferrable on failure.
- Trailing group
  - The TG contains the original primary database (Trailing Group Master [TGM]).
  - Other databases in the Trailing Group protect the TGM.
  - TGM responsibility is transferrable on failure.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The leading group contains the designated future primary database, known as the Leading Group Master database (LGM), and the physical standbys that you can configure to protect the designated future primary. All other standbys in the leading group can only be physical standbys. The LGM is first converted into a logical standby database and then the new database software is installed on it and the upgrade process is run. Other standby databases in the leading group also must have their software upgraded at this point. In the event of a failure during the upgrade process, you can fail over to any of the additional standby databases in the leading group and then designate the failover target database to take over the role of the LGM.

The trailing group contains the original primary database known as the Trailing Group Master database (TGM) and standby databases that will protect the original primary during the rolling upgrade process. While the databases in the leading group are going through the upgrade process, user applications can still be connected to the original primary and making changes. The trailing group databases continue running the old database software until all the databases in the leading group are upgraded and the future primary has caught up with the original primary by applying the changes that were generated at the original primary database during the upgrade window. New software is then installed on the databases that are part of the trailing group after switchover, and they are reinstated into the configuration as standbys to the new primary database. The role of TGM can be transferred to other standby databases in the trailing group in the event a failure occurs during the upgrade.

## DBMS\_ROLLING: Key Features

- Specify – Compile – Execute protocol
  - Configuration errors are detected during the Compilation stage.
  - Runtime errors are detected during the Execution stage.
- State is kept in the database.
  - Provides robustness
- Runtime steps are constant.
  - Regardless of how many databases are involved
- Handles failure at the original primary database
- Allows data protection for the upgraded primary right from the start



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The DBMS\_ROLLING package provides for a Specify – Compile - Execute protocol, all driven from a centralized interface using procedures. Configuration errors that would prevent execution of the upgrade plan can be identified during the Compilation stage and resolved before beginning the actual upgrade. DBA\_ROLLING\_\* views provide diagnostic information should any runtime errors occur during the Execution stage.

Plan parameters are persisted in the database, and remain even after completion of a rolling upgrade. After the rolling upgrade has been successfully executed, you can remove your rolling upgrade specification by calling the DBMS\_ROLLING.DESTROY\_PLAN procedure.

In the previous versions of Oracle Data Guard, a rolling upgrade using a transient logical standby database required at least 42 steps for a simple two-database configuration. The number of steps would significantly increase for each additional standby database contained in the Data Guard configuration. Using the DBMS\_ROLLING package, the runtime steps of the Execution stage are constant, regardless of how many standby databases exist.

The concept of leading group standbys and trailing group standbys allow for multiple databases in each group to provide fault tolerance for the group. In addition, the additional databases, if present, can provide protection for the upgraded primary immediately following the upgrade.

# Database Rolling Upgrade: Specification and Compilation Stages

Generate an *upgrade plan*:

- Specify parameters of the rolling upgrade, such as target software versions, participating databases, apply lag requirements, and logging levels.
  - DBMS\_ROLLING.INIT\_PLAN procedure
  - DBMS\_ROLLING.SET\_PARAMETER procedure
  - DBA\_ROLLING\_PARAMETERS view
- Use the DBMS\_ROLLING.BUILD\_PLAN procedure to generate an upgrade plan and perform validations.
- Use the DBA\_ROLLING\_PARAMETERS, DBA\_ROLLING\_PLAN, and DBA\_ROLLING\_EVENTS views to display the current plan and diagnose any problems with the plan.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Specification and Compilation stages involve the following six steps to create a centralized upgrade plan that drives the rolling upgrade process:

1. Initialize the upgrade parameters.
2. View the current upgrade parameters.
3. Modify the upgrade parameter values as necessary.
4. Build the upgrade plan.
5. View the current plan.
6. Revise the upgrade plan as necessary.

The DBMS\_ROLLING.INIT\_PLAN procedure generates system default parameters for all databases specified in the DG\_CONFIG parameter. You can adjust parameters with the DBMS\_ROLLING.SET\_PARAMETER procedure. All parameters for the rolling upgrade are visible with the DBA\_ROLLING\_PARAMETERS view.

When you finalize the parameters, use the DBMS\_ROLLING.BUILD\_PLAN procedure to generate the actual plan and perform validation against the plan. If the validation identifies any errors with the plan, the DBMS\_ROLLING\_EVENTS view displays the errors that need attention.

## Specification Stage Examples

- Initialize the upgrade parameters:

```
SQL> exec DBMS_ROLLING.INIT_PLAN(future_primary=>'boston');
```

- View the current upgrade parameter values:

```
SQL> select scope, name, curval from dba_rolling_parameters order by
      scope, name;
      SCOPE          NAME           CURVAL
-----  -----
seattle        INVOLVEMENT     FULL
              SWITCH_LGM_LAG_WAIT  60
...
...
```

- Configuring the plan to wait for the apply lag to fall below 60 seconds before switching over to the future primary:

```
SQL> exec DBMS_ROLLING.SET_PARAMETER('SWITCH_LGM_LAG_WAIT','1');
SQL> exec DBMS_ROLLING.SET_PARAMETER('SWITCH_LGM_LAG_TIME','60');
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Plan parameters must be initialized to system-generated default values before they can be customized. To initialize plan parameters, call the DBMS\_ROLLING.INIT\_PLAN procedure. This procedure identifies the DB\_UNIQUE\_NAME of the future primary database (that is, the Leading Group Master or LGM). The INIT\_PLAN procedure returns an initial set of system-generated plan parameters. Once the database-related parameters have been defined, the INIT\_PLAN procedure defines operational parameters with system-supplied defaults. In most cases, the plan parameters will be ready for plan validation; however, to ensure that they meet your needs, you should review each parameter. Plan parameters are persisted in the database until you call the DESTROY\_PLAN procedure to remove all states related to the rolling upgrade.

You can query the DBA\_ROLLING\_PARAMETERS view to see the plan parameters and their current values. Plan parameters are either global or local in scope. Global parameters are attributes of the rolling upgrade as a whole and are independent of the database participants. Global parameters have a NULL value in the SCOPE column. Local parameters are associated with a specific database name in the SCOPE column.

To modify any existing rolling upgrade parameter, use the DBMS\_ROLLING.SET\_PARAMETER PL/SQL procedure.

## Compilation Stage Examples

- Build the upgrade plan:

```
SQL> exec DBMS_ROLLING.BUILD_PLAN;
```

- View the current upgrade plan:

```
SQL> select instid, target, phase, description from dba_rolling_plan;
```

INSTID	TARGET	PHASE	DESCRIPTION
1	seattle	START	Verify database is a primary
2	seattle	START	Verify MAXIMUM PROTECTION is disabled
3	boston	START	Verify database is a physical standby
4	boston	START	Verify physical standby is mounted
5	seattle	START	Verify server parameter file exists and is modifiable
...			



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After all the necessary parameters are specified, you build an upgrade plan. An upgrade plan is a custom generated set of instructions that guides your Data Guard configuration through a rolling upgrade. To build an upgrade plan, use the DBA\_ROLLING.BUILD\_PLAN PL/SQL procedure. This procedure requires the configuration to be exactly as described by the plan parameters with all of the instances started and reachable through the network. There are no arguments to specify, because the procedure gets all its input from the DBA\_ROLLING\_PARAMETERS view. The procedure validates plan parameters and performs site-specific validations of resources such as log transport and flash recovery area settings. In general, configuration settings that do not meet best-practice criteria generate a warning message.

After the BUILD\_PLAN procedure successfully returns, the complete upgrade plan is viewable in the DBA\_ROLLING\_PLAN view. Each record in the view identifies a specific instruction that is scheduled for execution and recorded in the DBA\_ROLLING\_EVENTS view.

## Database Rolling Upgrade: Execution Stage

- Call `DBMS_ROLLING.START_PLAN` to configure the primary and standby databases participating in the upgrade.
- Upgrade the RDBMS software for leading group databases.
- Call `DBMS_ROLLING.SWITCHOVER` to swap roles between the current primary database and the new primary database. Switchover is the only required down time.
- Restart the former primary and any bystander standby databases by using new binaries.
- Call `DBMS_ROLLING.FINISH_PLAN` to complete the upgrade of the former primary and any bystanders and resynchronize with the new primary.
- No arguments are needed on these three procedures.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Execution stage of the rolling upgrade process involves the following five steps:

1. Call the `DBMS_ROLLING.START_PLAN` procedure to configure the future primary and physical standbys designated to protect the future primary.
2. Manually upgrade the Oracle Database software at the future primary database and standbys that protect it.
3. Call the `DBMS_ROLLING.SWITCHOVER` procedure to switch roles between the current primary database and future primary database.
4. Manually restart the former primary and remaining standby databases on the higher version of the Oracle Database software.
5. Call the `DBMS_ROLLING.FINISH_PLAN` procedure to convert the former primary to a physical standby and to configure the remaining standby databases for recovery of the upgrade redo.

No arguments are required for the `START_PLAN`, `SWITCHOVER`, and `FINISH_PLAN` procedures.

## Logical Standby: New Data Type Support

Support for more data types helps eliminate the barriers to utilizing database rolling upgrades with transient logical standbys. New types include:

- Abstract data types (ADTs) and ADT tables
- Database file system (DBFS)
- LOBs stored as SecureFiles
- Objects stored as VARRAYs (except for Collections)
- Oracle SecureFiles (deduplication)
- Oracle Text
- Spatial and multimedia (with exceptions)
- User-defined types
- XDB



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The rolling upgrade process converts a physical standby database to a transient logical standby database. While physical standby databases support all Oracle Database data types, a logical standby database has restrictions on supported data types. If the primary database uses the restricted data types, the Data Guard rolling upgrade process may not be feasible.

To help eliminate this barrier for using Data Guard to perform rolling upgrades with minimal down time, Data Guard supports the following data types in a logical standby database:

- Abstract data types (ADTs) and ADT tables
- Database file system (DBFS)
- LOBs stored as SecureFiles
- Objects stored as VARRAYs (except for Collections)
- Oracle SecureFiles (deduplication)
- Oracle Text
- Spatial (except MDSYS.SDO\_GEOGRAPHY and MDSYS.SDO\_TOPO\_GEOMETRY) and multimedia (Opaque type restrictions and REFS are not supported)
- User-defined types
- XDB

## Road Map

### Oracle Database 12c: Data Guard New Features

- Far Sync and Data Guard Transport Enhancements
- Active Data Guard
- Database Rolling Upgrades
- **Other Data Guard Enhancements**
- Data Guard Broker



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

We will now examine several miscellaneous Data Guard enhancements.

## Simpler Role Transitions

- It is no longer necessary to shut down all but one primary Oracle RAC database instance when performing a switchover to a physical standby database.
- A new verify clause confirms switchover readiness:

```
SQL> ALTER DATABASE SWITCHOVER TO CHICAGO VERIFY;
```

- Returns a “Database Altered” message if ready for switchover
- Returns an error message if not ready for switchover
- General role transition enhancements include:
  - New single-command switchover and failover DDLs, which are simpler processing, faster, more reliable
  - Use the FORCE clause if the initial switchover command is not successful.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Role transitions, which include switchover and failover, have been simplified. Effective with Oracle Database 12c, release 1 (12.1), when you perform a switchover from an Oracle RAC primary database to a physical standby database, you no longer have to shut down all but one primary database instance.

Prior to Oracle Database 12c, release 1 (12.1), role transitions such as switchover required “ALTER DATABASE COMMIT TO SWITCHOVER ...” commands to be issued on both the primary and standby databases, with queries and shutdown statements surrounding them. These role transitions have been replaced with a new “ALTER DATABASE SWITCHOVER TO <TARGET>” clause.

A VERIFY option has been added to the “ALTER DATABASE SWITCHOVER ...” clause that confirms whether both the primary and standby databases are ready for switchover. If not, an error message is displayed to help resolve the issue.

If a switchover command fails during the process, an invalid configuration could result in two standby databases and no primary database. A FORCE clause has been added to the “ALTER DATABASE SWITCHOVER ...” command to convert either one of the two standbys to a physical standby and recover from the invalid configuration.

## Support for Moving Online Data Files

- Data Guard supports MOVE of online data files.

```
SQL> ALTER DATABASE MOVE DATAFILE ...
```

- Online move operations can be performed independently on either primary or standby.
  - A move on one does not affect the other.
- Active Data Guard is required to move an online data file on a physical standby when recovery is active.
- The physical standby database must be open as read-only.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can move the location of an online data file from one physical file to another physical file while the database is actively accessing the file. To do so, you use the `ALTER DATABASE MOVE DATAFILE` SQL statement.

An operation performed with the `ALTER DATABASE MOVE DATAFILE` statement increases the availability of the database, because it does not require the database to be shut down to move the location of an online data file. In releases prior to Oracle Database 12c, you could only move the location of an online data file if the database was down or not open or if you first took the file offline.

You can perform an online move data file operation independently on the primary database and on the standby database (either physical or logical). The standby is not affected when a data file is moved on the primary and vice versa.

On a physical standby, an online move data file operation can be executed while standby recovery is running if the instance that opens the database is in read-only mode.

## Support for Separation of Duties

- SYSDG is the administrative privilege that is specific to Data Guard.
- Enables the following privileges:

— STARTUP	— SELECT ANY DICTIONARY
— SHUTDOWN	— SELECT
— ALTER DATABASE	• X\$ tables
— ALTER SESSION	• V\$ and GV\$ views
— ALTER SYSTEM	• APPQOSSYS.WLM_CLASSIFIER_PLAN
— CREATE RESTORE POINT	— DELETE
— CREATE SESSION	• APPQOSSYS.WLM_CLASSIFIER_PLAN
— DROP RESTORE POINT	— EXECUTE
— FLASHBACK DATABASE	• SYS.DBMS_DRS
- The SYSDG privilege enables connection to a database even if it is not open.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For better separation of duty, Oracle Database 12c, release 1 (12.1) provides a set of administrative privileges that are tailored for specific administrative tasks: backup and recovery, Data Guard, and encryption key management for transparent data management. SYSDG, an administration privilege that is specific to Data Guard, has been added to handle standard administration duties for Data Guard. A list of those privileges included with the SYSDG administrative privilege is shown in the slide.

In previous releases, you needed to have the SYSDBA privilege to perform these tasks. To support backward compatibility, you still can use the SYSDBA privilege for these tasks, but Oracle recommends that you use the specific separation of duties privileges.

## Data Guard Support for Oracle Multitenant

- A multitenant container database (CDB) can have a physical standby database and/or a logical standby database.
- A database role is defined only at the CDB level.
- Individual pluggable databases (PDBs) do not have their own roles.
- Role transitions are executed at the CDB level.
- DDL related to role changes is executed in the root container (CDB\$ROOT) of the CDB.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Guard includes support for the multitenant architecture in Oracle Database 12c, release 1 (12.1).

Data Guard is managed at the CDB level. Individual PDBs cannot have a different database role than that of the CDB. Role transitions such as switchover and failover are performed at the CDB level. A primary database that is a CDB can have both physical and logical standby databases. You are not required to have the same set of PDBs at the primary database and standby. However, only tables that exist in the same container at both the primary and standby are replicated.

## Road Map

### Oracle Database 12c: Data Guard New Features

- Far Sync and Data Guard Transport Enhancements
- Active Data Guard
- Database Rolling Upgrades
- Other Data Guard Enhancements
- **Data Guard Broker**



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The next category of enhancements that we will examine is those features that relate to the Data Guard broker. The Data Guard broker is a distributed management framework that automates and centralizes the creation, maintenance, and monitoring of Data Guard configurations.

## Data Guard Broker 12c Enhancements

- Advanced manageability and monitoring
  - Broker validation for role changes
  - Resumable switchover
  - Redo apply lag monitoring
  - Configurable tracing
- Support for new Oracle Database 12c features
  - Far sync
  - Fast sync
  - Real-time cascade
  - Database rolling upgrade
  - Oracle Recovery Server



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Enhancements for Data Guard Broker 12c include enhancements for advanced manageability and monitoring, and support for new Oracle Database 12c, release 1 (12.1) features.

## Broker Validation for Role Changes

- A single command to assess readiness for role change

```
DGMGRL> VALIDATE DATABASE [VERBOSE] database-name;
```

- Automatically performs numerous health checks, including:
  - Validates each database's current status
  - Verifies that there are no archive log gaps
  - Performs a log switch on the primary to verify that the log is applied on all standbys
  - Shows databases or Oracle RAC instances that are not discovered
  - Detects inconsistencies between database properties and values stored in database
  - Ensures that online redo log files were cleared before role transition
  - Checks for previously disabled redo threads
  - Ensures that the primary and all standbys are on the same redo branch



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The VALIDATE DATABASE command can be performed against primary, physical standby, and logical standby databases. It performs a comprehensive set of database checks prior to a role change by using information that is available in the various Data Guard views and in the Automatic Diagnostic Repository.

## Broker Resumable Switchover

- Switchover can still be impacted by problems that occur after a `VALIDATE DATABASE` command is issued.
- In earlier releases, a failed switchover usually required the broker configuration to be deleted and re-created. All actions to extricate from the failed state were initiated on the SQL command line.
- With Data Guard Broker Resumable Switchover, you can:
  - Resolve the problem and reissue broker switchover (It picks up where it left off.)
  - Use the broker to switch back to the original primary while you resolve the problem
  - Use the broker to switch to another standby database in a multiple-standby configuration



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If the switchover fails because of problems with the configuration, the broker reports the problems in the alert log files or in the broker log files. If you can correct the reported problems, you can retry the switchover operation and it will usually succeed.

In some cases, you may not be able to correct the reported problems or the switchover operation may fail even after you correct the reported problems. For those cases, you can choose another database for the switchover or you can restore the configuration to its pre-switchover state and then retry the switchover.

If fast-start failover is enabled, the broker does not allow switchover to any standby database except to the target standby database. In addition, switchover to the target standby database is allowed only when the value of the `FS_FAILOVER_STATUS` column in the `V$DATABASE` view on the target standby database is either `READY` or `SUSPENDED`.

## Broker Automatic Lag Monitoring

- Transport Lag and Apply Lag are included in the SHOW DATABASE output. They quickly assess the metrics that are most critical to recovery point and recovery time objectives.
- Automatic apply lag monitoring is new in Data Guard 12c.
  - It is a configurable database property that sets a threshold for apply lag.
  - The broker automatically monitors the state of the configuration and reports apply lag warnings when the actual lag exceeds the threshold.
  - Lag is expressed in seconds.

```
DGMGRL> EDIT DATABASE boston SET PROPERTY ApplyLagThreshold = 15;
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The ApplyLagThreshold configurable property generates a warning status for a logical or physical standby when the database's apply lag exceeds the value specified by the property. The property value is expressed in seconds. A value of zero seconds results in no warnings being generated when an apply lag exists.

If this property has a non-zero value, health check warnings for snapshot and far sync standby databases are not generated because neither standby runs apply.

## Configurable Broker Tracing

- Previous broker log files contained extensive tracing that was far beyond what is useful for the typical user.
- In Oracle Database 12c the broker has configurable tracing (also back ported to 11.2.0.3).
- The `TraceLevel` configuration property controls the tracing:

```
DGMGRL> EDIT CONFIGURATION SET PROPERTY TraceLevel = User | Support;
```

- Sets tracing level for the entire broker configuration
- USER: The default tracing generates minimum output, similar to the level of detail in the alert log, and shows all commands issued and health check errors and warnings.
- SUPPORT: This creates full diagnostic output identical to that of previous releases, which is useful for service requests.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Use the `TraceLevel` configuration property to control the amount of tracing performed by the broker for every member in the configuration. Setting the property to `USER` limits the tracing to completed operations and to any warning or error messages resulting from an operation or health check. Setting the property to `SUPPORT` increases the amount of tracing to include lower-level information needed by Oracle Support Services.

## Broker Support for Far Sync

- It is simple to add a far sync to a broker configuration.

```
DGMGRL> ADD FAR_SYNC boston2 AS CONNECT IDENTIFIER IS boston2.example.com;
DGMGRL> ENABLE FAR_SYNC boston2;
```

- The broker automatically configures redo transport.
  - On the primary, to ship redo to the far sync
  - On the far sync, to ship redo to the failover target
- The broker monitors redo transport on both the far sync and the failover target.
- When you use maximum availability, zero-data-loss failover to the failover target through the far sync is supported.
- Fast-start failover also supports far sync.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The following example shows how to add a far sync instance to a broker configuration:

```
DGMGRL> ADD FAR_SYNC boston2 AS CONNECT IDENTIFIER IS
      boston2.example.com;
Far Sync BOSTON2 added
DGMGRL> ENABLE FAR_SYNC boston2;
Enabled.
DGMGRL> SHOW CONFIGURATION;
Configuration - The SUPER cluster
  Protection Mode: MaxPerformance
  Databases:
    BOSTON1 - Primary database
    BOSTON2 - Far Sync
    LONDON1 - Physical standby database
  Fast-Start Failover: DISABLED
  Configuration Status:
    SUCCESS
```

## Broker Support for Complex Redo Routing

- By default, a primary database sends its redo to every possible redo transport destination in a broker configuration.
- A new `RedoRoutes` broker property allows more complex routing to be defined, and it supports:
  - Cascaded configurations
  - Far Sync configurations
  - Real-time cascading and non-real-time cascading
  - Rules dependent on which database is the current primary
  - The new Fast Sync configuration



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, a primary database sends the redo that it generates to every other redo transport destination in the configuration. You can use the `RedoRoutes` property with Oracle Data Guard 12c to create a more complex redo transport topology, such as one in which a physical standby database or a far sync forwards redo received from the primary database to one or more destinations, or one in which the redo transport mode used for a given destination depends on which database is in the primary role.

## Defining RedoRoutes Using DGMGRL

- The RedoRoutes property is set to a character string that contains one or more redo routing rules.

```
DGMGRL> EDIT DATABASE database-name
      SET PROPERTY 'RedoRoutes' = '(redo routing rule 1)(redo routing rule n)';
```

- Each rule contains one or more redo sources and one or more redo destinations, separated by a colon.
- The redo source field must contain the LOCAL keyword or a comma-separated list of DB\_UNIQUE\_NAME values.
- The redo destination field must contain the keyword ALL or a comma-separated list of database names, each of which can be followed by an optional redo transport attribute.

```
(redo source : redo destination) (LOCAL : redo destination ATTRIBUTE)
(redo source : redo destination, redo destination ATTRIBUTE)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The RedoRoutes property is set to a character string that contains one or more redo routing rules. Each rule contains one or more redo sources and one or more redo destinations. A redo routing rule becomes active when one of the redo sources in the rule is in the primary role. This results in redo from the primary database being sent to every redo destination in that rule. A redo routing rule contains a redo source field and a redo destination field, separated by a colon:

```
(redo source : redo destination)
```

The redo source field must contain the LOCAL keyword or a comma-separated list of DB\_UNIQUE\_NAME values:

```
{LOCAL | db_unique_name_1 [,db_unique_name_n]}
```

You cannot set the RedoRoutes property on a logical or snapshot standby database.

## RedoRoutes Usage Guidelines

- The RedoRoutes property has a default value of NULL, which is treated as (LOCAL : ALL) at a primary database.
- A redo routing rule is active if its redo source field specifies the current primary database.
- If a rule is active, primary database redo is sent by the database at which the rule is defined to each destination specified in the redo destination field of that rule.
- The ASYNC redo transport attribute must be explicitly specified for a cascaded destination to enable real-time cascading to that destination.
- The RedoRoutes property cannot be set on a logical or snapshot standby database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The RedoRoutes property has a default value of NULL, which is treated as (LOCAL : ALL) at a primary database. A redo routing rule is active if its redo source field specifies the current primary database. If a rule is active, primary database redo is sent by the database at which the rule is defined to each destination specified in the redo destination field of that rule. The ASYNC redo transport attribute must be explicitly specified for a cascaded destination to enable real-time cascading to that destination. The RedoRoutes property cannot be set on a logical or snapshot standby database.

## How to Read Redo Routing Rules

If Berlin is the current primary database, the rule is active.



```
DGMGRL> EDIT DATABASE Paris SET PROPERTY 'RedoRoutes' = '(Berlin : Madrid ASYNC);'
```

Which causes Paris to ship redo to Madrid.

This enables real-time cascading.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The diagram in the slide provides an example of interpreting a redo routing rule. The command syntax specifies Paris as the database name for which the RedoRoutes property is being set, Berlin as the redo source, and Madrid as the redo destination. The ASYNC attribute is explicitly specified. If Berlin (redo source) is the current primary database, then the rule is active. An active rule would cause Paris (database name) to ship redo to Madrid (redo destination). The redo transport attribute can be set to SYNC, FASTSYNC, or ASYNC. The ASYNC setting used in the example is required to enable real-time cascading.

## Far Sync Example with RedoRoutes

- An example Far Sync configuration for a Maximum Availability broker configuration:
  - Primary database in Boston, MA
  - Far Sync in Cambridge, MA
  - Physical standby in Chicago, IL
  - Far Sync in Shaumburg, IL (for role reversal)

```
DGMGRL> EDIT DATABASE Boston SET PROPERTY 'RedoRoutes' = '(LOCAL : Cambridge
SYNC)';
DGMGRL> EDIT FAR_SYNC Cambridge SET PROPERTY 'RedoRoutes' = '(Boston : Chicago
ASYNC)';
DGMGRL> EDIT DATABASE Chicago SET PROPERTY 'RedoRoutes' = '(Local : Shaumburg
SYNC)';
DGMGRL> EDIT FAR_SYNC Shaumburg SET PROPERTY 'RedoRoutes' = '(Chicago : Boston
ASYNC)';
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After Data Guard broker has been configured for a primary database, far sync, physical standby, and a second far sync for role reversal, the configuration will appear as follows:

```
DGMGRL> SHOW CONFIGURATION;
Configuration - The SUPER cluster
  Protection Mode: MaxAvailability
  Databases:
    Boston - Primary database
    Cambridge - Far Sync
    Chicago - Physical standby database
    Shaumburg - Far Sync (Inactive)
  Fast-Start Failover: DISABLED
  Configuration Status:
    SUCCESS
```

## Cascading Databases Example with RedoRoutes

- To set up a cascading configuration assuming Berlin (Primary) -> Paris (Standby) -> Madrid (Standby) -> Lisbon (Standby):

```
DGMGRL> EDIT DATABASE Berlin SET PROPERTY 'RedoRoutes' = '(LOCAL : Paris FASTSYNC)';
DGMGRL> EDIT DATABASE Paris SET PROPERTY 'RedoRoutes' = '(Berlin : Madrid ASYNC)';
DGMGRL> EDIT DATABASE Madrid SET PROPERTY 'RedoRoutes' = '(Berlin : Lisbon ASYNC)';
```

- The broker automatically configures redo transport:
  - Berlin ships redo to Paris using Fast Sync while in Maximum Availability protection mode.
  - Paris cascades redo in real time to Madrid while Berlin is the active primary database.
  - Madrid cascades redo in real time to Lisbon while Berlin is the active primary database.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

After Data Guard broker has been configured for multiple cascaded destinations, the configuration will appear as follows:

```
DGMGRL> SHOW CONFIGURATION;
Configuration - The SUPER cluster
  Protection Mode: MaxAvailability
  Databases:
    Berlin - Primary database
    Paris - Physical standby database
    Madrid - Physical standby database
    Lisbon - Physical standby database
  Fast-Start Failover: DISABLED
  Configuration Status:
    SUCCESS
```

## Broker Support for Fast Sync

- FASTSYNC is a new value for the LogXptMode database configurable property.

```
DGMGRL> EDIT DATABASE boston SET PROPERTY 'LogXptMode'='FASTSYNC';
```

- FASTSYNC:
  - Configures redo transport by using the SYNC and NOAFFIRM attributes of the LOG\_ARCHIVE\_DEST\_n initialization parameter
  - Can be used only in maximum availability mode
  - Can be used with fast-start failover



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The LogXptMode configurable property enables you to set the redo transport service. You set the redo transport services on each configuration member to one of the following modes: SYNC, ASYNC, or FASTSYNC. The LogXptMode value of FASTSYNC configures redo transport services for this configuration member by using the SYNC and NOAFFIRM attributes of the LOG\_ARCHIVE\_DEST\_n initialization parameter. This mode is available only in maximum availability protection mode. Because the FASTSYNC transport mode uses the NOAFFIRM attribute of the LOG\_ARCHIVE\_DEST\_n parameter, data loss is possible.

## Using DBMS\_ROLLING Package with Data Guard Broker

- Data Guard Broker 12c coexists with the rolling upgrade process.
  - A single command disables the broker configuration:

```
DGMGRL> DISABLE CONFIGURATION;
```

- Then you perform the rolling upgrade.
- Then you connect to the current primary and use a single command to reenable the broker:

```
DGMGRL> ENABLE CONFIGURATION;
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you use the Data Guard broker to manage a Data Guard configuration and you want to preserve that configuration when you perform a rolling upgrade, you must perform the following:

1. Disable the broker configuration by using the DGMGRL DISABLE CONFIGURATION command.
2. Perform a rolling upgrade.
3. Connect to the current primary database (that is, the database whose control file role is PRIMARY) and reenable the broker configuration by using the DGMGRL ENABLE CONFIGURATION command.

This procedure preserves your broker configuration so that you do not have to rebuild it after a rolling upgrade is completed.

## Quiz

The maximum availability redo protection mode requires the same redo transport attribute settings as does the maximum protection mode.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: b**

## Quiz

In a multitenant container database (CDB), one individual pluggable database (PDB) can be a primary database while another PDB can be a logical standby database.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: b**

## Summary

In this lesson, you should have learned how to describe the Data Guard 12c new features designed to improve the following areas:

- Far Sync
- Data Guard Transport
- Active Data Guard
- Database Rolling Upgrades
- Data Guard Broker



Copyright © 2013, 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

# 12

## Oracle Global Data Services Overview



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Objectives

After completing this lesson, you should be able to:

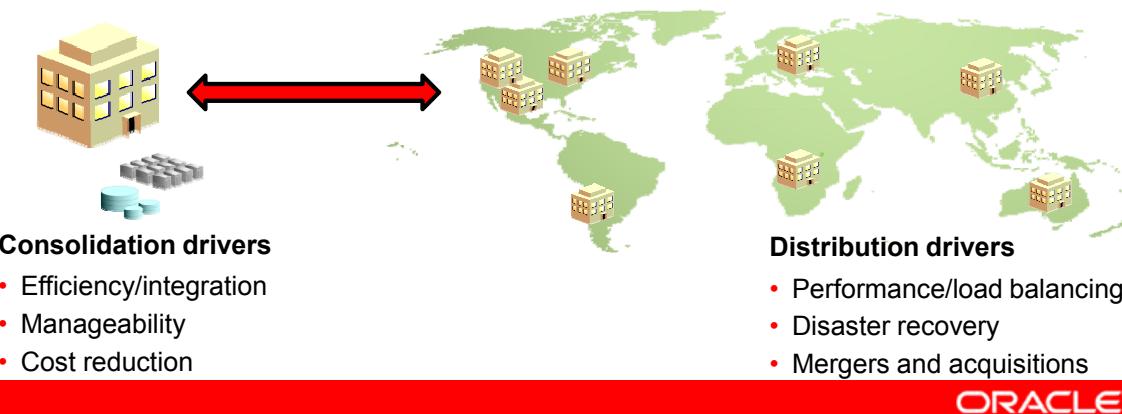
- Explain the benefits provided by Global Data Services for managing cloud-deployed distributed databases
- List the components of the Global Services Framework
- Explain how Global Service connections are load balanced
- Describe the process of Global Services failover
- Describe supported database or replication in common usage scenarios



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

# Global Data Consolidation

- Many organizations are going through consolidation of their IT infrastructure to improve business efficiency.
  - It is impossible to achieve database consolidation solely by aggregating hundreds of databases in a few large ones.
- These organizations still need to maintain multiple replicas of their databases both locally and in geographically disparate regional data centers.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Many large enterprises consolidate their information technology infrastructure to improve business efficiency, and database consolidation is a critical part of this process. In today's global economy, it is impossible to achieve database consolidation only by aggregating hundreds of databases into a few large databases or collocating them in a single data center, or a combination of both solutions.

These enterprises often use technologies such as Oracle Active Data Guard and Oracle GoldenGate for their disaster recovery and replication needs. Distributing workload over multiple databases can bring data closer to clients, improve performance and scalability, and get more value out of IT assets. However, when applications are spread across multiple databases and potentially also across data centers, it can be challenging to efficiently use all your databases for best performance and availability.

# Global Data Services

- Many companies maintain replicas of their databases locally and in geographically disparate data centers.
- Reasons for having both local and global replicas include:
  - Business continuity and disaster recovery
  - Performance optimization for local clients
  - Content localization and caching
  - Compliance with local laws
  - Integration of data centers spurred by mergers/acquisitions
- Some enterprises using Oracle databases implement their own distributed database workload management solutions.
- These generic solutions cannot provide critical functionality like runtime load balancing and reliable failover.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There are multiple business reasons for having both local and global replicas, including:

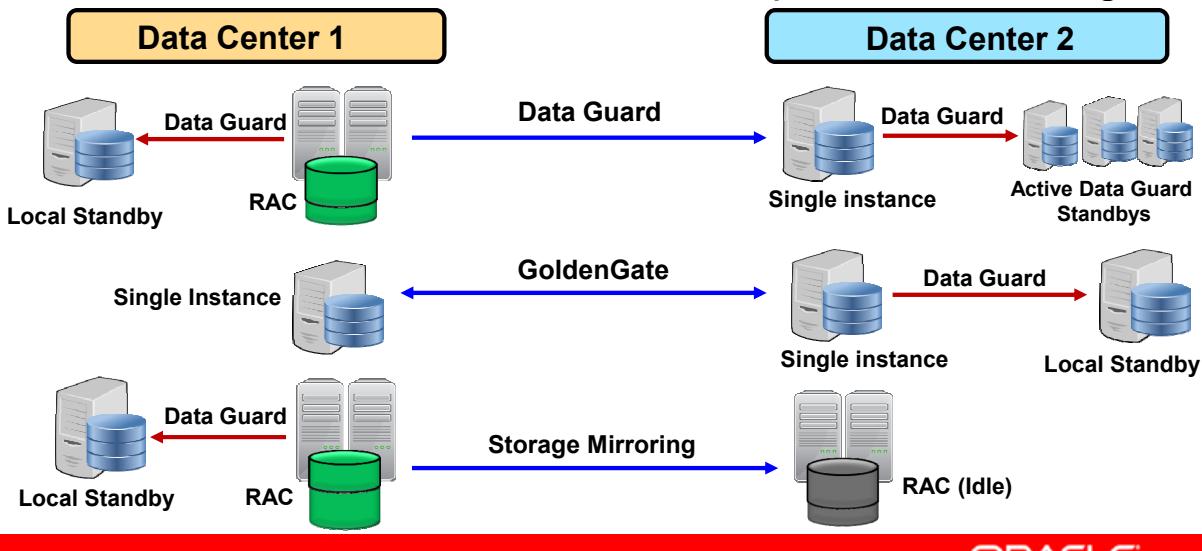
- Business continuity and disaster recovery
- Performance optimization for local clients
- Content localization and caching
- Compliance with local laws
- Integration of data centers obtained through mergers and acquisitions

In a system containing multiple replicated databases, a particular database server may cause a slower response time because of an increased demand for a database service, whereas replica servers capable of offering the same service may be underutilized.

Many large enterprises using Oracle databases implement their own solutions for workload management in distributed database systems. These solutions, however, cannot provide critical functionality, such as runtime load balancing and reliable failover, because they are not integrated with the Oracle software stack.

# Oracle Global Data Services

- Global Data Services (GDS) for database clouds applies the RAC service model to sets of globally distributed databases.
- GDS works with a single instance or RAC databases using Data Guard, GoldenGate, or other replication technologies.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Global Data Services for database clouds applies the Oracle RAC service model to sets of globally distributed, heterogeneous databases, providing dynamic load balancing, failover, and centralized service management for a set of replicated databases that offer common services. The set of databases can include Oracle RAC and noncluster Oracle databases interconnected through Oracle Data Guard, Oracle GoldenGate, or any other replication technology.

Features of Global Data Services enable you to integrate your locally and globally distributed, loosely coupled databases running on heterogeneous platforms into a scalable and highly available private database cloud that can be shared by clients around the globe.

Distributed database systems, in most cases, do not maintain absolute data consistency across replicas. Therefore, not all services that can currently run on multiple instances of an Oracle RAC database can be scaled to run in a multidatabase environment. Global Data Services is primarily intended for applications that are replication aware, use read-only services, or both. Applications that cannot be modified to work with replicated data can still benefit from improved high availability and disaster-recovery capabilities of a distributed database system.

## The Global Data Services Framework

- Global Data Services employs a distributed framework, which:
  - Automates and centralizes configuration, maintenance, and monitoring of a GDS environment
  - Enables load balancing and failover for services
- The framework includes logical and physical components
  - Logical components include:
    - The Global Data Services configuration
    - Global Data Services pools
    - Global Data Services regions
  - Physical components include:
    - Global Service Manager
    - Global Data Services catalog
    - Oracle Notification Service servers
    - Global Data Services control utility (`gdsctl`)



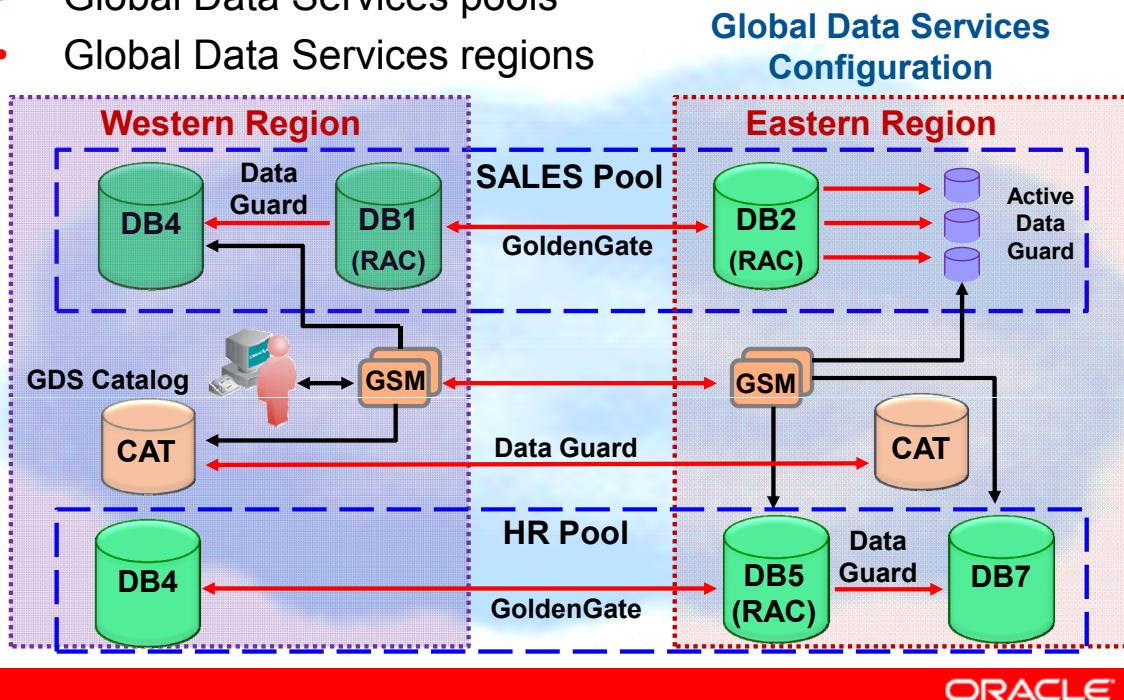
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Global Data Services uses a distributed framework that automates and centralizes configuration, maintenance, and monitoring of a Global Data Services configuration, and enables load balancing and failover for services provided by Global Data Services.

The Global Data Services framework includes logical and physical components. Logical components refer to a group of components that share the same property, including the Global Data Services configuration, Global Data Services pools, and Global Data Services regions. Physical components are the global service manager, the Global Data Services catalog, databases, Oracle Notification Service servers, and the Global Data Services control utility, `gdsctl`.

## Logical Global Data Services Components

- The Global Data Services configuration
- Global Data Services pools
- Global Data Services regions



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

The logical components of Oracle Global Data Services include the Global Data Services configuration itself and the Global Data Services pools and Global Data Services regions contained within the GDS environment. The figure in the slide shows the relationships between the Global Data Services framework logical components using an example of a configuration containing two Global Data Services pools (Sales and HR) and two Global Data Services regions (West and East).

## Logical Global Data Services Components: The Global Data Services Configuration

- A Global Data Services configuration is a set of databases that provide global services.
- Each Global Data Services configuration has a name that the global services administrator can specify.
- The default name is `oradbcloud`.
- A name can contain up to 30 bytes, including valid identifiers.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A Global Data Services configuration is a set of databases that provide global services. The Global Data Services framework is a self-contained system, in which all databases contained within a Global Data Services configuration are managed by Global Data Services components that belong to the same GDS environment.

Every Global Data Services configuration has a name that global service manager can specify. The default name is `oradbcloud`. A name can contain up to 30 bytes, including valid identifiers (an alphabetical character followed by zero or more alphanumeric ASCII characters, the underscore “\_”, or the number sign “#”, and possibly separated by periods if there are multiple identifiers).

## Logical Global Data Services Components: Global Data Services Pool

- Each Global Data Services pool contains replicated databases that provide a common set of global services.
- A database can only belong to a single Global Data Services pool.
- It is not necessary that all databases in a pool provide the same set of global services.
  - An individual database can support only a subset of services provided by the pool.
  - All databases that provide the same global service must belong to the same pool.
- A Global Data Services pool must have a unique name within the GDS configuration.
  - The pool name can be up to 30 bytes long.
  - The default name is `oradbpool`.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A Global Data Services pool is a named subset of databases within a GDS configuration that provides a unique set of global services and belongs to the same administrative domain. Partitioning of GDS configuration databases into pools simplifies service management and provides higher security by allowing each pool to be administered by a different administrator.

A database can only belong to a single Global Data Services pool. All databases in a pool need not provide the same set of global services. However, all databases that provide the same global service must belong to the same pool.

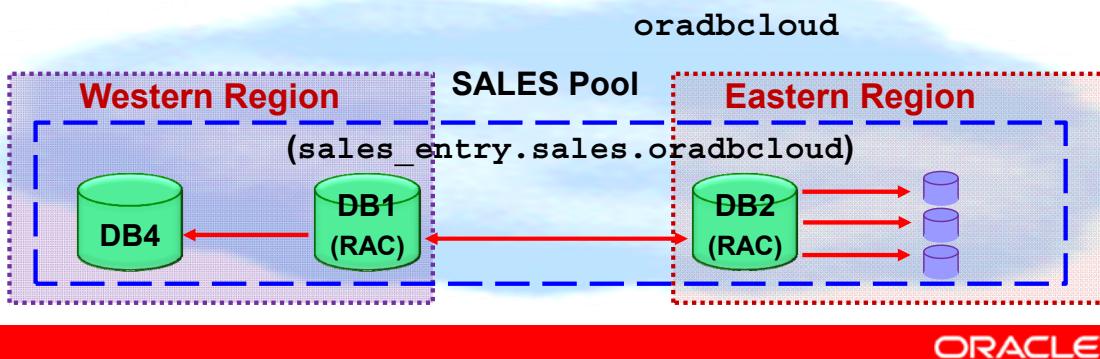
Within a Global Data Services configuration, there can be many different pools of databases that do not share anything other than Global Data Services components. Each Global Data Services pool contains replicated databases that provide a common set of global services and that belong to the same administrative domain.

A database can only belong to a single Global Data Services pool. It is not necessary that all databases in a pool provide the same set of global services; an individual database can support only a subset of the services provided by the pool. However, all databases that provide the same global service must belong to the same pool.

A Global Data Services pool must have a unique name within the GDS configuration. If you do not specify a name for the pool when you create it, the name defaults to `oradbpool`. The pool name can be up to 30 bytes long and can be any valid identifier (an alphabetical character followed by zero or more alphanumeric ASCII characters or the underscore "\_").

## Logical Global Data Services Components: Global Services

- A global service is provided by a set of databases that belong to the same Global Data Services pool.
- A global service must have a unique name.
- If a global service name is not fully qualified at creation, `pool_name.cloud_name` is the default domain.
- In the example below, the service `sales_entry` would be given the fully qualified name:  
`sales_entry.sales.oradbcloud`



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

A global service is provided by a set of databases that belong to the same Global Data Services pool. However, because services in all pools of a Global Data Services configuration are managed by the same Global Data Services framework components, a global service must have a unique name within the GDS configuration. Because pool administrators create services independently, to provide uniqueness, if you attempt to create a fully qualified service name (which includes a service name and a domain) that already exists as a global service in a different pool in the same configuration, then an error results.

If a global service name is not qualified with a domain when you create it, then, instead of using the database domain as the default domain, as is normal for local services, Oracle uses `pool_name.cloud_name` as the default domain. Therefore, a partially qualified service name gets created as the fully qualified name `service_name.pool_name.cloud_name`.

## Logical Global Data Services Components: Global Data Services Region

- A Global Data Services region usually corresponds to a data center or LAN.
- A Global Data Services configuration can span one or more Global Data Services regions.
- A region can have databases belonging to different pools, but the pools should belong to the same GDS configuration.
- A Global Data Services region name should be unique within the corresponding Global Data Services configuration.
- If no name is specified at the first region creation time, the name defaults to `oraregion`.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A Global Data Services region is a named subset of databases in a GDS configuration and database clients that share network proximity such that the network latency between members of a region is typically lower than between members of different regions. A region usually corresponds to a local area or metropolitan area network. For example, a data center hosting one or more GDS configuration databases and database clients in geographical proximity to the data center might belong to the same region.

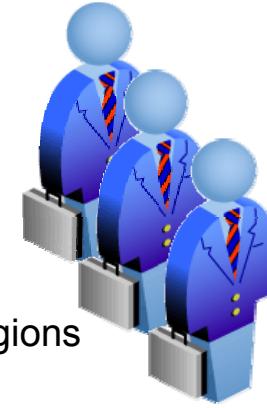
A Global Data Services configuration can span one or more Global Data Services regions. A region can contain databases that belong to different Global Data Services pools, but the pools should belong to the same Global Data Services configuration.

For high-availability purposes, each region in the Global Data Services framework has a buddy region. A Global Data Services region should have a name that is unique within the corresponding Global Data Services configuration. If no name is specified at the first region creation time, the default name, `oraregion`, is given to the region. The region name can be up to 30-characters long and can be any valid identifier: an alphabetical character followed by zero or more alphanumeric ASCII characters or “\_”.

For high-availability purposes, each region in a GDS configuration should have a designated buddy region, which is a region that contains global service managers that can provide continued access to a GDS configuration if the global service managers in the local region become unavailable.

## Physical Global Data Services Components: Global Service Manager

- The global service manager (GSM) is the central component of Global Data Services.
- The GSM performs the following tasks:
  - Acts as a regional listener connecting clients to global services
  - Measures network latency between its own Global Data Services region and all other regions
  - Performs connection load balancing
  - Monitors database instances, and generates and publishes Fast Application Notification (FAN) runtime load-balancing events
  - Maintains Global Data Services framework configuration
  - Manages cardinality and failover of global services
- Each GSM in a Global Data Services configuration manages all of the global services in that configuration.



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The global service manager (GSM) is the central component of Global Data Services. Every global service manager in a Global Data Services configuration manages all of the global services that the configuration provides. A global service manager can be associated with only one Global Data Services configuration. There must be at least one global service manager for each Global Data Services region but, typically, you configure more than one for high availability and improved performance. If there are multiple Global Data Services pools in a GDS configuration, they share all of the global service managers that belong to the configuration. The GSM performs the following tasks:

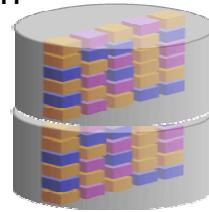
- The GSM acts as a regional listener used by clients to connect to global services. Clients in a Global Data Services region establish database connections to local and remote databases that provide global services through regional global service managers. Client connection requests are randomly distributed among all regional GSMS. A client can connect to any of the GSMS, which redirects the client to an appropriate database according to a connection load-balancing policy. However, when all GSMS in a region are down, global service managers in the buddy region start acting as listeners for the region with failed global service managers. Therefore, the clients' connection string should contain global service managers from the local and buddy regions.
- The GSM measures network latency between its own Global Data Services region and all other regions, and exchanges this information with global service managers in other regions.

- The GSM performs connection load balancing by directing a client connection to the most appropriate database instance. The global service manager determines the appropriate instance based on connection load-balancing metrics that the global service manager receives from all instances, estimated network latency, and region affinity of the service.
- The GSM monitors database instances, and generates and publishes FAN runtime load-balancing events for clients in the local Global Data Services region by combining and normalizing runtime load balancing metrics that it receives from all instances and integrating them with estimated network latency. Each global service manager also generates FAN events when it detects that other global service managers start or stop. When all global service managers in a region are down, runtime load-balancing events are generated by global service managers in the buddy region.
- The GSM maintains Global Data Services framework configuration by making changes to configuration data in all of the Global Data Services framework components and verifying mutual consistency of the data across components.
- The GSM manages cardinality and failover of global services by starting, stopping, and moving the services among instances and databases.

A GSM is associated with one and only one GDS configuration. Each region in the GDS configuration must have at least one GSM. It is recommended that multiple global service managers be configured in each region to improve availability and performance. Every global service manager in a GDS configuration manages all global services supported by the configuration. All GSMS in a Global Data Services region receive runtime load-balancing metrics from all databases in their Global Data Services configuration and measure network latency between regions

## Physical Global Data Services Components: Global Data Services Catalog

- The Global Data Services catalog stores:
  - Configuration data for a Global Data Services configuration
  - All global services provided by the configuration
- There can be only one catalog per GDS configuration.
- A Global Data Services catalog resides in an Oracle Database 12c database.
- The catalog can reside on a database outside the GDS configuration.
- You can use existing high availability (HA) technologies, such as RAC, Data Guard, and Oracle Clusterware, to protect the catalog.
- If GoldenGate is used, ensure that the Global Data Services catalog gets replicated to a secondary database.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Global Data Services catalog is a repository that stores configuration data for a Global Data Services configuration and all global services provided by the configuration. A Global Data Services catalog must be associated with a particular Global Data Services configuration, and there can only be one Global Data Services catalog per configuration. A Global Data Services catalog resides in an Oracle Database 12c database and can be any one of the databases within the Global Data Services configuration. The Global Data Services catalog can also reside on a database outside the Global Data Services configuration.

It is strongly recommended that high availability technologies such as Oracle RAC, Oracle Data Guard, and Oracle Clusterware be used to enhance the availability of the database where the Global Data Services catalog resides.

## Physical Global Data Services Components: Databases

- A global service is provided by a set of databases residing in the same Global Data Services pool.
- The GDS framework makes the pool appear to clients as a single database with many instances.
- Each database in a Global Data Services configuration must be associated with:
  - One Global Data Services region
  - One Global Data Services pool
  - One Global Data Services configuration



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A global service is provided by a set of databases residing in the same Global Data Services pool. Although a Global Data Services pool consists of multiple databases, the GDS framework makes the pool appear to clients as a single database with many instances. Each database in a Global Data Services configuration must be associated with at the most one Global Data Services region, one Global Data Services pool, and one Global Data Services configuration.

## Physical Global Data Services Components: Oracle Notification Servers

- An Oracle Notification Server (ONS) runs with each GSM, delivering FAN events and runtime load-balancing metrics to clients.
- The GSM connects to each database, detects FAN events, and publishes the events to the collocated ONS server.
- RAC ONS servers are only for local services and they are not connected to the ONS servers running with the GSM.
  - Because of this, clients of global services do not subscribe to local ONS servers.
- The GSM connects to each database, detects FAN events, and publishes the events to the collocated ONS servers.
- For each GDS region, all GSMS and ONS servers are fully connected but only the primary GSM publishes FAN events.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In a Global Data Services configuration, an Oracle Notification Service (ONS) server runs with each global service manager, and these servers deliver FAN events and runtime load-balancing metrics to clients. The difference between these ONS servers and those running within Oracle RAC databases is that the ONS servers running in Oracle RAC are only for local services, and they are not connected to the ONS servers running with the global service managers. As a result, clients of global services do not subscribe to local ONS servers.

The global service manager connects to each database, detects FAN events, and publishes the events to the collocated ONS server. For each Global Data Services region, all global service managers and collocated ONS servers are fully connected, but only the primary global service manager publishes FAN events. Global Data Services clients connect to ONS servers of all global service managers in their region and its buddy region; however, ONS servers located in different Global Data Services regions are not connected.

## Physical Global Data Services Components: The gdsctl Utility

- The gdsctl utility provides a command-line interface for configuring and managing the GDS framework.
- To execute a command, gdsctl may need to establish a connection to a:
  - Global service manager
  - Global Data Services catalog database
  - Database in the Global Data Services configuration
- Unless specified, gdsctl resolves connect strings with the current name resolution methods such as TNSNAMES.
  - The exception is the GSM name that gdsctl resolves by querying the gsm.ora file.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The gdsctl utility provides a command-line interface for configuring and managing the Global Data Services framework. To execute a command, gdsctl may need to establish a connection to a global service manager, a Global Data Services catalog database, or a database in the Global Data Services configuration.

## Global Service: Overview

- For database clients, a Global Data Services configuration is represented by a set of global services.
- A GSM serving a Global Data Services configuration is aware of all global services provided by the configuration.
  - It acts as a mediator between database clients and databases in the GDS configuration.
- A client program connects to a regional global service manager and requests a connection to a global service.
- The GSM forwards the client's request to the optimal instance in the GDS configuration that offers the global service.
- The configuration and runtime status of global services are stored in the Global Data Services catalog.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For database clients, a Global Data Services configuration is represented by a set of global services. A global service manager serving a Global Data Services configuration is aware of all global services that the GDS configuration provides and acts as a mediator between database clients and databases in the GDS configuration. A client program connects to a regional global service manager and requests a connection to a global service. The client does not need to specify which database or instance it requires. The global service manager forwards the client's request to the optimal instance in the GDS configuration that offers the global service. Database clients that share a global service must have the same service-level requirements.

The functionality of local services is not changed by global services. Oracle Database 12c can provide local and global services simultaneously. A client application can also work with global and local services simultaneously.

The configuration and runtime status of global services are stored in the Global Data Services catalog. Each database that offers global services also maintains information about those services in a local repository (such as a system dictionary or Oracle Cluster Registry), together with data on local services. Global services that are stored in a local repository have a special flag to distinguish them from traditional local services.

**Note:** Databases earlier than Oracle Database 12c can provide local services, but only Oracle Database 12c, and later, can provide global services.

## Global Service Attributes

- Global services attributes control:
  - Global service startup
  - Load-balancing connections to the global services
  - Failing over those connections
- Local service attributes, including those specific to RAC and Data Guard, are also applicable to global services.
- Attributes unique to global services include:
  - Preferred or available databases
  - Replication lag
  - Region affinity
- You can enable and disable, move, and change the properties of a global service just like a local service.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Global services have a set of attributes associated with them that control starting the global services, load-balancing connections to the global services, failing over those connections, and more. Attributes applicable to local services, including those specific to Oracle RAC and Oracle Data Guard broker environments, are also applicable to global services.

The following attributes are unique to global services:

- Preferred or available databases
- Replication lag
- Region affinity

You can modify global services as you can local services. You can enable and disable a global service, you can move the global service to a different database, and you can change the properties of the global service.

**Note:** You cannot upgrade a local service to a global service.

## Global Services in a RAC Database

- Some properties of a global service are only applicable to RAC databases and are unique for each RAC database.
- These properties are related to placement of global services with instances within a RAC database, including:
  - Server pools and service cardinality for policy-managed databases
  - Distributed transaction processing
- You can specify attributes for these properties by using the `srvctl` utility.
- Global Data Services supports policy-managed RAC databases only.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Some properties of a global service are only applicable to RAC databases and are unique for each RAC database included in a GDS configuration. These properties are related to placement of global services with instances within an Oracle RAC database, including:

- Server pools and service cardinality for policy-managed databases
- Distributed transaction processing

You can specify attributes for these properties using `srvctl`; however, you must manage these properties in an Oracle RAC database. This means that all current and future service placement functionality on Oracle RAC databases will be supported for global services. Local management of database-specific service properties also provides better performance and availability. All other existing global service attributes, like load balancing, role, transparent application failover parameters, and database edition must be the same for all databases offering a global service. You must specify these attributes at the global service level.

By default, in an Oracle RAC environment, a SQL statement executed in parallel can run across all of the nodes in the cluster. The cross-node parallel execution is not intended to be used with GDS load balancing. For an Oracle RAC database in a GDS configuration, it is recommended that you restrict the scope of the parallel execution to an Oracle RAC node by setting the `PARALLEL_FORCE_LOCAL` initialization parameter to `TRUE`.

**Note:** Global Data Services does not support administrator-managed Oracle RAC databases. It supports policy-managed databases only.

## Global Services in an Data Guard Broker Configuration

- When you include a broker configuration in a GDS configuration, broker configurations are managed as a single unit.
  - Only an entire broker configuration can be added to or deleted from a Global Data Services pool.
  - A broker configuration cannot span multiple pools.
- A database is added to the GDS pool by adding it to the broker configuration by using the Data Guard utility `dgmgrl`.
- After adding a database to the broker configuration, run this command to synchronize GDS and Data Guard:

```
$ gdsctl sync brokerconfig
```

- Global services can be configured with a role attribute to be active in a specific role such as primary or physical standby.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Data Guard enables one primary database to be connected to up to 30 standby databases. The Oracle Data Guard broker logically groups these primary and standby databases into a broker configuration that enables the broker to manage and monitor the databases together as an integrated unit. When you include a broker configuration in a Global Data Services configuration, you manage the broker configuration as a single unit. Only an entire broker configuration can be added to or deleted from a Global Data Services pool, and a broker configuration cannot span multiple pools.

If you attempt to add or remove a database that belongs to a broker configuration to or from a Global Data Services pool, an error occurs. You can only add a database to the Global Data Services pool by adding it to the broker configuration using the Data Guard `dgmgrl` utility. When you add a database to the broker configuration, you must run the `gdsctl sync brokerconfig` command to synchronize Global Data Services and Data Guard.

Conversely, when you remove a database from a broker configuration, it is removed from the Global Data Services pool to which this broker configuration belongs. This is the only way to remove a database from a pool that contains a broker configuration.

You can configure global services with a role attribute to be active in a specific database role, such as primary or physical standby database. If you enable fast-start failover, the Oracle Data Guard broker automatically fails over to a standby database if the primary database fails.

The global service managers configured to work with Oracle Data Guard broker ensure that the appropriate database services are active and that the appropriate Fast Application Notification (FAN) events are published after a role change. The Global Data Services framework supports the following Oracle Data Guard broker configurations:

- The set of databases in a Global Data Services pool can be either the set of databases that belong to a single broker configuration or a set of databases that do not belong to a broker configuration. You can add a broker configuration only to an empty Global Data Services pool and, if a pool already contains a broker configuration, then, to add a database to the pool, you must add the database to the broker configuration contained in the pool.
- Role-based global services are supported only for database pools that contain a broker configuration.

## Database Placement of a Global Service

- You can specify which databases will support a service.
  - These databases are referred to as *preferred databases*.
- GSM ensures that a global service runs on all preferred databases for which it has been specified.
- The number of preferred databases is referred to as the *database cardinality* of a global service.
- When a global service is added, a list of databases is specified for the service.
- If one of the preferred databases fails to provide a global service, the GSM relocates the service to an available database to maintain the specified database cardinality.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can specify which databases will support a service. These databases are referred to as preferred databases. The GSM ensures that a global service runs on all preferred databases for which it has been specified. The number of preferred databases is referred to as the database cardinality of a global service. You must specify at least one preferred database for a global service.

When you add or modify a global service, you can specify a list of available databases for this global service. If one of the preferred databases fails to provide a global service, the global service manager relocates the service to an available database to maintain the specified database cardinality for that service.

In a Global Data Services pool that contains an Oracle Data Guard broker configuration, a role-based global service can be started on a database only if the database is listed as preferred or available for the service and the role attribute of the database corresponds to the role attribute specified for the service. For example, a global service that can run on any database in a broker configuration (as long as the role attribute of the database is set to primary) must have primary specified for its role attribute and have all other databases in the broker configuration with role attributes set to preferred.

**Note:** If you set `preferred_all` for which databases will support a service, you do not have to explicitly specify preferred or available databases. The `preferred_all` setting implies that all databases in the pool are preferred.

Do not confuse database cardinality of global services with their instance cardinality. Instance cardinality is specified and maintained separately for each Oracle RAC database and is not maintained across databases of a Global Data Services pool.

## Replication Lag and Global Services

- For performance reasons, distributed environments often use asynchronous replication of data between databases.
- This increases the probability of a delay between the time an update is made on a primary and when it appears on a replica.
  - This is known as *replication lag*.
- GDS enables apps to differentiate global services providing real-time data from those returning out-of-date data.
- For applications that can tolerate a certain degree of lag, you can configure a maximum acceptable lag value.
- A client request can only be forwarded to a replica not lagging behind the primary database longer than the configured lag time for the service.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For performance reasons, distributed database systems often use asynchronous replication of data between databases, which means that there is the possibility of a delay between the time an update is made to data on a primary database and the time this update appears on a replica. When this happens, the replica database lags behind its primary database.

Global Data Services enables applications to differentiate between global services that provide real-time data from services that can return out-of-date data because of replication lag. For applications that can tolerate a certain degree of lag, you can configure a maximum acceptable lag value. For applications that cannot tolerate any replication lag, you can set the lag time for global services to zero. Requests for this global service are forwarded only to a primary database, or to a replica database that is synchronized with the primary database. For applications that cannot tolerate any replication lag, you can set the lag time for global services to zero. Requests for this global service are forwarded only to a primary database, or to a replica database that is synchronized with the primary database.

For many applications, it is acceptable to read out-of-date data as long as it is consistent. Such applications can use global services running on any database replica irrespective of the length of the replication lag time. If you configure the lag time to a value other than zero, then a client request can be forwarded only to a replica database that is not lagging behind the primary database by longer than the configured lag time for the service. Specification of the maximum replication lag is only supported for Active Data Guard configurations.

# Global Connection Load Balancing

- A client connecting to a RAC database using a service can take advantage of Oracle Net connection load balancing.
- Clients connecting to a global service are load balanced as necessary, across different databases and regions.
- Global connection load-balancing functionality includes:
  - Client-side load balancing
  - Server-side load balancing
  - Region affinity for global services



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a client connects to an Oracle RAC database using a service, the client can take advantage of the Oracle Net connection load-balancing feature to spread user connections across all the instances that support that service. Similarly, in a Global Data Services configuration, clients connecting to a global service are load balanced, as necessary, across different databases and regions. Discussion of global connection load balancing includes:

- Client-side load balancing
- Server-side load balancing
- Region affinity for global services

## Client-Side Load Balancing

- Client-side load balancing and failover in a Global Data Services configuration is similar to that for a RAC database.
- In a GDS configuration, a client in a GDS region first tries to connect to any of the GSMS in its local region.
- If a GSM from the local region does not respond, the client tries a GSM in another region.
- To enable client-side load balancing and failover across multiple regions, clients must:
  - Use a connect descriptor containing a list of addresses of local and buddy GSMS for load balancing and intra-region failover



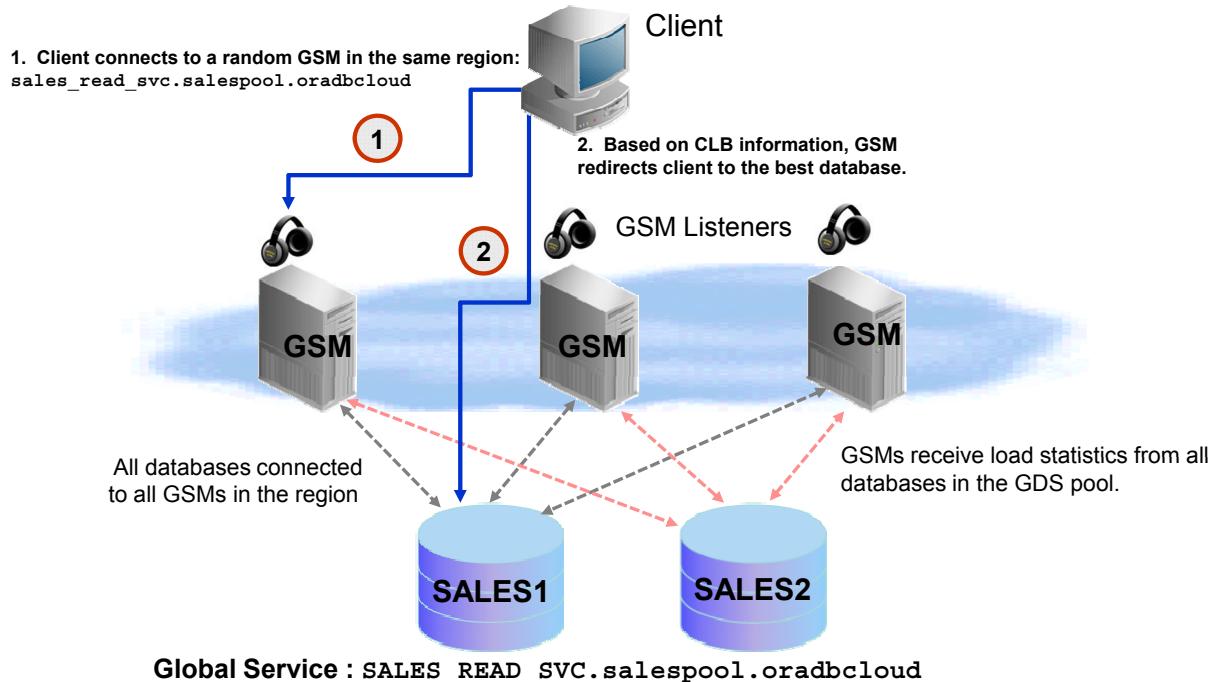
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Client-side load balancing balances connection requests across listeners and includes connection failover. With connection failover, if an error is returned from the chosen listener address, Oracle Net Services tries the next address in the address list until either a successful connection is made or it has exhausted all the addresses in the list.

Client-side load balancing and failover in a Global Data Services configuration is similar to that for an Oracle RAC database. In a Global Data Services configuration, however, a client in a Global Data Services region first tries to connect to any of the global service managers in its local region. If a global service manager from the local region does not respond, the client tries a global service manager in another region.

To enable client-side load balancing and failover across multiple regions in a GDS configuration, clients must use an Oracle Net connect descriptor that contains a list of addresses of local and buddy GSMS for load balancing and intra-region failover. If a region is not specified, it defaults to the region name of the global service manager to which the client is connected. You can also configure timeout and retry attempts for each list to enable multiple connection attempts to the current global service manager before moving to another global service manager in the list.

# Server-Side Load Balancing



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Server-side connection load balancing for an Oracle RAC database has the listener directing connection requests to the best Oracle RAC database instance. Some applications have long-lived connections, whereas other applications have short-lived connections.

For global services, server-side connection load balancing works similarly, except that, instead of being limited to a single database, workloads are balanced across multiple databases in the Global Data Services configuration. In most cases, a global service manager directs a client request for a global service to a database server in the same region, unless all local servers are overloaded and a remote server can provide significantly better quality of service.

In some cases, to take advantage of data caching on a local server, you might want to direct requests to the local region. Global Data Services enables you to specify a desired level of client/server affinity for a global service.

## Region Affinity for Global Services

- *Region affinity* is the ability to configure global services within specific regions or in any region in the GDS configuration.
- The Global Data Services framework supports three types of region affinity:
  - Any-region affinity
  - Affinity to a local region
  - Affinity to a local region with interregion failover



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can configure global services to operate within specific regions or in any region in the Global Data Services configuration. This is called region affinity. The Global Data Services framework supports three types of region affinity:

- Any-region affinity
- Affinity to a local region
- Affinity to a local region with interregion failover

## Any-Region Affinity

- Any-region affinity routes a connection request to the best database in the GDS configuration, regardless of region.
- Any-region affinity is the default value for region affinity.
- The database is chosen based on its performance and network latency between the regions where the client and database reside.
- If databases in different regions are equally loaded, this policy gives preference to a local region.
- An *interregion* connection is made only if the difference in performance between regions outweighs network latency.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Any-region affinity (the default) for a global service routes a client connection request to the best database in the Global Data Services configuration, regardless of region, that can meet the connection load-balancing goal for the requested service. The choice of the database is based on its performance and network latency between the regions where the client and database reside. If databases in different regions are equally loaded, this policy gives preference to a local region. An interregion connection is made only if the difference in database performance between regions outweighs network latency.

If you specify preferred, available databases for a global service with any-region affinity, service cardinality is maintained at the Global Data Services pool level. If a service fails on a preferred database, it will be started on any available database in the Global Data Services configuration, and the number of service providers in the pool remains the same.

When starting the service on an available database, databases in the region where the service failed have preference. If there is no available database for this service in the local region, an available database is chosen from a nearby region.

## Affinity to a Local Region

- Affinity to a local GDS region routes a client connection request to the best database in the client's region.
- The GSM chooses the database based only on performance.
- A global service with affinity to a local region can be provided in multiple GDS regions at the same time.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Affinity to a local Global Data Services region for a global service routes a client connection request to the best database in the client's region that can meet the connection load-balancing goal for the requested service. The global service manager chooses the database based only on its performance. A global service with affinity to a local region can be provided in multiple Global Data Services regions at the same time, but a client from one region never gets connected to a database in another region.

If you specify preferred or available databases for a global service with local region affinity, service cardinality is maintained at the regional level. If a service fails on a preferred database, it will start on an available database in the same region, so the number of service providers in the region remains the same. If there is no available database for this global service in the local region, no action is taken and the service cardinality decreases. If there is no database in the local region offering the global service, the client connection request is denied.

## Affinity to a Local Region with Interregion Failover

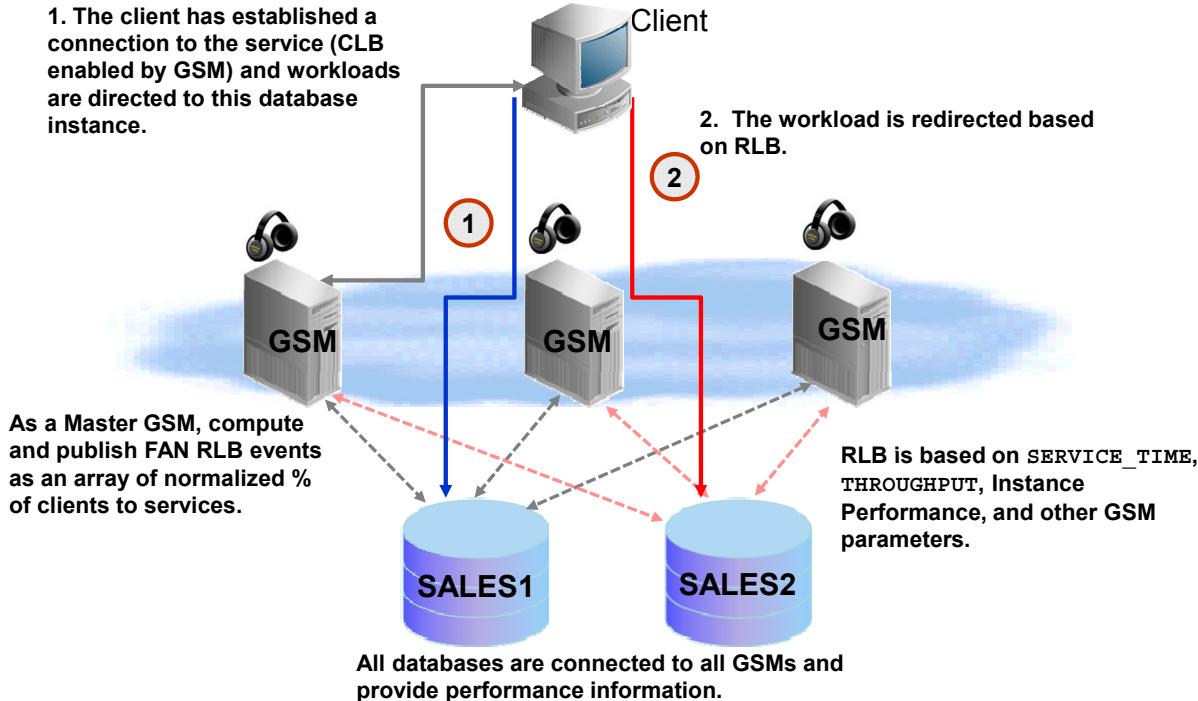
- If no databases in the local region offer a global service, the request is forwarded to the best database in another region where the requested global service is running.
- This service failover does not trigger the service to be started on an available database in another region.
  - Cardinality is maintained independently in each region and will not change as a result of service failure in another region.
- If regional databases become overloaded because of interregion failover, you can manually add a preferred database for the service.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This type of affinity is similar to that of affinity to a local Global Data Services region, except that, if there are no databases in the local region offering a global service, then, instead of denying a client request, the request is forwarded to the best database in another region where the requested global service is running. This service failover does not trigger the service to be started on an available database in another region because, with affinity to a local region, database cardinality is maintained independently in each region, and should not change as a result of service failure in another region. If regional databases become overloaded because of interregion failover, you can manually add a preferred database for the service.

# Global Runtime Connection Load Balancing



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Runtime connection load balancing distributes client work requests across persistent connections that span the instances of an Oracle RAC database, based on load-balancing information from the database. The database uses the runtime connection load-balancing goal for a service and the relative performance of database instances to generate a recommendation about where to direct service requests. There are two types of service-level goals for runtime connection load balancing:

- **SERVICE\_TIME:** Attempts to direct work requests to instances according to response time. Load-balancing data is based on elapsed time for work done in the service plus available bandwidth to the service.
- **THROUGHPUT:** Attempts to direct work requests according to throughput. The load-balancing data is based on the rate at which work is completed in the service plus available bandwidth to the service.

The Global Data Services framework also supports balancing of work requests at run time to a global service. In the Global Data Services framework, the requests are spread across connections to instances in multiple databases. Work is routed to provide the best service times globally and routing responds gracefully to changing system conditions.

To provide global runtime connection load balancing, a global service manager receives performance data for each service from all database instances in the Global Data Services configuration.

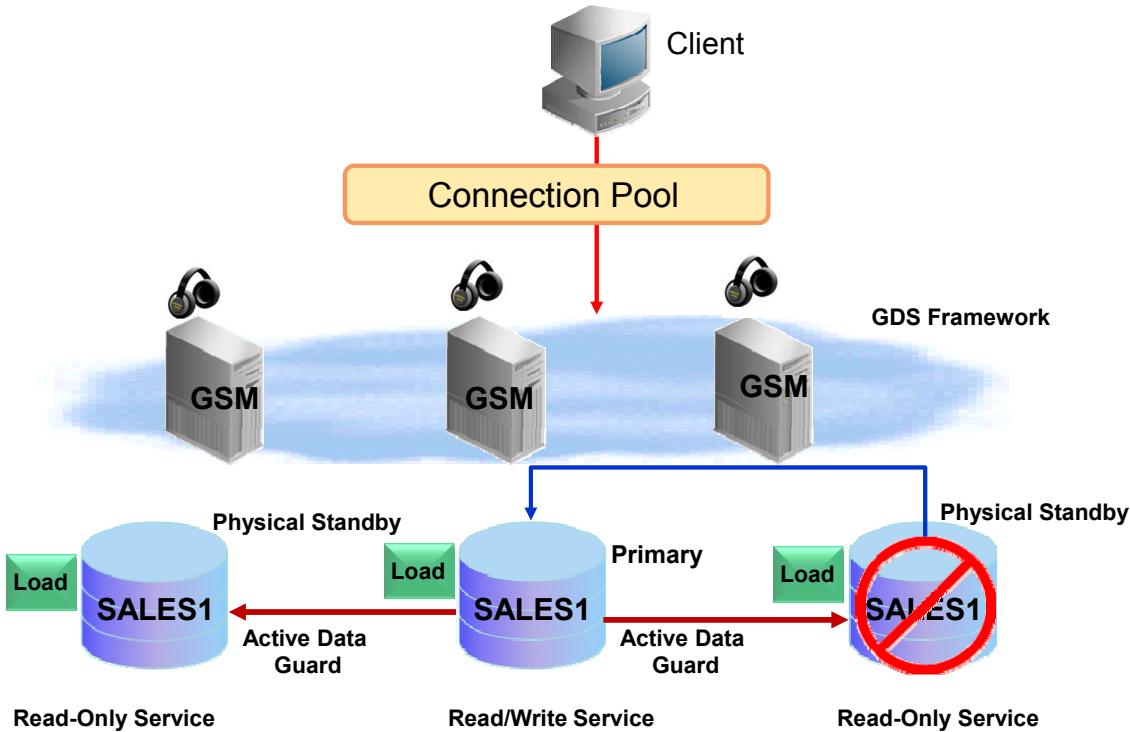
The GSM also measures interregion network latency by periodically exchanging messages with global service managers in other Global Data Services regions.

If the load-balancing goal for a global service is set to SERVICE\_TIME, a global service manager considers interregion network latency and instance performance data when deciding how to distribute work requests. For example, clients in Region A will have runtime load-balancing metrics that are weighted toward Region A, and clients in Region B will have metrics that are weighted toward Region B. This implies that, even though the service may be the same, clients in different regions receive different runtime load-balancing metrics.

If the load-balancing goal for a global service is set to THROUGHPUT, runtime load-balancing metrics are calculated based only on the performance of database instances.

In addition to runtime load-balancing metrics for local clients, a global service manager may also need to calculate runtime load-balancing metrics of remote regions and publish them for clients residing in a region where all global service managers are not available.

## Failover of Global Services



**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When a global service or database fails, a global service that was running on the database fails over to another database where the global service is enabled but not yet running (assuming that the database role matches the service role). The global service manager considers preferred databases as the failover target before available databases.

If you stop a global service using `gdsctl`, the service does not fail over to another database. However, the database where the service was stopped remains a failover target for this service. If the service fails on another database, it can start on that database. When a global service fails over to an available database, the Global Data Services framework does not move the service back to the preferred database when the preferred database restarts because of the following:

- The service has the desired cardinality.
- Keeping the service on the current instance or database provides a higher level of service availability.
- Not moving the service back to the initial preferred instance or database prevents a second outage.

If necessary, you can manually relocate the global service back to the preferred database after it has restarted, gracefully and without terminating active sessions.

## Role-Based Services

- In a GDS pool with a Data Guard broker configuration, the GDS framework supports role-based global services.
- Valid roles are:
  - PRIMARY
  - PHYSICAL\_STANDBY
  - LOGICAL\_STANDBY
  - SNAPSHOT\_STANDBY
- A global service is started *only* when the database role matches the role specified for the service.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In a Global Data Services pool that contains an Oracle Data Guard broker configuration, the Global Data Services framework supports role-based global services. Valid roles are PRIMARY, PHYSICAL\_STANDBY, LOGICAL\_STANDBY, and SNAPSHOT\_STANDBY. The Global Data Services framework automatically starts a global service only when the database role matches the role specified for the service.

If a database switches roles or fails, the Oracle Data Guard broker notifies the Global Data Services framework about the role change, and the global service manager ensures that services start according to the new database roles. A global service cannot fail over from a database in one Global Data Services region to a database in another region if the locality parameter is set to LOCAL\_ONLY, and interregion failover is not enabled.

When a global service fails over, fast connection failover, if enabled on Oracle clients, provides rapid failover of the connections to that global service. The Global Data Services framework, similar to Oracle RAC, uses Fast Application Notification (FAN) to notify applications about service outages. Instead of waiting for the application to poll the database and detect a problem, clients receive FAN events and react immediately. Sessions to the failed instance or node will be terminated, and new connections will be directed to available instances providing the global service.

All global service managers monitor service availability on all databases in a Global Data Services configuration. When a global service cannot be provided anymore because of a failure, the GSM that detects that a global service is unavailable and connects to the Global Data Services catalog database tries to start the service on an available database.

**Note:** A GSM cannot automatically fail over a service if it is unable to connect to the Global Data Services catalog.

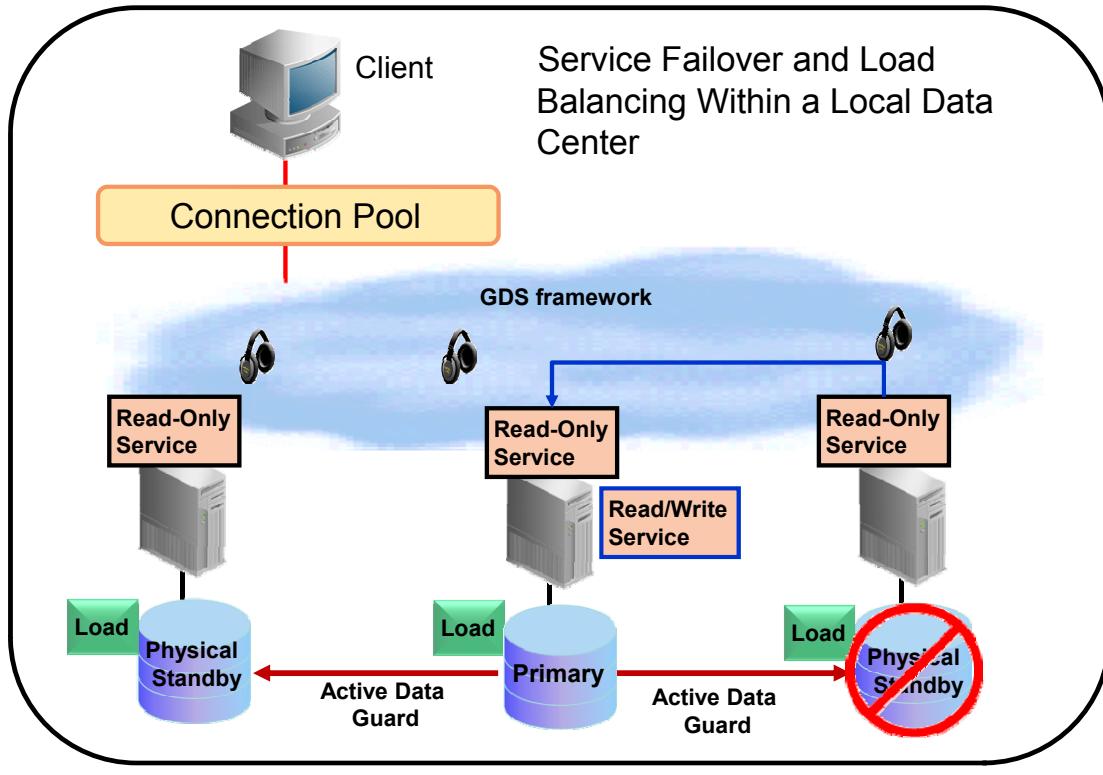
## GDS Use Cases

- With Active Data Guard
  - Simple failover within a local data center
  - Service failover and load balancing within a local data center
  - Service failover and load balancing across data centers
  - Automatic role-based services upon Data Guard role transitions
  - Load balancing for reader farms
- With Oracle GoldenGate
  - Load balancing across data centers (Multi-Master)
  - Service failover and load balancing across data centers (Master-Replica)



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

## GDS with Active Data Guard



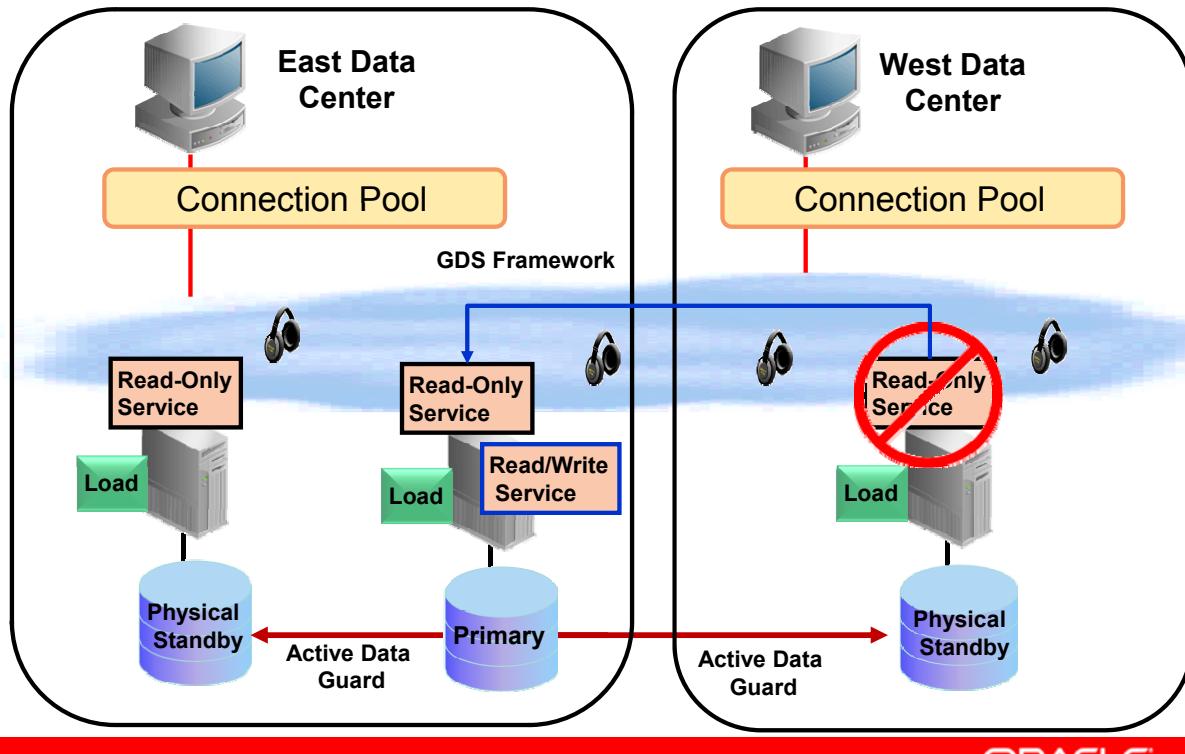
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this GDS configuration, the read/write service runs on the primary database, whereas the read-only service includes the physical standby databases located in the same region. If a physical standby fails, GDS will fail over the read-only service to an available database in the region. This configuration integrates Active Data Guard–replicated databases into a scalable and highly available private data cloud.

# GDS with Active Data Guard

Service Failover and Load Balancing Across Data Centers



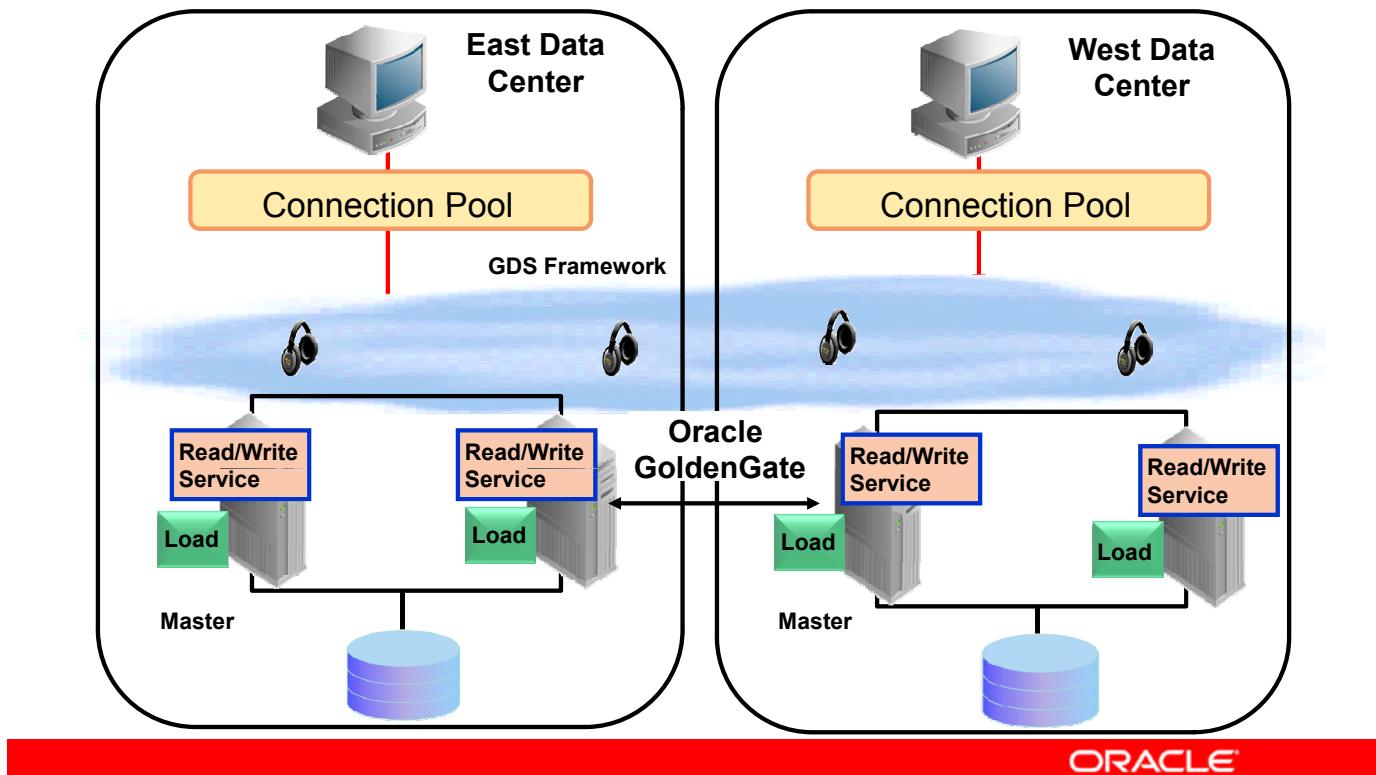
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The scenario illustrated in the slide shows a multiregion GDS configuration incorporating Active Data Guard-replicated databases. In this example, the read/write service runs on the primary database, whereas the read-only service is load balanced across all standby databases as defined by the service. If a physical standby fails, GDS will fail over the read-only service to the primary database. This approach allows you to integrate Active Data Guard-replicated databases, both local and remote, into a scalable and highly available private data cloud.

## GDS with GoldenGate (Multi-Master)

Service Failover and Load Balancing Across Data Centers



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

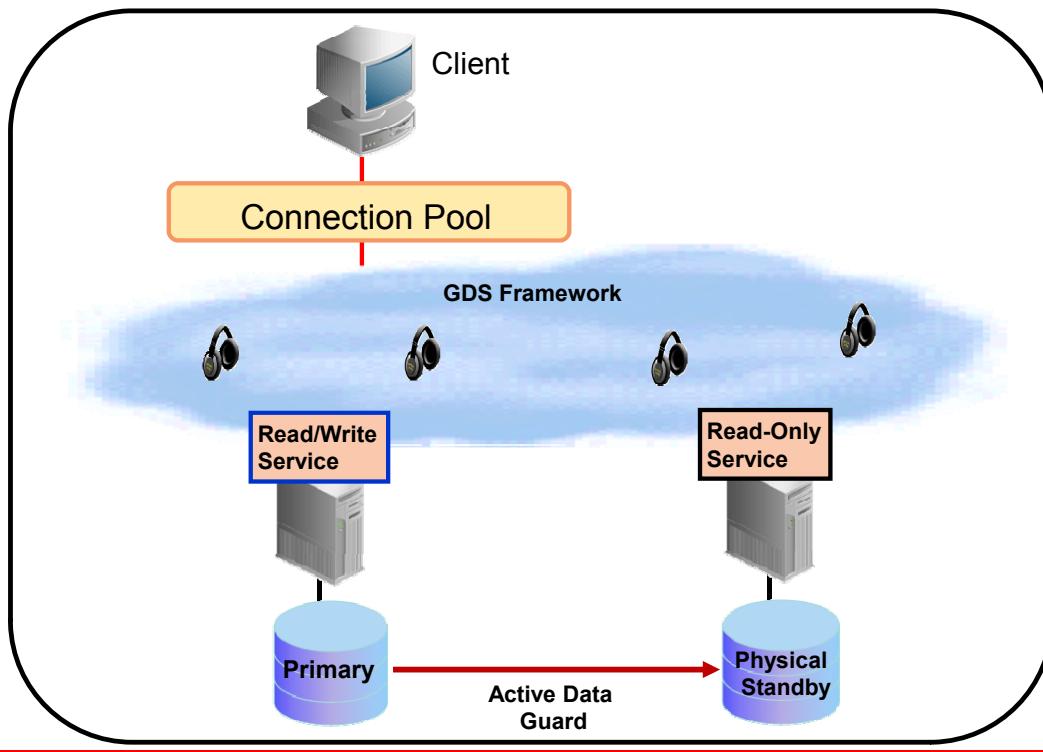
The configuration shown in the slide is an active-active replication scheme incorporating Oracle GoldenGate. Data would most likely be partitioned by schema, table, and row to avoid data collisions, while the partitioning would be managed by the application. Oracle RAC is responsible for scaling application workloads within a cluster, whereas GDS can scale application workloads in the GDS configuration.

GDS defers to the Oracle RAC server pool configuration for which nodes to participate in the server pool. GDS will start the service on the master. Which nodes it will run on depends on the Oracle RAC server pool configuration.

**Note:** The upcoming release of Oracle GoldenGate will support Oracle Database 12c, allowing GDS to fully support GoldenGate.

## GDS with Active Data Guard

Data Guard Role Transition



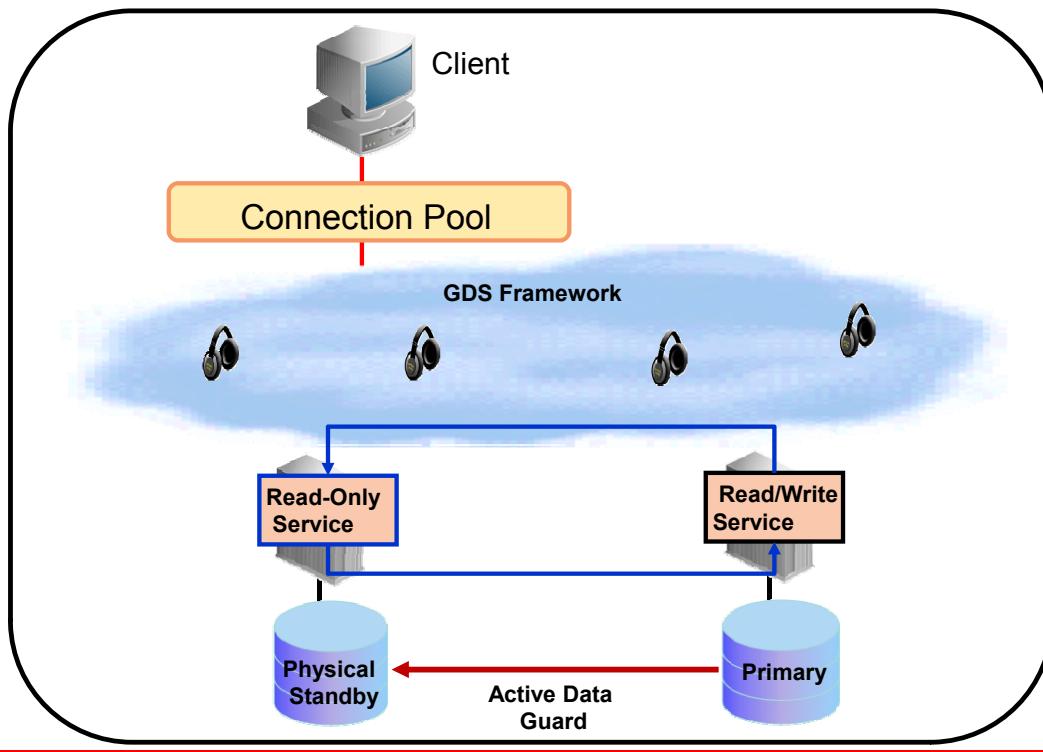
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The example in the slide illustrates a simple GDS Active Data Guard configuration supporting Data Guard role transition. This configuration has a read/write service running on the primary database, whereas a read-only service runs on the standby database. Data Guard Broker is required for role transition. The following slide illustrates the results of the role transition.

## GDS with Active Data Guard

Data Guard Role Transition



ORACLE

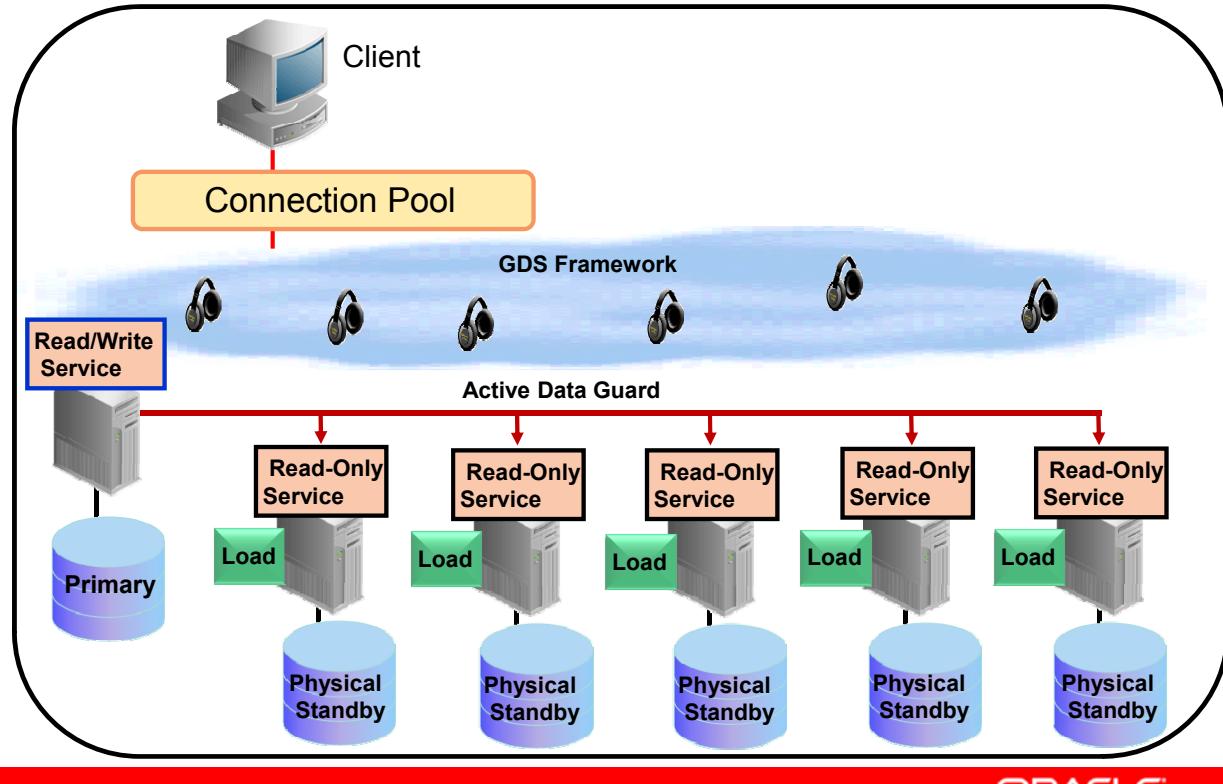
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Upon initiating the Data Guard role change via Data Guard Broker, GDS will fail over the read/write service to the new primary database and fail over the read-only service to the new standby database as illustrated in the slide.

GDS performs automatic role-based service management for Data Guard configurations. Oracle Clusterware is not needed in this configuration.

# GDS Employing Reader Farms

## Load Balancing in an Active Data Guard Reader Farm



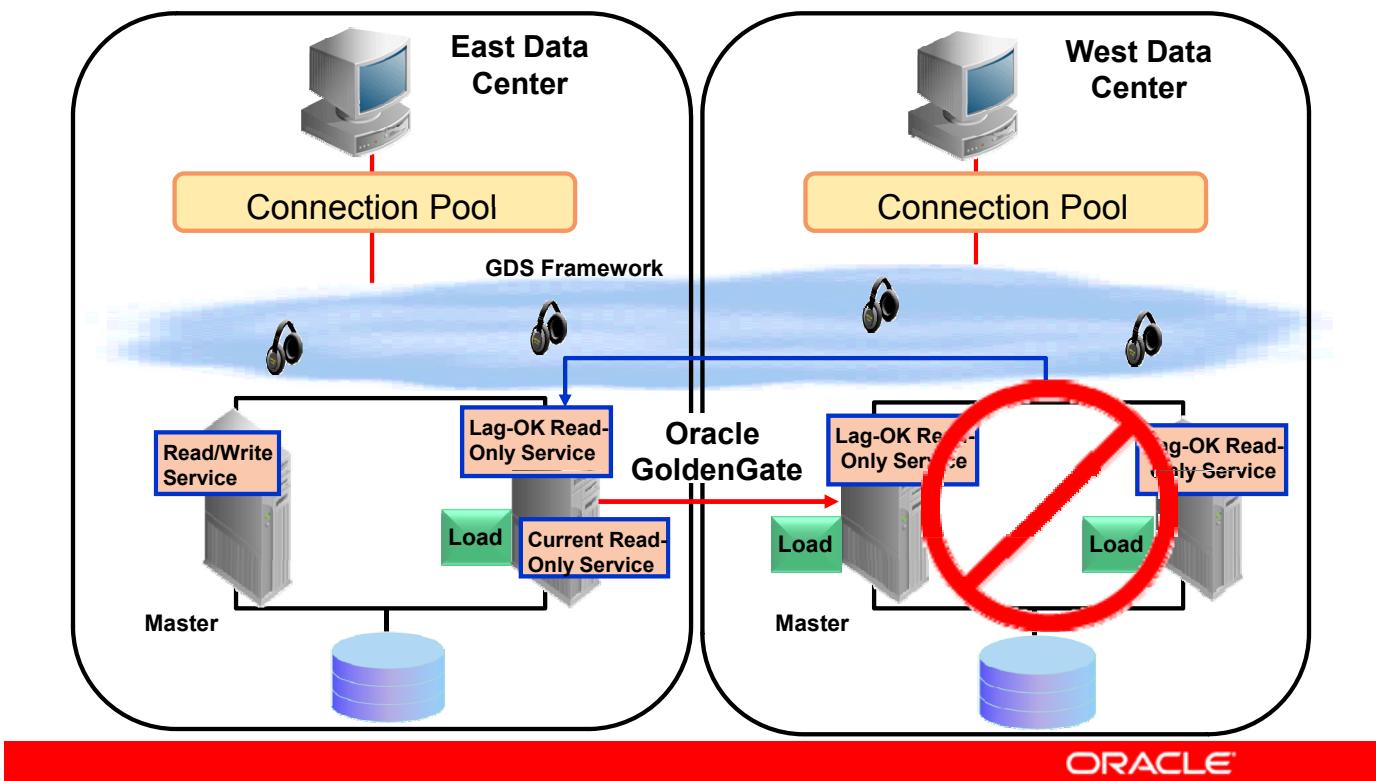
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The scenario in the slide illustrates how GDS can be used to manage load balancing in an Active Data Guard reader farm configuration. Note that the read/write service runs on the primary database, whereas the read-only service is load balanced across the standby databases in the reader farm. This configuration effectively optimizes the utilization of Active Data Guard databases in the GDS configuration.

# GDS with GoldenGate (Master-Replica)

## Service Failover and Load Balancing Across Data Centers



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The GDS configuration in the slide uses Oracle GoldenGate to manage replication in the GDS regions. This scenario illustrates service failover and load balancing across GDS regions.

In this example, the read/write service runs on a dedicated master node and the current read-only service is load balanced across other Master instances. The lag-OK read-only service is load balanced across all instances of the replica.

Upon a database failure of the replica, the lag-OK read-only service is failed over to the master, as illustrated in the slide. This approach allows you to integrate Oracle GoldenGate replicated databases, both local and remote, into a scalable and highly available private database cloud.

## Oracle Database 12c Global Data Services: Summary

- Extends RAC-style service management, load balancing, and failover for distributed environments of replicated databases
- Provides:
  - **Higher Availability:** Can fail over a service across databases located anywhere
  - **Effective Scalability:** Can balance workload over a set of replicated databases
  - **Better Manageability:** Supports centralized administration of global resources
- Is useful for planned and unplanned outage reduction for various workloads



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To sum up, Global Data Services for database clouds applies the Oracle RAC service model to sets of globally distributed, heterogeneous databases, providing dynamic load balancing, failover, and centralized service management for a set of replicated databases that offer common services. The set of databases can include Oracle RAC and noncluster Oracle databases interconnected through Oracle Data Guard, Oracle GoldenGate, or any other replication technology.

GDS enables you to integrate your locally and globally distributed, loosely coupled databases running on heterogeneous platforms into an easily managed, scalable, and highly available private database cloud that can be shared by clients around the globe.

## Quiz

Which statements regarding Global Data Services are true?

- a. Global Data Services applies the RAC service model to sets of globally distributed, heterogeneous databases.
- b. Global Data Services employs a centralized, vertical framework.
- c. Global Data Services provides dynamic load balancing, failover, and centralized service management for a set of replicated databases offering common services.
- d. The set of databases can include RAC and noncluster Oracle databases interconnected through Data Guard, GoldenGate, or any other replication technology.



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: a, c, d**

## Quiz

Global Data Services supports Data Guard role transition without the need of Data Guard Broker.

- a. True
- b. False



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

**Answer: b**

## Summary

In this lesson, you should have learned how to:

- Explain the benefits provided by Global Data Services for managing cloud-deployed distributed databases
- List the components of the Global Services Framework
- Explain how Global Service connections are load balanced
- Describe the process of Global Services failover
- Describe supported database or replication in common usage scenarios



Copyright © 2013, 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