Contents

Preface .................................................................................................................................................................. ix
  Audience ......................................................................................................................................................... ix
  Documentation Accessibility ......................................................................................................................... ix
  Related Documents ........................................................................................................................................ x
  Conventions .................................................................................................................................................... x

Changes in This Release for Oracle Database High Availability Overview .................. xi
  Changes in Oracle Database 12c Release 1 (12.1) ...................................................................................... xi

1 Overview of High Availability
  1.1 What Is High Availability? .................................................................................................................. 1-1
  1.2 Importance of Availability .................................................................................................................. 1-2
  1.3 Cost of Downtime ............................................................................................................................... 1-3
  1.4 Causes of Downtime ............................................................................................................................. 1-3
  1.5 Roadmap to Implementing the Maximum Availability Architecture ............................................. 1-7

2 Determining Your High Availability Requirements
  2.1 About Determining High Availability Requirements ........................................................................ 2-1
  2.2 Analysis Framework for Determining High Availability Requirements ......................................... 2-2
    2.2.1 Business Impact Analysis ............................................................................................................. 2-3
    2.2.2 Cost of Downtime .......................................................................................................................... 2-3
    2.2.3 Recovery Time Objective .............................................................................................................. 2-4
    2.2.4 Recovery Point Objective .............................................................................................................. 2-4
    2.2.5 Manageability Goal ....................................................................................................................... 2-4
    2.2.6 Total Cost of Ownership and Return on Investment ................................................................. 2-4
  2.3 High Availability Architecture Requirements ..................................................................................... 2-5
    2.3.1 Business Performance, Budget, and Growth Plans .................................................................... 2-5

3 Features for Maximizing Availability
  3.1 Oracle Data Guard .................................................................................................................................. 3-1
    3.1.1 Oracle Active Data Guard ............................................................................................................. 3-3
    3.1.2 Data Guard and Planned Maintenance .......................................................................................... 3-5
      3.1.2.1 Data Guard Redo Apply and Standby-First Patching ......................................................... 3-6
      3.1.2.2 Data Guard Transient Logical Rolling Upgrades ............................................................... 3-7
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.2.3</td>
<td>Rolling Upgrade Using Oracle Active Data Guard</td>
<td>3-7</td>
</tr>
<tr>
<td>3.2</td>
<td>Oracle GoldenGate</td>
<td>3-8</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Oracle GoldenGate 11g Release 2</td>
<td>3-9</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Oracle GoldenGate and Maximum Availability Architecture</td>
<td>3-10</td>
</tr>
<tr>
<td>3.3</td>
<td>Best Practice: Oracle Active Data Guard and Oracle GoldenGate</td>
<td>3-11</td>
</tr>
<tr>
<td>3.3.1</td>
<td>When to Use Oracle Active Data Guard</td>
<td>3-11</td>
</tr>
<tr>
<td>3.3.2</td>
<td>When to Use Oracle GoldenGate</td>
<td>3-11</td>
</tr>
<tr>
<td>3.3.3</td>
<td>When to Use Oracle Active Data Guard and Oracle GoldenGate Together</td>
<td>3-12</td>
</tr>
<tr>
<td>3.4</td>
<td>Recovery Manager</td>
<td>3-13</td>
</tr>
<tr>
<td>3.5</td>
<td>Oracle Secure Backup</td>
<td>3-14</td>
</tr>
<tr>
<td>3.6</td>
<td>Oracle Real Application Clusters and Oracle Clusterware</td>
<td>3-15</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Benefits of Using Oracle Clusterware</td>
<td>3-16</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Benefits of Using Oracle Real Application Clusters and Oracle Clusterware</td>
<td>3-17</td>
</tr>
<tr>
<td>3.7</td>
<td>Oracle RAC One Node</td>
<td>3-18</td>
</tr>
<tr>
<td>3.8</td>
<td>Oracle Automatic Storage Management</td>
<td>3-18</td>
</tr>
<tr>
<td>3.9</td>
<td>Fast Recovery Area</td>
<td>3-20</td>
</tr>
<tr>
<td>3.10</td>
<td>Corruption Prevention, Detection, and Repair</td>
<td>3-20</td>
</tr>
<tr>
<td>3.11</td>
<td>Data Recovery Advisor</td>
<td>3-21</td>
</tr>
<tr>
<td>3.12</td>
<td>Oracle Security Features</td>
<td>3-22</td>
</tr>
<tr>
<td>3.13</td>
<td>Oracle Flashback Features</td>
<td>3-23</td>
</tr>
<tr>
<td>3.13.1</td>
<td>Oracle Flashback Query</td>
<td>3-24</td>
</tr>
<tr>
<td>3.13.2</td>
<td>Oracle Flashback Version Query</td>
<td>3-24</td>
</tr>
<tr>
<td>3.13.3</td>
<td>Oracle Flashback Transaction</td>
<td>3-25</td>
</tr>
<tr>
<td>3.13.4</td>
<td>Oracle Flashback Transaction Query</td>
<td>3-25</td>
</tr>
<tr>
<td>3.13.5</td>
<td>Oracle Flashback Table</td>
<td>3-25</td>
</tr>
<tr>
<td>3.13.6</td>
<td>Oracle Flashback Drop</td>
<td>3-25</td>
</tr>
<tr>
<td>3.13.7</td>
<td>Oracle Flashback Restore Points</td>
<td>3-26</td>
</tr>
<tr>
<td>3.13.8</td>
<td>Oracle Flashback Database</td>
<td>3-26</td>
</tr>
<tr>
<td>3.13.9</td>
<td>Block Media Recovery Using Flashback Logs or Physical Standby Database</td>
<td>3-27</td>
</tr>
<tr>
<td>3.13.10</td>
<td>Flashback Data Archive</td>
<td>3-27</td>
</tr>
<tr>
<td>3.14</td>
<td>Oracle Data Pump and Data Transport</td>
<td>3-27</td>
</tr>
<tr>
<td>3.15</td>
<td>Oracle Replication Technologies for Non-Database Files</td>
<td>3-27</td>
</tr>
<tr>
<td>3.15.1</td>
<td>Oracle Database File System</td>
<td>3-28</td>
</tr>
<tr>
<td>3.15.2</td>
<td>Oracle ASM Cluster File System</td>
<td>3-29</td>
</tr>
<tr>
<td>3.15.3</td>
<td>Oracle Solaris ZFS Storage Appliance Replication</td>
<td>3-29</td>
</tr>
<tr>
<td>3.16</td>
<td>Client and Application Failover</td>
<td>3-30</td>
</tr>
<tr>
<td>3.16.1</td>
<td>Client Failover Processing for Connections</td>
<td>3-32</td>
</tr>
<tr>
<td>3.16.1.1</td>
<td>Services</td>
<td>3-33</td>
</tr>
<tr>
<td>3.16.1.2</td>
<td>Fast Application Notification</td>
<td>3-34</td>
</tr>
<tr>
<td>3.16.2</td>
<td>Transaction Failover and Protection</td>
<td>3-36</td>
</tr>
<tr>
<td>3.16.2.1</td>
<td>Transaction Guard</td>
<td>3-36</td>
</tr>
<tr>
<td>3.16.2.2</td>
<td>Application Continuity</td>
<td>3-36</td>
</tr>
<tr>
<td>3.16.3</td>
<td>Oracle Database with Global Data Services</td>
<td>3-37</td>
</tr>
</tbody>
</table>
5 Oracle Database High Availability Solutions for Planned Downtime

5.1 High Availability Solutions for Migration

5.1.1 Platform Migration

5.1.1.1 Migrating a Database to Oracle Exadata Database Machine or SPARC SuperCluster

5.1.2 Database Migration to a Different Character Set

5.1.3 Migrating to Multitenant Architecture

5.1.4 Migration to Oracle ASM Storage

5.1.4.1 Migrate to Oracle ASM-Managed Storage Using Data Guard

5.1.4.2 Migrate to New Storage Using Oracle ASM

5.1.5 Migrating a Database from a Single-Instance System to an Oracle RAC Cluster

5.2 Dynamic and Online Resource Provisioning

5.2.1 Renaming and Relocating Online Datafiles

5.2.2 Dynamic Reconfiguration of the Database

5.2.3 Automatic Tuning of Memory Management

5.2.4 Automated Distribution of Data Files, Control Files, and Log Files

5.3 Online Reorganization and Redefinition

5.4 Oracle High Availability Solutions for System and Software Maintenance

5.4.1 Operating System Upgrades and Hardware Upgrades

5.4.2 Online Patching

5.4.3 System and Cluster Upgrades Using Data Guard

5.4.3.1 Upgrading the Physical Standby Database

5.4.3.2 Best Practices for System and Cluster Upgrades

5.4.4 Rolling Upgrade With Oracle Real Application Clusters

5.4.4.1 Rolling Patch Installation with Oracle Real Application Clusters

5.4.4.2 Rolling Patch Installation with Data Guard

5.4.5 Rolling Upgrade with Oracle Clusterware

5.4.6 Rolling Upgrade with Oracle Automatic Storage Management

5.4.7 Rolling Upgrade of Exadata Storage Server Software

5.4.8 Database Rolling Upgrade with Data Guard

5.4.8.1 Performing Database Upgrades Using Data Guard and Physical Standby Databases

5.4.8.2 Performing Database Upgrades Using Data Pump Full Transportable Export/Import

5.4.8.3 Performing Database Upgrades Using Transportable Tablespace

5.4.8.4 Performing Database Upgrades Using Oracle GoldenGate

5.4.8.5 Performing Database Upgrades Using Oracle GoldenGate and Data Guard

5.5 Online Application Maintenance and Upgrades

5.5.1 Edition-Based Redefinition

5.5.1.1 Editions

5.5.1.2 Editioning Views

5.5.1.3 Cross-edition Triggers
6 Operational Prerequisites to Maximizing Availability

6.1 Understand Availability and Performance SLAs ............................................................... 6-1
6.2 Implement a High Availability Environment and Scalable Architecture .................... 6-2
6.3 Establish Test Practices and Environment ................................................................. 6-2
6.3.1 Configuring the Test System and QA Environments ............................................. 6-2
6.3.2 Performing Preproduction Validation Steps .......................................................... 6-3
6.4 Set up and Use Security Best Practices ...................................................................... 6-5
6.5 Establish Change Control Procedures ....................................................................... 6-5
6.6 Apply Recommended Patches and Software Periodically ............................................ 6-5
6.7 Execute Data Guard Role Transitions ....................................................................... 6-6
6.8 Establish Escalation Management Procedures ......................................................... 6-6
6.9 Configure Monitoring and Service Request Infrastructure for High Availability .......... 6-7
6.9.1 Execute Database Health Checks Periodically ....................................................... 6-7
6.9.2 Configure Oracle Enterprise Manager Monitoring Infrastructure for High Availability. 6-8
6.9.3 Configure Automatic Service Request Infrastructure ............................................. 6-8
6.10 Check the Latest MAA Best Practices ...................................................................... 6-9

7 High Availability Architectures and Solutions

7.1 Oracle Database High Availability Architectures ......................................................... 7-1
7.1.1 Oracle Database with Oracle Restart ..................................................................... 7-2
7.1.2 Oracle Database with Oracle RAC One Node ...................................................... 7-3
7.1.3 Oracle Database with Oracle Real Application Clusters (Oracle RAC) .................. 7-4
7.1.3.1 Oracle Database with Oracle RAC on Extended Clusters ............................... 7-6
7.1.4 Oracle Database with Data Guard ........................................................................ 7-8
7.1.4.1 Overview of Single Standby Database Architectures ...................................... 7-11
7.1.4.2 Overview of Multiple Standby Database Architectures .................................. 7-13
7.1.5 Oracle Database with Oracle Restart and Data Guard ........................................ 7-15
7.1.6 Oracle Database with Oracle RAC One Node and Data Guard ............................. 7-15
7.1.7 Oracle Database with Oracle RAC and Data Guard ............................................ 7-15
7.1.8 Oracle Database with Oracle GoldenGate .......................................................... 7-16
7.2 Choosing the Correct High Availability Architecture ................................................ 7-17
7.3 Integrating Oracle Fusion Middleware High Availability ........................................... 7-28
7.3.1 Oracle WebLogic Server High Availability Architectures ................................... 7-28
7.3.2 Redundant Architectures ..................................................................................... 7-29
7.3.3 High Availability Services in Oracle Fusion Middleware ..................................... 7-29
7.4 Integrating High Availability for All Applications ..................................................... 7-30
8 Oracle Engineered Systems

8.1 Oracle Exadata Database Machine ................................................................. 8-1
8.2 Oracle SPARC SuperCluster ............................................................................ 8-4
8.3 Oracle Database Appliance .............................................................................. 8-5

9 Optimizing Return on Investment

9.1 Grid Computing ............................................................................................... 9-1
9.1.1 Database Server Grid .................................................................................. 9-2
9.1.2 Database Storage Grid ................................................................................ 9-2
9.2 Higher Utilization Using Active Standby Databases ........................................... 9-3
  9.2.1 Oracle Active Data Guard Option for Physical Standby Databases .............. 9-3
  9.2.2 Oracle Active Data Guard Reader Farms .................................................... 9-4
  9.2.3 Data Guard (Standby) Hub ......................................................................... 9-5
9.3 Oracle Database Consolidation ........................................................................... 9-6
  9.3.1 Multitenant Architecture ........................................................................... 9-6
  9.3.2 Oracle Virtualization .................................................................................. 9-7
9.4 Oracle Global Data Services ............................................................................... 9-8

Glossary

Index
This book introduces you to Oracle best practices for deploying a highly available database environment. It provides an overview of high availability and helps you to determine your high availability requirements. It describes the Oracle Database products and features that are designed to support high availability and describes the primary database architectures that can help your business achieve high availability.

This preface contains these topics:
- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This book is intended for chief technology officers, information technology architects, database administrators, system administrators, network administrators, and application administrators who perform the following tasks:
- Plan data centers
- Implement data center policies
- Maintain high availability systems
- Plan and build high availability solutions

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
Related Documents

Knowledge of Oracle Database, Oracle RAC, and Data Guard concepts and terminology is required to understand the configuration and implementation details described in this book. For more information, see the Oracle Database documentation set. These books may be of particular interest:

- *Oracle Database High Availability Best Practices*
  This book typically lags behind the *Oracle Database High Availability Overview* because extensive testing is required to determine the best practices. Until the release 11.2 book is available, you may find some of the methodologies in the *Oracle Database High Availability Best Practices* for release 11.1.0.7 to be useful.

- *Oracle Database Administrator’s Guide*
- *Oracle Database 2 Day + Real Application Clusters Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Automatic Storage Management Administrator’s Guide*
- *Oracle Data Guard Concepts and Administration*
- *Oracle Database Backup and Recovery User’s Guide*

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle Database. See *Oracle Database Sample Schemas* for information about using these schemas.

Also, you can download the Oracle MAA best practice white papers at [http://www.oracle.com/goto/maa](http://www.oracle.com/goto/maa).

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Changes in This Release for Oracle Database High Availability Overview

This preface lists changes in Oracle Database High Availability Overview.

Changes in Oracle Database 12c Release 1 (12.1)

The following are changes in Oracle Database High Availability Overview for Oracle Database 12c Release 1 (12.1).

New Features
The following features are new in this release:

- Global Data Services
Global Data Services applies the Oracle Real Application Clusters service model to sets of globally distributed, heterogeneous databases, providing load balancing and failover capabilities to database clouds with global services, which are services provided by multiple databases.

See "Oracle Database with Global Data Services" on page 3-37.

- Oracle Flex Clusters and Oracle Flex ASM
Oracle Clusterware and Oracle Real Application Clusters can be configured in large clusters, called an Oracle Flex Cluster. Oracle Flex ASM decouples the Oracle ASM instance from the database servers. Oracle ASM instances may be run on separate physical servers (from the database servers). Any number of Oracle ASM servers can be clustered together to support a large set of databases.


- Far sync
A Data Guard far sync instance is a remote Data Guard destination that accepts redo from the primary database and then ships that redo to other members of the Data Guard configuration. A far sync instance manages a control file, receives redo into standby redo logs (SRLs), and archives those SRLs to local archived redo logs, but that is where the similarity with standbys ends. A far sync instance does not have user data files, cannot be opened for access, cannot run redo apply, and can never function in the primary role or be converted to any type of standby database.

See "Benefits of Oracle Active Data Guard" on page 3-4.

- Consolidation and multitenant architecture
The multitenant architecture feature enables an Oracle database to contain a portable set of schemas, objects, and related structures that appears logically to an application as a separate database.

See "Oracle Database Consolidation" on page 9-6.

■ Rolling upgrade using Oracle Active Data Guard

Rolling Upgrade using Oracle Active Data Guard provides new PL/SQL packages that automate much of the process of performing a database rolling upgrade using a physical standby database.

See "Performing Database Upgrades Using Data Guard and Physical Standby Databases" on page 5-23.

■ Oracle Active Data Guard enhancements

Support for Global Temporary Tables, replication of XMLType tables and columns, and enhanced security.

See "Benefits of Oracle Active Data Guard" on page 3-4.

■ Application failover improvements

Application availability has been improved with the enhancement of Fast Application Notification, Oracle Service, and with the addition of Global Data Services, Application Continuity, and Transaction Guard.

See "Client and Application Failover" on page 3-30.
Overview of High Availability

This chapter contains the following sections:

- What Is High Availability?
- Importance of Availability
- Cost of Downtime
- Causes of Downtime
- Roadmap to Implementing the Maximum Availability Architecture

1.1 What Is High Availability?

Availability is the degree to which an application, service, or function is accessible on demand. Availability is measured by the perception of an application's user. Users experience frustration when their data is unavailable or the computing system is not performing as expected, and they do not understand or care to differentiate between the complex components of an overall solution. Performance failures due to higher than expected usage create the same disruption as the failure of critical components in the architecture. If a user cannot access the system, it is said to be unavailable.

Generally, the term downtime is used to refer to periods when a system is unavailable. Users who want their systems to be always ready to serve them need high availability. A system that is highly available is designed to provide uninterrupted computing services during essential time periods, during most hours of the day, and most days of the week throughout the year; this measurement is often shown as 24x365. Such systems may also need a high availability solution for planned maintenance operations such as upgrading a system's hardware or software.

Reliability, recoverability, timely error detection, and continuous operations are primary characteristics of a highly available solution:

- Reliability: Reliable hardware is one component of a high availability solution. Reliable software—including the database, web servers, and applications—is just as critical to implementing a highly available solution. A related characteristic is resilience. For example, low-cost commodity hardware, combined with software such as Oracle Real Application Clusters (Oracle RAC), can be used to implement a very reliable system. The resilience of an Oracle RAC database allows processing to continue even though individual servers may fail.

- Recoverability: There may be many ways to recover from a failure. Therefore, it is important to determine what types of failures may occur in your high availability environment and how to recover from those failures quickly in order to meet your business requirements. For example, if a critical table is accidentally deleted from the database, what action should you take to recover it? Does your architecture
provide the ability to recover in the time specified in a service-level agreement (SLA)?

- **Timely error detection**: If a component in your architecture fails, then fast detection is essential to recover from the unexpected failure. Although you may be able to recover quickly from an outage, if it takes an additional 90 minutes to discover the problem, then you may not meet your SLA. Monitoring the health of your environment requires reliable software to view it quickly and the ability to notify the database administrator of a problem.

- **Continuous operation**: Providing continuous access to your data is essential when very little or no downtime is acceptable to perform maintenance activities. Activities, such as moving a table to another location in the database or even adding CPUs to your hardware, should be transparent to the user in a high availability architecture.

More specifically, a high availability architecture should have the following traits:

- Tolerate failures such that processing continues with minimal or no interruption
- Be transparent to—or tolerant of—system, data, or application changes
- Provide built-in preventive measures
- Provide active monitoring and fast detection of failures
- Provide fast recoverability
- Automate detection and recovery operations
- Protect the data to minimize or prevent data loss
- Implement the operational best practices to manage your environment
- Achieve the goals set in SLAs (for example, recovery time objectives (RTOs) and recovery point objectives (RPOs)) for the lowest possible total cost of ownership

### 1.2 Importance of Availability

The importance of high availability varies among applications. Databases and the Internet have enabled worldwide collaboration and information sharing by extending the reach of database applications throughout organizations and communities. This reach emphasizes the importance of high availability in data management solutions.

Both small businesses and global enterprises have users all over the world who require access to data 24 hours a day. Without this data access, operations can stop, and revenue is lost. Users now demand service-level agreements from their information technology (IT) departments and solution providers, reflecting the increasing dependence on these solutions. Increasingly, availability is measured in dollars, euros, and yen, not just in time and convenience.

Enterprises have used their IT infrastructure to provide a competitive advantage, increase productivity, and empower users to make faster and more informed decisions. However, with these benefits has come an increasing dependence on that infrastructure. If a critical application becomes unavailable, then the business can be in jeopardy. The business might lose revenue, incur penalties, and receive bad publicity that has a lasting effect on customers and on the company’s stock price.

It is important to examine the factors that determine how your data is protected and maximize availability to your users.
1.3 Cost of Downtime

The need to deliver increasing levels of availability continues to accelerate as enterprises reengineer their solutions to gain competitive advantage. Most often, these new solutions rely on immediate access to critical business data. When data is not available, the operation can cease to function. Downtime can lead to lost productivity, lost revenue, damaged customer relationships, bad publicity, and lawsuits.

It is not always easy to place a direct cost on downtime. Angry customers, idle employees, and bad publicity are all costly, but not directly measured in currency. On the other hand, lost revenue and legal penalties incurred because SLA objectives are not met can easily be quantified. The cost of downtime can quickly grow in industries that are dependent on their solutions to provide service.

Other factors to consider in the cost of downtime are:

- The maximum tolerable length of a single unplanned outage
  
  If the event lasts less than 30 seconds, then it may cause very little impact and may be barely perceptible to users. As the length of the outage grows, the effect may grow exponentially and negatively affect the business.

- The maximum frequency of allowable incidents
  
  Frequent outages, even if short in duration, may similarly disrupt business operations.

When designing a solution, it is important to recognize the true cost of downtime to understand how the business can benefit from availability improvements.

Oracle provides a range of high availability solutions to fit every organization regardless of size. Small workgroups and global enterprises alike are able to extend the reach of their critical business applications. With Oracle and the Internet, applications and data are reliably accessible everywhere, at any time.

1.4 Causes of Downtime

One of the challenges in designing a high availability solution is examining and addressing all of the possible causes of downtime. It is important to consider causes of both unplanned and planned downtime when designing a fault-tolerant and resilient IT infrastructure. Planned downtime can be just as disruptive to operations as unplanned downtime, especially in global enterprises that support users in multiple time zones.

Table 1–1 describes unplanned outage types and provides examples of each type.
## Table 1–1 Causes of Unplanned Downtime

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site failure</td>
<td>A site failure may affect all processing at a data center, or a subset of applications supported by a data center.</td>
<td>• Extended sitewide power failure&lt;br&gt;• Sitewide network failure&lt;br&gt;• Natural disaster makes a data center inoperable&lt;br&gt;• Terrorist or malicious attack on operations or the site</td>
</tr>
<tr>
<td>Clusterwide failure</td>
<td>The whole cluster hosting an Oracle RAC database is unavailable or fails. This includes:&lt;br&gt;  ■ Failures of nodes in the cluster&lt;br&gt;  ■ Failure of any other components that result in the cluster being unavailable and the Oracle database and instances on the site being unavailable</td>
<td>• The last surviving node on the Oracle RAC cluster fails and the node or database cannot be restarted&lt;br&gt;• Both redundant cluster interconnections fail or clusterware failure&lt;br&gt;• Database corruption so severe that continuity is not possible on the current database server&lt;br&gt;• Disk storage failure</td>
</tr>
<tr>
<td>Computer failure</td>
<td>A computer failure outage occurs when the system running the database becomes unavailable because it has failed or is no longer available. When the database uses Oracle RAC then a computer failure represents a subset of the system (while retaining full access to the data).</td>
<td>• Database system hardware failure&lt;br&gt;• Operating system failure&lt;br&gt;• Oracle instance failure</td>
</tr>
<tr>
<td>Network failure</td>
<td>A network failure outage occurs when a network device stops or reduces network traffic and communication from your application to database, database to storage, or any system to system that is critical to your application service processing.</td>
<td>• Network switch failure&lt;br&gt;• Network interface failure&lt;br&gt;• Network cable failures</td>
</tr>
<tr>
<td>Storage failure</td>
<td>A storage failure outage occurs when the storage holding some or all of the database contents becomes unavailable because it has shut down or is no longer available.</td>
<td>• Disk drive failure&lt;br&gt;• Disk controller failure&lt;br&gt;• Storage array failure</td>
</tr>
</tbody>
</table>
Causes of Downtime

Overview of High Availability

Data corruption

A corrupt block is a block that was changed so that it differs from what Oracle Database expects to find. Block corruptions can be categorized as physical or logical:

- In a **physical corruption**, which is also called a media corruption, the database does not recognize the block at all: the checksum is invalid, the block contains all zeros, or the header and footer of the block do not match.

- In a **logical corruption**, the contents of the block are logically inconsistent. Examples of logical corruption include corruption of a row piece or index entry.

Block corruptions can also be divided into interblock corruption and intrablock corruption:

- In **intrablock corruption**, the corruption occurs in the block itself and can be either a physical or a logical corruption.

- In an **interblock corruption**, the corruption occurs between blocks and can only be a logical corruption.

A data corruption outage occurs when a hardware, software, or network component causes corrupt data to be read or written. The service-level impact of a data corruption outage may vary, from a small portion of the database (down to a single database block) to a large portion of the database (making it essentially unusable).

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data corruption</td>
<td>A corrupt block is a block that was changed so that it differs from what</td>
<td>Operating system or storage device driver</td>
</tr>
<tr>
<td></td>
<td>Oracle Database expects to find. Block corruptions can be categorized as</td>
<td>failure</td>
</tr>
<tr>
<td></td>
<td>physical or logical:</td>
<td>Faulty host bus adapter</td>
</tr>
<tr>
<td></td>
<td>- In a <strong>physical corruption</strong>, which is also called a media corruption,</td>
<td>Disk controller failure</td>
</tr>
<tr>
<td></td>
<td>the database does not recognize the block at all: the checksum is invalid,</td>
<td>Volume manager error</td>
</tr>
<tr>
<td></td>
<td>the block contains all zeros, or the header and footer of the block do not</td>
<td>causing a bad disk read or write</td>
</tr>
<tr>
<td></td>
<td>match.</td>
<td>Software or hardware defects</td>
</tr>
<tr>
<td></td>
<td>- In a <strong>logical corruption</strong>, the contents of the block are logically</td>
<td></td>
</tr>
<tr>
<td></td>
<td>inconsistent. Examples of logical corruption include corruption of a row</td>
<td></td>
</tr>
<tr>
<td></td>
<td>piece or index entry.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Block corruptions can also be divided into interblock corruption and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>intrablock corruption:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- In <strong>intrablock corruption</strong>, the corruption occurs in the block itself</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and can be either a physical or a logical corruption.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- In an <strong>interblock corruption</strong>, the corruption occurs between blocks and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>can only be a logical corruption.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A data corruption outage occurs when a hardware, software, or network</td>
<td></td>
</tr>
<tr>
<td></td>
<td>component causes corrupt data to be read or written. The service-level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>impact of a data corruption outage may vary, from a small portion of the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>database (down to a single database block) to a large portion of the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>database (making it essentially unusable).</td>
<td></td>
</tr>
</tbody>
</table>
### Table 1–1 (Cont.) Causes of Unplanned Downtime

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| Human error         | A human error outage occurs when unintentional or other actions are committed that cause data in the database to become incorrect or unusable. The service-level impact of a human error outage can vary significantly, depending on the amount and critical nature of the affected data. | ■ File deletion (at the file system level)  
■ Dropped database object  
■ Inadvertent data changes  
■ Malicious data changes |
| Lost writes         | A lost write is another form of data corruption, but it is much more difficult to detect and repair quickly. A data block stray or lost write occurs when:  
■ For a lost write, an I/O subsystem acknowledges the completion of the block write even though the write I/O did not occur in the persistent storage. On a subsequent block read on the primary database, the I/O subsystem returns the stale version of the data block, which might be used to update other blocks of the database, thereby corrupting it.  
■ For a stray write, the write I/O completed but it was written somewhere else, and a subsequent read operation returns the stale value.  
■ For an Oracle RAC system, a read I/O from one cluster node returns stale data after a write I/O is completed from another node (lost write). For example, this occurs if a network file system (NFS) is mounted in Oracle RAC without disabling attribute caching (for example, without using the `noac` option). In this case, the write I/O from one node is not immediately visible to another node because it is cached. | ■ Operating system or storage device driver failure  
■ Faulty host bus adapter  
■ Disk controller failure  
■ Volume manager error  
■ Other application software  
■ Lack of network file systems (NFS) write visibility across a cluster |
| Delay or slowdown   | A delay or slowdown occurs when the database or the application cannot process transactions because of a resource or lock contention. A perceived delay can be caused by lack of system resources. | ■ Database or application deadlocks  
■ Runaway processes that consume system resources  
■ Logon storms or system faults  
■ Combination of application peaks with lack of system or database resources  
■ Archived redo log destination or fast recovery area destination becomes full |

Table 1–2 describes planned outage types and provides examples of each type.
Oracle offers high availability solutions to help avoid both unplanned and planned downtime, and recover from failures. Chapter 4 and Chapter 5 discuss each of these high availability solutions in detail.

1.5 Roadmap to Implementing the Maximum Availability Architecture

Oracle high availability solutions and sound operational practices are key to the successful implementation of an IT infrastructure. However, technology alone is not enough.

Choosing and implementing an architecture that best fits your availability requirements can be a daunting task. Maximum Availability Architecture (MAA) simplifies the process of choosing and implementing a high availability architecture to fit your business requirements. The MAA:

- Encompasses redundancy across all components
- Provides protection and tolerance from computer failures, storage failures, human errors, data corruption, lost writes, system delays or slowdowns, and site disasters
- Recovers from outages as quickly and transparently as possible
- Provides solutions to eliminate or reduce planned downtime
- Provides consistent high performance and robust security
- Is straightforward to deploy, can be managed efficiently, and scales easily
Roadmap to Implementing the Maximum Availability Architecture

- Achieves SLAs at the lowest possible total cost of ownership

To build, implement, and maintain this type of architecture, you need to:

1. Analyze your specific high availability requirements, including both the technical and operational aspects of your IT systems and business processes, as described in Chapter 2, "Determining Your High Availability Requirements."

2. Familiarize yourself with Oracle high availability features, as described in Chapter 3, "Features for Maximizing Availability."

3. Understand the key effects of the Oracle high availability features on businesses and applications, as described in Chapter 4, "Oracle Database High Availability Solutions for Unplanned Downtime" and Chapter 5, "Oracle Database High Availability Solutions for Planned Downtime."

4. Use operational best practices to provide a successful MAA implementation, as described in Chapter 6, "Operational Prerequisites to Maximizing Availability."

5. Choose a high availability architecture, as described in Chapter 7, "High Availability Architectures and Solutions."

6. Learn how Oracle's Engineered Systems such Oracle Exadata Database Machine, Sparc Supercluster, or Oracle Database Appliance can contribute to your overall MAA, as described in Chapter 8, "Oracle Engineered Systems."

7. Optimize your return on investment (ROI) of any of the high availability architectures and solutions, with Oracle Grid Computing, Oracle Active Data Guard real time reporting and utilization, Oracle Database multitenant architecture using pluggable databases, or Oracle Virtualization and Oracle Data Cloud, as described in Chapter 9, "Optimizing Return on Investment."

8. Implement a high availability architecture using the following resources:

   - MAA and high availability best practices white papers and other information
     Oracle offers various best practices white papers, customer MAA papers with proof of concepts, customer case studies, recorded web casts, demonstrations, and presentations. These resources provide technical details about the MAA various high availability technologies, along with best practice recommendations for configuring and using such technologies.

     These MAA resources are available at http://www.oracle.com/goto/maa

   - Oracle Database High Availability Best Practices
     This book provides detailed best practice recommendations and information. It can help you to configure a new high availability environment, or migrate an existing configuration to create a redundant, reliable system without sacrificing simplicity and performance.
Determining Your High Availability Requirements

This chapter contains the following sections:

- About Determining High Availability Requirements
- Analysis Framework for Determining High Availability Requirements
- High Availability Architecture Requirements

2.1 About Determining High Availability Requirements

Any enterprise that is designing and implementing a high availability strategy must begin by performing a thorough analysis of the business drivers that require high availability. Implementing high availability may involve critical tasks such as:

- Retiring legacy systems
- Investing in more capable and robust systems and facilities
- Redesigning the overall IT architecture and operations to adapt to this high availability model
- Redesigning business processes
- Hiring and training personnel

This chapter provides a framework to effectively evaluate the high availability requirements of a business. By combining your business analysis with an understanding of the level of investment required to implement different high availability solutions, you can develop a high availability architecture that achieves both business and technical objectives.

You can use the high availability analysis framework to:

- Complete a business impact analysis
- Identify and categorize the critical business processes that have the high availability requirements
- Formulate the cost of downtime
- Establish utilization, recovery time objective (RTO), and recovery point objective (RPO) goals for these various business processes
- Understand goals for manageability, total cost of ownership (TCO), and return on investment (ROI)
This framework enables you to define service-level agreements (SLAs) in terms of high availability for critical aspects of your business. For example, this framework can categorize your business processes into several high availability tiers:

- Tier 1 processes have the maximum business impact. They have the most stringent high availability requirements, with RTO and RPO close to zero, and requiring continuously available supporting systems. For a business with a high-volume e-commerce presence, this may be the web-based customer interaction system.

- Tier 2 processes that have slightly relaxed high availability and RTO and RPO requirements. The second tier of an e-commerce business may be its supply chain and merchandising systems. For example, these systems do not need to maintain extremely high degrees of availability and may have nonzero RTO and RPO values. Thus, the high availability systems and technologies chosen to support the tier 2 processes are likely to be different from those of the tier 1 processes.

- Tier 3 processes may be related to internal development and quality assurance processes. Systems supporting these processes do not need to have the rigorous high availability requirements of the other tiers.

As shown in Figure 2–1, the next step for the business is to evaluate the capabilities of the various high availability systems and technologies, and to choose the ones that meet its SLA requirements, within the guidelines dictated by business performance issues, budgetary constraints, and anticipated business growth.

**Figure 2–1  Planning and Implementing a Highly Available Enterprise**

2.2 Analysis Framework for Determining High Availability Requirements

The elements of this analysis framework are:

- Business Impact Analysis
- Cost of Downtime
- Recovery Time Objective
2.2.1 Business Impact Analysis

A rigorous business impact analysis:

- Identifies the critical business processes in an organization
- Calculates the quantifiable loss risk for unplanned and planned IT outages affecting each of these business processes
- Outlines the effects of these outages
- Considers essential business functions, people and system resources, government regulations, and internal and external business dependencies
- Is based on objective and subjective data gathered from interviews with knowledgeable and experienced personnel
- Reviews business practice histories, financial reports, IT systems logs, and so on

The business impact analysis categorizes the business processes based on the severity of the impact of IT-related outages. For example, consider a semiconductor manufacturer with chip fabrication plants located worldwide. Semiconductor manufacturing is an intensely competitive business requiring a huge financial investment that is amortized over high production volumes. The human resource applications used by plant administration are unlikely to be considered as mission-critical as the applications that control the manufacturing process in the plant. Failure of the applications that support manufacturing affects production levels and have a direct impact on the financial results of the company.

Similarly, an internal knowledge management system is likely to be considered mission-critical for a management consulting firm, because the business of a client-focused company is based on internal research accessibility for its consultants and knowledge workers. The cost of downtime of such a system is extremely high for this business.

2.2.2 Cost of Downtime

A complete business impact analysis provides the insight needed to quantify the cost of unplanned and planned downtime. Understanding this cost is essential because it helps prioritize your high availability investment and directly influences the high availability technologies that you choose to minimize the downtime risk.

Various reports have been published, documenting the costs of downtime in different industries. Examples include costs that range from millions of dollars for each hour of brokerage operations and credit card sales, to tens of thousands of dollars for each hour of package shipping services.

These numbers are staggering. The Internet can connect the business directly to millions of customers. Application downtime can disrupt this connection, cutting off a business from its customers. In addition to lost revenue, downtime can negatively affect customer relationships, competitive advantages, legal obligations, industry reputation, and shareholder confidence.
2.2.3 Recovery Time Objective

The business impact analysis determines your **recovery time objective (RTO)**. RTO is defined as the maximum amount of time that an IT-based business process can be down before the organization starts suffering unacceptable consequences (financial losses, customer dissatisfaction, reputation, and so on). RTO indicates the downtime tolerance of a business process or an organization in general.

The RTO requirements are driven by the mission-critical nature of the business. Thus, for a system running a stock exchange, the RTO is zero or near to zero. An organization is likely to have varying RTO requirements across its various business processes. Thus, for a high volume e-commerce website, for which there is an expectation of rapid response times and for which customer switching costs are very low, the web-based customer interaction system that drives e-commerce sales is likely to have an RTO close to zero. However, the RTO of the systems that support back-end operations, such as shipping and billing, can be higher. If these back-end systems are down, then the business may resort to manual operations temporarily without a significant visible impact.

The ability to take orders through the e-commerce website immediately (the RTO) may be more important than the RPO, because lost data can be reloaded later.

2.2.4 Recovery Point Objective

The business impact analysis also determines your **recovery point objective (RPO)**. RPO is the maximum amount of data that an IT-based business process can lose without harm to the organization. RPO indicates the data-loss tolerance of a business process or an organization in general. This data loss is often measured in terms of time, for example, 5 hours or 2 days of data loss.

A stock exchange where millions of dollars worth of transactions occur every minute cannot afford to lose any data. Thus, its RPO must be zero. The web-based sales system in the e-commerce example does not require an RPO of zero, although a low RPO is essential for customer satisfaction. However, its back-end merchandising and inventory update system can have a higher RPO because lost data can be reentered.

2.2.5 Manageability Goal

A **manageability goal** is more subjective than either the RPO or the RTO. It results from an objective evaluation of the skill sets and management resources available in an organization and the degree to which the organization can successfully manage all elements of a high availability architecture. Just as RPO and RTO measure an organization’s tolerance for downtime or data loss, your manageability goal measures the organization’s tolerance for complexity in the IT environment. When less complexity is a requirement, simpler methods of achieving high availability are preferred over methods that may be more complex to manage, even if the latter could attain more aggressive RTO and RPO objectives. Understanding manageability goals helps organizations differentiate between what is possible and what is practical to implement.

2.2.6 Total Cost of Ownership and Return on Investment

Understanding **total cost of ownership (TCO)** and **return on investment (ROI)** is essential to selecting a high availability architecture that also achieves the business goals of your organization. TCO includes all costs (such as acquisition, implementation, systems, networks, facilities, staff, training, and support), over the
useful life of the solution chosen. Likewise, the ROI calculation captures all of the financial benefits that accrue to a given high availability architecture.

For example, consider a high availability architecture in which IT systems and storage at a remote standby site remain idle with no other business use that can be served by the standby systems. The only return on investment for the standby site is the cost of downtime avoided by its use in a failover scenario. Contrast this with a different high availability architecture that enables IT systems and storage at the standby site to be used productively while in the standby role (for example, for reports or for off-loading the primary system of the overhead of user queries). The return on investment of such an architecture includes both the cost of downtime avoided and the financial benefits that accrue to its productive use while in the standby database role.

2.3 High Availability Architecture Requirements

The following sections provide further information about high availability system capabilities and business performance, budget and growth plans. See Section 7.2, "Choosing the Correct High Availability Architecture."

2.3.1 Business Performance, Budget, and Growth Plans

High availability solutions must address business requirements. For example, a business may use a zero-data-loss solution that synchronously mirrors every transaction on the primary database to a remote database. However, considering the speed-of-light limitations and the physical limitations associated with a network, there are round-trip delays in the network transmission. These delays increase with distance and vary based on network bandwidth, traffic congestion, router latencies, and so on. Thus, this synchronous mirroring, if performed over large wide area network (WAN) distances, affects the primary site performance.

Online buyers may notice these system latencies and be frustrated with long system response times; consequently, they may go somewhere else for their purchases. This is an example where the business must make a trade-off between having a zero-data-loss solution and maximizing system performance. Conversely, if the business drivers justify the investment to avoid making this trade-off, a multisite architecture can be implemented that places a synchronous zero-data-loss standby site in close proximity to the primary site and a second asynchronous standby site located up to thousands of miles away.

High availability solutions must also be based on financial considerations and future growth estimates. It is tempting to build redundancies throughout the IT infrastructure and claim that the infrastructure is completely failure-proof. Although you cannot always equate higher availability with higher cost, imprudent decisions can lead to budget overruns or an unmanageable and unscalable combination of solutions that is complex and expensive to integrate and maintain.

A high availability solution that has impressive performance benchmark results may be enticing. However, investing in such a solution without careful analysis of how the technology capabilities match the business drivers is unwise. The business could end up with a solution that:

- Does not integrate well with the rest of the system infrastructure
- Has annual integration and maintenance costs that easily exceed the implementation costs
Prudent and judicious decision-makers must invest only in solutions that are well-integrated; standards-based; straightforward to implement, maintain, and manage; and have a scalable architecture for accommodating future business growth.
This chapter describes the Oracle Database features used in MAA solutions.

- Oracle Data Guard
- Oracle GoldenGate
- Best Practice: Oracle Active Data Guard and Oracle GoldenGate
- Recovery Manager
- Oracle Secure Backup
- Oracle Real Application Clusters and Oracle Clusterware
- Oracle RAC One Node
- Oracle Automatic Storage Management
- Fast Recovery Area
- Corruption Prevention, Detection, and Repair
- Data Recovery Advisor
- Oracle Security Features
- Oracle Flashback Technology
- Oracle Data Pump and Data Transport
- Oracle Replication Technologies for Non-Database Files
- Client and Application Failover

See Also:
- The overview of high availability in Oracle Database Concepts
- The list of new high availability features in Oracle Database New Features Guide

3.1 Oracle Data Guard

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable Oracle databases to survive outages of any kind, including natural disasters and data corruptions. A Data Guard standby database is an exact replica of the production database and thus can be transparently utilized in combination with traditional backup, restoration, flashback, and cluster techniques to provide the highest possible
level of data protection and data availability. Data Guard is included in Oracle Enterprise Edition.

A Data Guard configuration consists of one primary database and one or more standby databases. A primary database can be either a single-instance Oracle database or an Oracle RAC database. Similar to a primary database, a standby database can be either a single-instance Oracle database or an Oracle RAC database. Using a backup copy of the primary database, you can create up to 30 standby databases that receive redo directly from the primary database. Optionally you can use a cascaded standby to create Data Guard configurations where the primary transmits redo to a single remote destination, and that destination forwards redo to multiple standby databases. This enables a primary database to efficiently synchronize many more than 30 standby databases if desired.

---

**Note:** Oracle Active Data Guard is an extension of basic Data Guard providing advanced features that off-load various types of processing from a production database, extend zero data loss protection over any distance, and that enhance high availability. Oracle Active Data Guard is licensed separately from Oracle Database Enterprise Edition. Oracle Active Data Guard is discussed more completely in Section 3.1.1, “Oracle Active Data Guard.”

---

There are several types of standby databases. Data Guard physical standby database is the MAA best practice for data protection and disaster recovery and is the most common type of standby database used. A physical standby database uses Redo Apply (an extension of Oracle media recovery) to maintain an exact, physical replica of the production database. When configured using MAA best practices, Redo Apply uses multiple Oracle-aware validation checks to prevent corruptions that can impact a primary database from impacting the standby. Other types of Data Guard standby databases include: snapshot standby (a standby open read/write for test or other purposes) and logical standby (used to reduce planned downtime).

**Benefits of Using Data Guard**

- Continuous Oracle-aware validation of all changes using multiple checks for physical and logical consistency of structures within an Oracle data block and redo, before updates are applied to a standby database. This isolates the standby database and prevents it from being impacted by data corruptions that can occur on the primary system.
- Transparent operation: There are no restrictions on the use of Data Guard physical standby for data protection. Redo Apply supports all data and storage types, all DDL operations, and all applications (custom and packaged applications), and guarantees data consistency across primary and standby databases.
- Highest performance: Fast redo transport for best recovery point objective, fast apply performance for best recovery time objective.
- Fast failover to a standby database to maintain availability should the primary database fail for any reason. Failover is either a manual or automatic operation depending on how Data Guard is configured.
- Integrated client notification framework to enable application clients to connect to a new primary database after a failover occurs.
- Automatic or automated (depending upon configuration) resynchronization of a failed primary database, quickly converting it to a synchronized standby database after a failover occurs.
Choice of flexible data protection levels to support all network configurations, availability and performance SLAs, and business requirements.

Management of a primary and all of its standby databases as a single configuration to simplify management and monitoring using either the Data Guard Broker command-line interface or Oracle Enterprise Manager Cloud Control.

Data Guard Broker 12c greatly improves manageability with additional features for comprehensive configuration health checks, resumable switchover operations, streamlined role transitions, support for cascaded standby configurations, and user-configurable thresholds for transport and apply lag to automatically monitor the ability of the configuration to support SLAs for recovery point and recovery time objectives at any instant in time.

Efficient transport to multiple remote destinations using a single redo stream originating from the primary production database and forwarded by a cascading standby database.

Snapshot Standby enables a physical standby database to be open read/write for testing or any activity that requires a read/write replica of production data. A snapshot standby continues to receive but does not apply updates generated by the primary. When testing is complete, a snapshot standby is converted back into a synchronized physical standby database by first discarding the changes made during the open read/write, and then applying the redo received from the primary database. Primary data is always protected. Snapshot standby is particularly useful when used in conjunction with Oracle Real Application Testing (workload is captured at the production database for replay and subsequent performance analysis at the standby database— an exact replica of production).

Reduction of planned downtime by utilizing a standby database to perform maintenance in rolling fashion. The only downtime is the time required to perform a Data Guard switchover; applications remain available while the maintenance is being performed. (See Section 3.3.3, "When to Use Oracle Active Data Guard and Oracle GoldenGate Together" and Table 5-7, "Oracle High Availability Solutions for System and Software Maintenance" for more details).

Increased flexibility for Data Guard configurations where the primary and standby systems may have different CPU architectures or operating systems subject to limitations defined in My Oracle Support note 413484.1 at http://support.oracle.com/.

Efficient disaster recovery for an Oracle Database 12c container database (CDB). Data Guard failover and switchover completes using a single command at a CDB level regardless of how many databases (pluggable databases or PDBs) are consolidated within the CDB.

Data Guard 12c enables a specific administration privilege, SYSDG, to handle standard administration duties for Data Guard. This new privilege is based on the least privilege principle, in which a user is granted only the necessary privileges required to perform a specific function and no more. The SYSDBA privilege continues to work as in previous releases.

### 3.1.1 Oracle Active Data Guard

Oracle Active Data Guard is Oracle’s strategic solution for real time data protection and disaster recovery for the Oracle database using a physical replication process. Oracle Active Data Guard also provides high return on investment in disaster recovery systems by enabling a standby database to be open read-only while it applies
changes received from the primary database. Oracle Active Data Guard is a separately licensed product that provides advanced features that greatly expand Data Guard capabilities included with Oracle Enterprise Edition.

**Figure 3–1  Oracle Active Data Guard Architecture**

Oracle Active Data Guard enables administrators to improve performance by offloading processing from the primary database to a physical standby database that is open read-only while it applies updates received from the primary database. Offload capabilities of Oracle Active Data Guard 12c were enhanced to include read-only reporting and ad-hoc queries (including DML to global temporary tables and unique global or session sequences), data extracts, fast incremental backups, redo transport compression, efficient servicing of multiple remote destinations, and the ability to extend zero data loss protection to a remote standby database without impacting primary database performance. Oracle Active Data Guard also increases high availability by performing automatic block repair and enabling High Availability Upgrades (new automation in Oracle Database 12c for more easily implementing database rolling upgrades).

---

**Note:** Oracle Active Data Guard is licensed separately as a database option license for Oracle Database Enterprise Edition. All Oracle Active Data Guard capabilities are also included in an Oracle GoldenGate license for Oracle Enterprise Edition. This provides customers with the choice of a standalone license for Oracle Active Data Guard, or licensing Oracle GoldenGate to acquire access to all advanced Oracle replication capabilities.

---

**Benefits of Oracle Active Data Guard**

Oracle Active Data Guard inherits all of the benefits previously listed for Data Guard, plus the following:

- Improves primary database performance: Production-offload to an Oracle Active Data Guard standby database of read-only applications, reporting, and ad hoc queries. Any application compatible with a read-only database can run on an Oracle Active Data Guard standby. Oracle also provides integration that enables the offloading of many Oracle E-Business Suite Reports, PeopleTools reporting, Oracle Business Intelligence Enterprise Edition (OBIEE), and Oracle TopLink applications to an Oracle Active Data Guard standby database.

- Oracle Active Data Guard 12c provides new support for DML to global temporary tables and the use of sequences at the standby database. This significantly expands the number of read-only applications that can be off-loaded from production databases to an Oracle Active Data Guard standby database.
- The unique ability to easily scale read performance using multiple Oracle Active Data Guard standby databases, also referred to as a Reader Farm.
- Production-offload of data extracts using Oracle Data Pump or other methods that read directly from the source database.
- Production-offload of the performance impact from network latency in a synchronous, zero data loss configuration where primary and standby databases are separated by hundreds or thousands of miles. Oracle Active Data Guard 12c far sync utilizes a lightweight instance (control file and archive log files, but no recovery and no data files), deployed on a system independent of the primary database. The far sync instance is ideally located at the maximum distance from the primary system that an application can tolerate the performance impact of synchronous transport to provide optimal protection. Data Guard transmits redo synchronously to the far sync instance and far sync forwards the redo asynchronously to a remote standby database that is the ultimate failover target. If the primary database fails, the same failover command used for any Data Guard configuration, or mouse click using Oracle Enterprise Manager 12c, or automatic failover using Data Guard Fast-Start Failover executes a zero data loss failover to the remote destination. This transparently extends zero data loss protection to a remote standby database just as if it were receiving redo directly from the primary database, while avoiding the performance impact to the primary database of WAN network latency in a synchronous configuration.
- Production-offload of the overhead of servicing multiple remote standby destinations using far sync. In a far sync configuration, the primary database ships a single stream of redo to a far sync instance using synchronous or asynchronous transport. The far sync instance is able to forward redo asynchronously to as many as 29 remote destinations with zero incremental overhead on the source database.
- Production-offload of CPU cycles required to perform redo transport compression. Redo transport compression can be performed by the far sync instance if the Data Guard configuration is licensed for Oracle Advanced Compression. This conserves bandwidth with zero incremental overhead on the primary database.
- Production-offload and increased backup performance by moving fast incremental backups off of the primary database and to the standby database by utilizing Oracle Active Data Guard support for RMAN block change tracking.
- Increased high availability using Oracle Active Data Guard automatic block repair to repair block corruptions detected at either the primary or standby, transparent to applications and users.
- Increased high availability by reducing planned downtime for upgrading to new Oracle Database patch sets and database releases using the additional automation provided by high availability Upgrade, new with Oracle Active Data Guard 12c (described in Section 3.1.2.3, "Rolling Upgrade Using Oracle Active Data Guard.")

### 3.1.2 Data Guard and Planned Maintenance

Data Guard standby databases can be used to reduce planned downtime by performing maintenance in a rolling fashion. Changes are implemented first at the standby database. The configuration is allowed to run with the primary at the old version and standby at the new version until there is confidence that the new version is ready for production. A Data Guard switchover is then performed, transitioning production to the new version. The only database downtime is the time required to perform the switchover.
There are several approaches to performing maintenance in a rolling fashion using a Data Guard standby. Customer requirements and preferences determine which approach is used. The following approaches are discussed in this document:

- Data Guard Redo Apply and Standby-First Patching
- Data Guard Transient Logical Rolling Upgrades
- Rolling Upgrade Using Oracle Active Data Guard

### 3.1.2.1 Data Guard Redo Apply and Standby-First Patching

Beginning with Oracle Database 10g, there has been increased flexibility in cross-platform support using Data Guard Redo Apply. In certain Data Guard configurations, primary and standby databases are able to run on systems having different operating systems (for example, Windows and Linux), word size (32bit/64bit), or hardware architectures. Redo Apply can also be used to migrate to Oracle Automatic Storage Management (ASM), to move from single instance Oracle databases to Oracle RAC, to perform technology refresh, or to move from one data center to the next.

Beginning with Oracle Database 11g release 2 (11.2), Standby-First Patch Apply (physical standby using Redo Apply) can support different software patch levels between a primary database and its physical standby database for the purpose of applying and validating Oracle patches in a rolling fashion. Patches eligible for Standby-First patching include:

- Database Patch Set Update (PSU)
- Database Critical Patch Update (CPU)
- Database bundled patch
- Oracle Exadata Database Machine bundled patch
- Exadata Storage Server Software patch
- Any operating system, system firmware, or system changes compatible with the existing Oracle database version

Standby-First Patch Apply is supported for certified software patches for Oracle Database Enterprise Edition 11g release 2 (11.2) and later.

In each of the types of planned maintenance previously described, the configuration begins with a primary and physical standby database (in the case of migration to a new platform, or to ASM or Oracle RAC, the standby is created on the new platform). After all changes are implemented at the physical standby database, Redo Apply (physical replication) is used to synchronize the standby with the primary. A Data Guard switchover is used to transfer production to the standby (the new environment).

**See Also:**

- My Oracle Support Note 413484.1 at [http://support.oracle.com/](http://support.oracle.com/) for information about mixed platform combinations supported in a Data Guard configuration.
- My Oracle Support Note 1265700.1 at [http://support.oracle.com/](http://support.oracle.com/) for more information about Standby First Patch Apply and the README for each patch to determine if a target patch is certified as being a Standby-First Patch.
3.1.2.2 Data Guard Transient Logical Rolling Upgrades

There are numerous types of maintenance tasks that are unable to use Redo Apply (physical replication) to synchronize the original version of a database with the changed or upgraded version. These tasks include:

- Database patches or upgrades that are not Standby-First Patch Apply-eligible. This includes database patch-sets (11.2.0.2 to 11.2.0.3) and upgrade to new Oracle Database releases (11.2.0.3 to 12.1).
- Maintenance must be performed that modifies the physical structure of a database that would require downtime (for example, adding partitioning to non-partitioned tables, changing Basicfile LOBs to Securefile LOBs, changing XML-CLOB to Binary XML, or altering a table to be OLTP-compressed).

All of the previous types of maintenance can be performed in a rolling fashion using a Data Guard standby database by using Data Guard SQL Apply (logical replication) to synchronize the old and new versions of the database. Prior to Oracle Database 11g, this required creating a logical standby database, performing the maintenance on the logical standby, resynchronizing the standby with the primary, and then switching over. Additionally, if a physical standby was being used for disaster recovery, then a new physical standby database would have to be created from a backup of the production database at the new version. This represented a number of logistical and cost challenges when upgrading a multi-terabyte database.

Beginning with Oracle Database 11g, database rolling upgrades can use a new procedure called Transient Logical that begins and ends with a physical standby database. SQL Apply is only used during the phase when Data Guard is synchronizing across old and new versions. A new logical standby database does not need to be created if there is already a physical standby in place. A new physical standby database does not need to be created from a backup of the production database at the new version after the maintenance is complete. Similar to the traditional process of upgrading a Data Guard configuration having an in-place physical standby, the original primary is upgraded or changed using redo from the new primary database and Redo Apply (a single catalog upgrade migrates both primary and standby databases to the new Oracle release).

Transient Logical upgrades require that the primary database be at Oracle Database 11g release 1 (11.1) or later and that the database meet the pre-requisites of SQL Apply.

Oracle provides a Bourne shell script that automates a number of the manual steps required by the Transient Logical rolling upgrade process. See the MAA Best Practice Paper "Database Rolling Upgrades Made Easy" at [http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-upgrades-made-easy-131972.pdf](http://www.oracle.com/technetwork/database/features/availability/maa-wp-11g-upgrades-made-easy-131972.pdf) for more information.

See Also: MAA Best Practice Paper "Database Rolling Upgrades Made Easy" at [http://www.oracle.com/goto/maa](http://www.oracle.com/goto/maa)

3.1.2.3 Rolling Upgrade Using Oracle Active Data Guard

Oracle Database 12c introduces rolling upgrade using Oracle Active Data Guard to provide a simpler, automated, and easily repeatable method for reducing planned downtime than represented by the manual Transient Logical rolling upgrade procedure previously described. Rolling upgrade using Oracle Active Data Guard transforms the 42 or more steps required by the manual procedure into several easy-to-use DBMS_ROLLING PL/SQL packages.

A rolling upgrade using Oracle Active Data Guard uses the following steps:

- Call DBMS_ROLLING.INIT_PLAN
Generates an upgrade plan with a configuration-specific set of instructions to guide the administrator through the upgrade process

- Call DBMS_ROLLING.SET_PARAMETER
  - Modifies parameters of the rolling upgrade

- Install new software at all databases participating in the upgrade

- Call DBMS_ROLLING.START_PLAN
  - Configures primary and standby databases participating in the upgrade

- Upgrade or make changes to the standby database

- Call DBMS_ROLLING.SWITCHOVER
  - Switchover moves the production to the new version
  - Switchover is the only downtime required

- Restart former primary using new binaries if appropriate

- Call DBMS_ROLLING.FINISH_PLAN
  - Completes the upgrade of the old primary and any additional standby databases in the Data Guard configuration and resynchronizes with the new primary

Rolling upgrade using Oracle Active Data Guard has the following benefits:

- Provides a simple specify-compile-execute protocol
  - Catches configuration errors at the compilation step
  - Runtime errors are detected during execution

- The state is kept in the database
  - Enables a reliable, repeatable process

- Runtime steps are constant regardless of how many databases are involved

- Handles failure at the original primary database

- Enables data protection for the upgraded primary at all times

Rolling upgrade using Oracle Active Data Guard requires an Oracle Active Data Guard license and that the primary database be at Oracle Database 12c release 1 (12.1) or later and that the database satisfy prerequisites of SQL Apply. If the primary database is on an earlier Oracle Database release, see the MAA white paper “Database Rolling Upgrades Made Easy” at http://www.oracle.com/goto/maa.

See Also: Oracle Data Guard Concepts and Administration “Using DBMS_ROLLING to Perform a Rolling Upgrade.”

### 3.2 Oracle GoldenGate

Oracle GoldenGate is Oracle’s strategic logical replication solution for data distribution and data integration. Unlike replication solutions from other vendors, Oracle GoldenGate is more closely integrated with Oracle Database while also providing an open, modular architecture ideal for replication across heterogeneous database management systems. This combination of attributes eliminates compromise, making Oracle GoldenGate the preferred replication solution for addressing requirements that span Oracle Database and non-Oracle Database environments.
3.2.1 Oracle GoldenGate 11g Release 2

Oracle GoldenGate offers a real-time, log-based change data capture and replication software platform. The software provides capture, routing, transformation, and delivery of transactional data across heterogeneous databases in real time.

A typical environment includes a capture, pump, and delivery process. Each of these processes can run on most of the popular operating systems and databases, including Oracle Database and non-Oracle databases. All or a portion of the data may be replicated, and the data within any of these processes can be manipulated for not only heterogeneous environments but also different database schemas. Oracle GoldenGate supports multimaster replication, hub-and-spoke deployment, and data transformation.

Oracle GoldenGate 11g Release 2 offers significant new features that greatly enhance its replication capabilities and integration with Oracle Database. The new features include:

- **GoldenGate Integrated Capture for Oracle Database.** The Integrated Capture mechanism relies on Oracle Database internal log parsing and processing to capture DML transactions. By moving closer to the Oracle database engine, Oracle GoldenGate can take advantage of new Oracle Database features and functionality more quickly. Integrated capture also supports two different deployment models: Local capture which runs on the source database and Downstream capture which runs on a separate server. With Integrated Capture, Oracle GoldenGate now supports all compression used by Oracle Database and Oracle Exadata, including support for Hybrid Columnar Compression (HCC), OLTP, and Segment compression. Integrated Capture also adds distributed transaction support for XA (distributed) and PDML (parallel DML) transactions on Oracle RAC. Finally, XML Object Relational and XML Binary data types are supported along with LOB full and partial reads (selective update) from the redo log.

- **Intelligent conflict detection and resolution for active-active or multimaster implementations.** Setting up conflict detection and resolution takes less time to implement using prebuilt functions to identify the conditions under which an error occurs and how to handle the record when the condition occurs.

- **Expanded globalization capability for international customers including multibyte support and character set conversions.**

- **Improved security with added support for the industry encryption standard Federal Information Protection Standard (FIPS).**

- **Improved performance with reduction of acknowledgement messages during data replication.**

- **Enhanced manageability and monitoring for customers wanting to take advantage of Oracle Enterprise Manager.**

- **Expanded heterogeneity with new real-time capture support for IBM i-Series (AS/400) and delivery support for Postgres.**

**See Also:**

3.2.2 Oracle GoldenGate and Maximum Availability Architecture

Oracle GoldenGate logical replication enables all databases in an Oracle GoldenGate configuration, both source and target databases, to be open read/write. This fact combined with Oracle GoldenGate advanced replication features make it a key component of MAA for addressing a broad range of high availability challenges for zero downtime maintenance, cross platform migration, and continuous data availability, specifically:

- Zero or near zero downtime maintenance. In this regard Oracle GoldenGate provides greater flexibility than basic capabilities provided by Data Guard. Oracle GoldenGate source and target databases can have a different physical and logical structure, can reside on different hardware and operating system architectures, can span wide differences in Oracle Database releases (for example, 9i to 12c), or be a mix of Oracle and non-Oracle systems. Maintenance is first performed on a target database while production runs on the source. After the maintenance is complete, production can be moved to the source all at once, similar to a Data Guard switchover. Optionally, bidirectional replication can be used to gradually move users over to the new system to create the perception of zero downtime. In either case, Oracle GoldenGate replication can be enabled in the reverse direction to keep the original source database synchronized during a transition period, making it simple to effect a planned fall-back to the previous version if needed with minimal downtime and no data loss.

- Zero or near-zero downtime application upgrades. Application upgrades that modify back-end database objects typically result in significant planned downtime while maintenance is being performed. Oracle GoldenGate replication enables data transformations that map database objects used by a previous version of an application to objects modified by the new version of an application. This enables database maintenance to be performed on a separate copy of the production database without impacting the availability of the application. After the maintenance is complete and Oracle GoldenGate has finished synchronizing old and new versions, users can be switched to the new version of the application.

- Oracle GoldenGate enables read/write access to a replica database while it is being synchronized with its source database. This is most often used to offload reporting to a copy of a production database when the reporting application requires a read/write connection to database in order to function. This is also relevant, however, to certain disaster recovery environments where the nature of the technology used for the application tier requires an active read/write connection to the DR database at all times in order to meet recovery time objectives. Oracle GoldenGate would be used in this later case in place of Oracle Active Data Guard where the additional data protection, simplicity, and transparency offered by an Oracle Active Data Guard standby is traded in return for a failover target that is always open read/write.

- Multimaster and bidirectional replication architectures where there are multiple databases that all contain the same data and synchronized by Oracle GoldenGate. An update at any database is immediately replicated to all other databases. Update conflicts are either handled by the application, or by conflict handlers configured using Oracle GoldenGate, or are manually resolved. This is an architecture that emphasizes workload balancing and data availability versus simplicity of operation. Each Oracle GoldenGate source can also be protected by a Data Guard standby database for optimal disaster protection. Optionally, due to cost considerations, each Oracle GoldenGate replica can be used to provide both data availability and DR protection to avoid the additional cost of a Data Guard physical standby database.
3.3 Best Practice: Oracle Active Data Guard and Oracle GoldenGate

While Oracle Active Data Guard and Oracle GoldenGate are each capable of maintaining a synchronized copy of an Oracle database, each has unique characteristics that result in high availability architectures that can use one technology or the other, or both at the same time, depending upon requirements. Examples of MAA Best Practice guidelines for use cases relevant to Oracle Database 12c are as follows:

3.3.1 When to Use Oracle Active Data Guard

Use Oracle Active Data Guard when the emphasis is on simplicity, data protection, and availability:

- Simplest, fastest, one-way replication of a complete Oracle database.
- No restrictions: Data Guard Redo Apply supports all data and storage types and Oracle features; transparent replication of DDL
- Features optimized for data protection: Detects silent corruptions that can occur on source or target; automatically repairs corrupt blocks
- Synchronized standby open read-only provides simple read-only offloading for maximum ROI
- Transparency of backups: A Data Guard primary and standby are physically exact copies of each other; RMAN backups are completely interchangeable
- Zero data loss protection at any distance, without impacting database performance
- Minimizing planned downtime and risk using standby first patching, database rolling upgrades, and select platform migrations
- Reduce risk of introducing change by dual purposing a DR system for testing using Data Guard Snapshot Standby
- Integrated automatic database and client failover
- Integrated management of a complete configuration: Data Guard Broker command line interface or Oracle Enterprise Manager Cloud Control

3.3.2 When to Use Oracle GoldenGate

Use Oracle GoldenGate when the emphasis is on advanced replication requirements not addressed by Oracle Active Data Guard:

- Any requirement where the replica database must be open read/write while synchronizing with the primary database
- Any advanced replication requirements such as multimaster and bidirectional replication, subset replication, many-to-one replication, cross-endian replication, and data transformations.
- Maintenance and migrations where zero downtime or near zero downtime is required. Oracle GoldenGate can be used to migrate between application versions, for example, from Application 1.0 to Application 2.0 without downtime.

See Also: Oracle GoldenGate Documentation
- Database rolling upgrades where it is desired to replicate from new version down to the old version for the purpose of fast fall-back if something is wrong with the upgrade.

- Zero downtime planned maintenance where bidirectional replication is used to gradually migrate users to the new version, creating the perception of zero downtime. Note that bidirectional replication requires avoiding or resolving update conflicts that can occur on disparate databases.

### 3.3.3 When to Use Oracle Active Data Guard and Oracle GoldenGate Together

Oracle Active Data Guard and Oracle GoldenGate are not mutually exclusive. The following are use cases of high availability architectures that include the simultaneous use of Oracle Active Data Guard and Oracle GoldenGate:

- An Oracle Active Data Guard standby is utilized for disaster protection and database rolling upgrades for a mission critical OLTP database. At the same time, Oracle GoldenGate is used to extract data from the Data Guard primary database (or from the standby database using Oracle GoldenGate ALO mode) for ETL update of an enterprise data warehouse.

- Oracle GoldenGate subset replication is used to create an operational data store (ODS) that extracts, transforms, and aggregates data from numerous data sources. The ODS supports mission critical application systems that generate significant revenue for the company. An Oracle Active Data Guard standby database is used to protect the ODS, providing optimal data protection and availability.

- Oracle GoldenGate bidirectional replication is utilized to synchronize two databases separated by thousands of miles. User workload is distributed across each database based upon geography, workload, and service level using Oracle 12c Global Data Services (GDS). Each Oracle GoldenGate copy has its own local synchronous Data Guard standby database that enables zero data loss failover if an outage occurs. Oracle GoldenGate capture and apply processes are easily restarted on the new primary database following a failover because the primary and standby are an exact, up-to-date replica of each other.

- An Oracle Active Data Guard standby database utilized steady state for disaster protection is temporarily converted into an Oracle GoldenGate target for the purpose of performing planned maintenance not supported by Data Guard. In this example, assume a Siebel application upgrade requires modification of back-end database objects. The process used is as follows:
  1. Before maintenance is performed, the Data Guard apply process is suspended and a guaranteed restore point is set on both the primary and standby database. For clarity, the original standby database is designated as the upgrade target.
  2. The upgrade target is activated as a primary database, and all database changes required by the new version of the application are implemented.
  3. Meanwhile, the original version of the application is available without disruption to the primary database.
  4. Data Guard transport continues to ship redo for primary transactions to the upgrade target. Redo is not applied and instead is retained in archive logs for protection if something is wrong with the upgrade or if a failover is required before the upgrade is complete. In such a case, the upgrade target can be quickly flashed back to the guaranteed restore point set in Step 1 and converted back into a synchronized standby of the primary database.
5. Oracle GoldenGate capture is also started on the original primary starting at the guaranteed restore point set in Step 1, but replication to the upgrade target is deferred until the maintenance is complete.

6. When all maintenance at the upgrade target is complete, Oracle GoldenGate replication is enabled, and the upgrade target is synchronized with all of the transactions that occurred at the original primary while maintenance was performed.

7. When synchronization is complete, production is switched to the upgrade target, making it the new primary database.

8. The original primary is flashed back to the guaranteed restore point set in step 1 and converted into a physical standby database. Data Guard transport is restarted pointing from the new primary operating at the new version, back to the original primary, which is now a physical standby database.

9. The physical standby is then upgraded using the redo stream received from the new primary. Both systems, primary and physical standby are now at the new version. The only downtime experienced by users is the time required for switchover in Step 7.

Oracle GoldenGate is used for zero downtime or near-zero downtime maintenance not supported by Data Guard in a configuration that operates steady state with a primary database and an Oracle Active Data Guard standby. This differs from the previous scenario by creating a parallel primary/standby environment running on the new platform and version that is completely separate from production. While production continues to run unaffected on the original primary/standby systems, Oracle GoldenGate one-way replication (near-zero downtime) or bidirectional replication (zero downtime) is configured between old and new environments. When Oracle GoldenGate has completed synchronizing old and new environments, production is switched to the new environment and the old environment is decommissioned. This provides zero or minimal downtime depending upon configuration, eliminates risk by providing complete isolation between the old and new environment, and avoids any impact to data protection and availability SLAs if problems are encountered during the upgrade process.

3.4 Recovery Manager

Recovery Manager (RMAN) is an Oracle Database utility to manage database backup and, more importantly, the recovery of the database. RMAN eliminates operational complexity while providing superior performance and availability of the database.

RMAN determines the most efficient method of executing the requested backup, restoration, or recovery operation and then submits these operations to the Oracle Database server for processing. RMAN and the server automatically identify modifications to the structure of the database and dynamically adjust the required operation to adapt to the changes.

RMAN provides the following benefits:

- Support for cross-platform backup and restore (new in Oracle Database 12c)
- Network-enabled restoration allows the RESTORE operations to copy data files directly from one database to another over the network (new in Oracle Database 12c)
- Simplified table restoration with the RECOVER TABLE command (new in Oracle Database 12c)
Oracle Secure Backup

- Automatic channel failover on backup and restore operations
- Automatic failover to a previous backup when the restore operation discovers a missing or corrupt backup
- Automatic creation of new database files and temporary files during recovery
- Automatic recovery through a previous point-in-time recovery—recovery through reset logs
- Block media recovery, which enables the data file to remain online while fixing the block corruption
- Fast incremental backups using block change tracking
- Fast backup and restore operations with intrafile and interfile parallelism
- Enhanced security with a virtual private catalog
- Lower space consumption when creating a database over the network by eliminating staging areas
- Merger of incremental backups into image copies in the background, providing up-to-date recoverability
- Optimized backup and restoration of required files only
- Retention policy to ensure that relevant backups are retained
- Ability to resume backup and restoration of previously failed operations
- Automatic backup of the control file and the server parameter file, ensuring that backup metadata is available in times of database structural changes and media failure and disasters
- Online backup that does not require you to place the database into hot backup mode
- Easily reinstantiate a new database from an existing backup or directly from the production database (thus eliminating staging areas) using the DUPLICATE command.

See Also: Oracle Database Backup and Recovery User’s Guide

3.5 Oracle Secure Backup

Oracle Secure Backup is a centralized tape backup management solution, providing heterogeneous data protection in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. By protecting file system and Oracle Database data, Oracle Secure Backup provides a complete tape backup solution for your IT environment.

Oracle Secure Backup is tightly integrated with RMAN to provide the media management layer for RMAN. With optimized integration points, Oracle Secure Backup and RMAN provide the fastest and most efficient tape backup capability for Oracle Database.

You can back up distributed servers to local and remote tape devices from a central Oracle Secure Backup administrative server using backup policies, calendar-based scheduling for lights out operations, or on-demand backup for immediate requirements. With its highly scalable client/server architecture, Oracle Secure Backup provides local and remote data protection, using Secure Sockets layer (SSL) for secure intradomain communication and two-way server authentication.
Oracle Secure Backup provides the following benefits:

- Optimized performance achieving 25-40% faster Oracle Database backups than comparable media management products with up to 10% less CPU utilization
  - Unused block and undo block compression
  - Shared tape buffers with RMAN
- Policy-based management that allows backup administrators to exercise precise control over the backup domain
- Dynamic drive sharing for increased tape resource use
- Heterogeneous Storage Area Network (SAN) support, enabling NAS, UNIX, Windows, and Linux to share tape drives and media
- File system backup at the file, directory, file system, or raw partition level with full, incremental, and offsite backup scheduling
- Integration with Oracle Enterprise Manager, providing an intuitive, familiar interface
- Backup encryption to tape with policy-based encryption key management leveraging either Oracle Secure Backup host-based encryption or hardware encryption (tape drive)
- Broad tape-device support for new and legacy tape devices in SAN and SCSI environments
- Network Data Management Protocol (NDMP) support for highly efficient backup of NAS files
- Scalable, low-cost licensing model that reduces IT costs and operational considerations
- Enhanced data throughput Reliable Datagram Socket over Remote Direct Memory Access (RDS/RDMA) over Infiniband networks for maximum backup and restore performance in Exadata Database Machine environments
- Oracle-aware backup and restoration on Non-Uniform Memory Access (NUMA) machines, ensuring OSB and Oracle Database background processes communicate in the same NUMA region for optimal performance

**See Also:** Oracle Secure Backup Administrator’s Guide

### 3.6 Oracle Real Application Clusters and Oracle Clusterware

Oracle RAC and Oracle Clusterware enable Oracle Database to run any packaged or custom application across a set of clustered servers. This capability provides the highest levels of availability and the most flexible scalability. If a clustered server fails, then Oracle Database continues running on the surviving servers. When more processing power is needed, you can add another server without interrupting access to data.

**Oracle RAC** enables multiple instances that are linked by an interconnect to share access to an Oracle database. In an Oracle RAC environment, Oracle Database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database system that spans multiple hardware systems, enabling Oracle RAC to provide high availability and redundancy during failures in the cluster. Oracle RAC accommodates all system types, from read-only data warehouse systems to update-intensive online transaction processing (OLTP) systems.
Oracle Clusterware is software that, when installed on servers running the same operating system, enables the servers to be bound together to operate as if they are one server, and manages the availability of user applications and Oracle databases. Oracle Clusterware also provides all of the features required for cluster management, including node membership, group services, global resource management, and high availability functions:

- For high availability, you can place Oracle databases (single-instance or Oracle RAC databases), and user applications (Oracle and non-Oracle) under the management and protection of Oracle Clusterware so that the databases and applications restart when a process fails or so that a failover to another node occurs after a node failure.

- For cluster management, Oracle Clusterware presents multiple independent servers as if they are a single-system image or one virtual server. This single virtual server is preserved across the cluster for all management operations, enabling administrators to perform installations, configurations, backups, upgrades, and monitoring functions. Then, Oracle Clusterware automatically distributes the execution of these management functions to the appropriate nodes in the cluster.

Oracle Clusterware is a requirement for using Oracle RAC. Oracle Clusterware is the only clusterware that you need for most platforms on which Oracle RAC operates. Although Oracle Database continues to support third-party clusterware products on specified platforms, using Oracle Clusterware provides these main benefits:

- Dispenses with proprietary vendor clusterware
- Uses an integrated software stack from Oracle that provides disk management with local or remote Oracle Automatic Storage Management (Oracle Flex ASM) to data management with Oracle Database and Oracle RAC
- Can be configured in large clusters, called an Oracle Flex Cluster.

In addition, Oracle Database features, such as Oracle Service, use the underlying Oracle Clusterware mechanisms to provide their capabilities.

Oracle Clusterware requires two clusterware components: a voting disk to record node membership information and the Oracle Cluster Registry (OCR) to record cluster configuration information. The voting disk and the OCR must reside on shared storage. Oracle Clusterware requires that each node be connected to a private network over a private interconnect.

See Also: Oracle Real Application Clusters Administration and Deployment Guide

### 3.6.1 Benefits of Using Oracle Clusterware

Oracle Clusterware provides the following benefits:

- Tolerates and quickly recovers from computer and instance failures.
- Simplifies management and support by means of using Oracle Clusterware together with Oracle Database. By using fewer vendors and an all Oracle stack you gain better integration compared to using third-party clusterware.
- Performs rolling upgrades for system and hardware changes. For example, you can apply Oracle Clusterware upgrades, patch sets, and interim patches in a rolling fashion.

When you upgrade to Oracle Database 12c, Oracle Clusterware and Oracle ASM binaries are installed as a single binary called the Oracle Grid Infrastructure. You
can upgrade Oracle Clusterware in a rolling manner from Oracle Clusterware 10g and Oracle Clusterware 11g; however, you can only upgrade Oracle ASM in a rolling manner from Oracle Database 11g release 1 (11.1).

- Automatically restarts failed Oracle processes.
- Automatically manages the virtual IP (VIP) address. When a node fails, the node’s VIP address fails over to another node on which the VIP address can accept connections.
- Automatically restarts resources from failed nodes on surviving nodes.
- Controls Oracle processes as follows:
  - For Oracle RAC databases, Oracle Clusterware controls all Oracle processes by default.
  - For Oracle single-instance databases, Oracle Clusterware enables you to configure the Oracle processes into a resource group that is under the control of Oracle Clusterware.
- Provides an application programming interface (API) for Oracle and non-Oracle applications that enables you to control other Oracle processes with Oracle Clusterware, such as restart or react to failures and certain rules.
- Manages node membership and prevents split-brain syndrome in which two or more instances attempt to control the database.
- Provides the ability to perform rolling release upgrades of Oracle Clusterware, with no downtime for applications.

**See Also:** *Oracle Clusterware Administration and Deployment Guide*

### 3.6.2 Benefits of Using Oracle Real Application Clusters and Oracle Clusterware

Together, Oracle RAC and Oracle Clusterware provide all of the Oracle Clusterware benefits listed in Section 3.6.1 plus the following benefits:

- Provides better integration and support of Oracle Database by using an all Oracle software stack compared to using third-party clusterware.
- Relocate Oracle Service automatically. Plus, when you perform additional fast application notification (FAN) and client configuration, distribute FAN events so that applications can react immediately to achieve fast, automatic, and intelligent connection and failover.
- Detect connection failures fast and automatically, and remove terminated connections for any Java application using Oracle Universal Connection Pool (Oracle UCP) Fast Connection Failover and FAN events.
- Balance work requests using Oracle UCP runtime connection load balancing.
- Use runtime connection load balancing with Oracle UCP, Oracle Call Interface (OCI), and Oracle Data Provider for .NET (ODP.NET).
- Distribute work across all available instances using load balancing advisory.
- Allow the flexibility to increase processing capacity using commodity hardware without downtime or changes to the application.
- Provide comprehensive manageability integrating database and cluster features.
- Provide scalability across database instances.
- Implement Fast Connection Failover for nonpooled connections.
3.7 Oracle RAC One Node

Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle RAC database that runs on one node in a cluster. This feature enables you to consolidate many databases into one cluster with minimal overhead, protecting them from both planned and unplanned downtime. The consolidated databases reap the high availability benefits of failover protection, online rolling patch application, and rolling upgrades for the operating system and Oracle Clusterware.

Oracle RAC One Node enables better availability than cold failover for single-instance databases because of the Oracle technology called online database relocation, which intelligently migrates database instances and connections to other cluster nodes for high availability and load balancing. Online database relocation is performed using the Server Control Utility (SRVCTL).

Oracle RAC One Node provides the following:
- Always available single-instance database services
- Built-in cluster failover for high availability
- Live migration of instances across servers
- Online rolling patches and rolling upgrades for single-instance databases
- Online upgrade from single-instance to multiple-instance Oracle RAC
- Better consolidation for database servers
- Enhanced server virtualization
- Lower cost development and test platform for full Oracle RAC
- Relocation of Oracle RAC primary and standby databases configured with Data Guard. This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2).

Oracle RAC One Node also facilitates the consolidation of database storage, standardizes your database environment, and, when necessary, enables you to transition to a full, multiple-instance Oracle RAC database without downtime or disruption.

See Also: Section 7.1.2, "Oracle Database with Oracle RAC One Node."

3.8 Oracle Automatic Storage Management

Oracle ASM provides a vertically integrated file system and volume manager directly in the Oracle Database kernel, resulting in:
- Significantly less work to provision database storage
- Higher level of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, Oracle ASM spreads files across all available storage. To protect against data loss, Oracle ASM extends the concept of SAME (stripe and mirror everything) and adds more flexibility because it can mirror at the database file level rather than at the entire disk level.
More important, Oracle ASM simplifies the processes of setting up mirroring, adding disks, and removing disks. Instead of managing hundreds or possibly thousands of files (as in a large data warehouse), DBAs using Oracle ASM create and administer a larger-grained object called a disk group. The disk group identifies the set of disks that are managed as a logical unit. Automation of file naming and placement of the underlying database files save administrators time and ensure adherence to standard best practices.

The Oracle ASM native mirroring mechanism (two-way or three-way) protects against storage failures. With Oracle ASM mirroring, you can provide an additional level of data protection with the use of failure groups. A failure group is a set of disks sharing a common resource (disk controller or an entire disk array) whose failure can be tolerated. After it is defined, an Oracle ASM failure group intelligently places redundant copies of the data in separate failure groups. This ensures that the data is available and transparently protected against the failure of any component in the storage subsystem.

By using Oracle ASM, you can:

- Mirror and stripe across drives and storage arrays.
- Automatically remirror from a failed drive to remaining drives.
- Automatically rebalance stored data when disks are added or removed while the database remains online.
- Allow for operational simplicity in managing database storage.
- Manage the Oracle Cluster Registry (OCR) and voting disks.
- Provide preferred read capability on disks that are local to the instance, which gives better performance for an extended cluster.
- Support very large databases.
- Support Oracle ASM rolling upgrades.
- Improve availability and reliability using the Oracle ASM disk scrubbing process to find and repair logical data corruptions using mirror disks.
- Support finer granularity in tuning and security.
- Provide fast repair after a temporary disk failure through Oracle ASM Fast Mirror Resync and automatic repair of block corruptions if a good copy exists in one of the mirrors.
- Provide disaster recovery capability for the file system by enabling replication of Oracle ACFS across the network to a remote site.
- Patch the Oracle ASM instance without impacting the clients that are being serviced using Oracle Flex ASM. A database instance can be directed to access Oracle ASM metadata from another location while the current Oracle ASM instance it is connected to is taken offline for planned maintenance.
- Monitor and manage the speed and status of Oracle ASM Disk Resync and Rebalance operations.
- Bring online multiple disks simultaneously and manage performance better by controlling resync parallelism using the Oracle ASM Resync Power Limit. Recover faster after a cell or disk failure, and the instance doing the resync is failing; this is made possible by using a Disk Resync Checkpoint which enables a resync to
resume from where it was interrupted or stopped instead of starting from the beginning.

- Automatically connect database instances to another Oracle ASM instance using Oracle Flex ASM. The local database instance can still access the required metadata and data if an Oracle ASM instance fails due to an unplanned outage.

**See Also:** *Oracle Automatic Storage Management Administrator’s Guide* for more information about ACFS

### 3.9 Fast Recovery Area

The **fast recovery area** is a unified storage location for all recovery-related files and activities in Oracle Database. After this feature is enabled, all RMAN backups, archived redo log files, control file autobackups, flashback logs, and data file copies are automatically written to a specified file system or Oracle ASM disk group, and the management of this disk space is handled by RMAN and the database server.

Performing a backup to disk is faster because using the fast recovery area eliminates the bottleneck of writing to tape. More important, if database media recovery is required, then data file backups are readily available. Restoration and recovery time is reduced because you do not need to find a tape and a free tape device to restore the needed data files and archived redo log files.

The fast recovery area provides the following benefits:

- Unified storage location of related recovery files
- Management of the disk space allocated for recovery files, which simplifies database administration tasks
- Fast, reliable, disk-based backup and restoration

**See Also:** *Oracle Database Backup and Recovery User’s Guide*

### 3.10 Corruption Prevention, Detection, and Repair

Data block corruptions can be very disruptive and challenging to repair (see Section 1.4, "Causes of Downtime"). Corruptions can cause serious application and database downtime when encountered and worse yet it can go undetected for hours, days and even weeks leading to even longer application downtime once detected. Unfortunately, there is not one way to comprehensively prevent, detect, and repair data corruptions within the database because the source and cause of corruptions can be anywhere in memory, hardware, firmware, storage, operating system, software, or user error. Worse yet, third-party solutions that do not understand Oracle data block semantics and how Oracle changes data blocks do not prevent and detect data block corruptions well. Third party remote mirroring technologies can propagate data corruptions to the database replica (standby) leading to a double failure, data loss, and much longer downtime.

Oracle MAA has a comprehensive plan to prevent, detect, and repair all forms of data block corruptions including physical block corruptions, logical block corruptions, stray writes, and lost writes. These additional safeguards provide the most comprehensive Oracle data block corruption prevention, detection, and repair solution. Details of this plan are described in the MAA white paper "Preventing, Detecting, and Repairing Block Corruption: Oracle Database." The solution centers around Oracle Active Data Guard and three database parameters; however, there is much more that Oracle recommends in dealing with data corruptions.
The following summary highlights the key points:

- **Set** `DB_BLOCK_CHECKSUM`, `DB_BLOCK_CHECKING` and `DB_LOST_WRITE_PROTECT` database initialization parameters on both the primary and the standby to detect and prevent the majority of physical block corruptions and logical block corruptions.
- Use Oracle Active Data Guard to detect and fail over to a standby database when physical block corruptions, logical block corruptions, or lost writes appear on the primary database. Oracle Active Data Guard with real-time apply provides an automatic block repair function for data block corruptions that can eliminate any downtime for physical block corruptions.
- Use Oracle ASM to detect physical corruptions in storage and automatic repair if there is a good mirror copy.
- Use Exadata storage and its comprehensive HARD checks to prevent and detect data block corruptions originating in the storage I/O subsystem.
- Use RMAN backup and restore operations to detect any physical block corruptions especially with infrequently queried data. Use the RMAN “check logical” option to detect logical block corruptions.
- Use Data Recovery Advisor to detect and repair data corruption.
- Proactively query `V$DATABASE_BLOCK_CORRUPTION` periodically for any detected data block corruptions.
- Use the `ANALYZE` statement with the `VALIDATE STRUCTURE` option to evaluate interobject or interblock corruptions.

**See Also:**

- *Oracle Database Reference* for more information about the views and initialization parameters

### 3.11 Data Recovery Advisor

**Data Recovery Advisor** automatically diagnoses persistent (on-disk) data failures, presents appropriate repair options, and runs repair operations at your request.

You can use Data Recovery Advisor to troubleshoot primary databases, logical standby databases, physical standby databases, and snapshot standby databases.

Data Recovery Advisor includes the following functionality:

- **Failure diagnosis**
  
  The first symptoms of database failure are usually error messages, alarms, trace files and dumps, and failed health checks. Assessing these symptoms can be complicated, error-prone, and time-consuming. Data Recovery Advisor automatically diagnoses data failures and informs you about them.

- **Failure impact assessment**
  
  After a failure is diagnosed, you must understand its extent and assess its impact on applications before devising a repair strategy. Data Recovery Advisor
automatically assesses the impact of a failure and displays it in an easily understood format.

- **Repair generation**
  
  Even if a failure was diagnosed correctly, selecting the correct repair strategy can be error-prone and stressful. Moreover, there is often a high penalty for making poor decisions in terms of increased downtime and loss of data. Data Recovery Advisor automatically determines the best repair for a set of failures and presents it to you.

- **Repair feasibility checks**
  
  Before presenting repair options, Data Recovery Advisor validates them with respect to the specific environment and availability of media components required to complete the proposed repair, including restoring files directly from the primary or standby database to complete the proposed repair.

- **Repair automation**
  
  If you accept the suggested repair option, Data Recovery Advisor automatically performs the repair, verifies that the repair was successful, and closes the appropriate failures.

- **Validation of data consistency and database recoverability**
  
  Data Recovery Advisor can validate the consistency of your data, and backups and redo stream, whenever you choose.

- **Early detection of corruption**
  
  Through Health Monitor, you can schedule periodic runs of Data Recovery Advisor diagnostic checks to detect data failures before a database process executing a transaction discovers the corruption and signals an error. Early warnings can limit the damage caused by corruption.

- **Integration of data validation and repair**
  
  Data Recovery Advisor is a single tool for data validation and repair.

---

**Note:** Data Recovery Advisor only supports single-instance databases. Oracle RAC databases are not supported. See *Oracle Database Backup and Recovery User’s Guide* for more information about Data Recovery Advisor supported database configurations.

---

**See Also:** *Oracle Database Backup and Recovery User’s Guide*

### 3.12 Oracle Security Features

The best protection against human errors is to prevent their occurrence. The best way to prevent human errors is to restrict user access to only those data and services required to perform business functions. Oracle Database provides a wide range of security tools to control access to application data by authenticating database users and then enabling administrators to grant them only those privileges required to perform their duties.

In addition, the Oracle Database security model provides the ability to restrict data access at a row level using Oracle Virtual Private Database, thereby further isolating database users from data that they do not need to access.

Oracle Database provides the following security benefits:
- Authentication control to validate the identities of entities using networks, databases, and applications. Network sessions between databases, such as redo transport sessions, are also authenticated.

- Authorization control to provide limits to access and actions linked by database user identities and roles.

- Access control to objects, providing protection regardless of the entity seeking to access or alter them.

- Auditing control to monitor and gather data about specific database activities, investigate suspicious activity, deter users (or others) from inappropriate activities, and detect problems with authorization or access control implementation.

- Security policy management using profiles.

- Encryption of data residing in the database and backups, or transferred to and from databases.

- Administration of Data Guard configurations can be delegated to a class of users who would not be granted SYSDBA privileges.

**See Also:**

- *Oracle Database Security Guide*
- *Oracle Data Guard Concepts and Administration*

### 3.13 Oracle Flashback Technology

Oracle Flashback technology is a group of Oracle Database features that let you view past states of database, database objects, transactions or rows or to rewind the database, database objects, transactions or rows to a previous state without using point-in-time media recovery.

With flashback features, you can:

- Perform queries to show data as it looked at a previous point in time
- Perform queries that return metadata that shows a detailed history of changes to the database
- Recover tables or rows to a previous point in time
- Automatically track and archive transactional data changes
- Roll back a transaction and its dependent transactions while the database remains online
- Undrop a table
- Recover a database to a point-in-time without a restore operation

Other than the flashback database feature, most Oracle Flashback features use the Automatic Undo Management (AUM) system to obtain metadata and historical data for transactions. They rely on undo data, which are records of the effects of individual transactions. For example, if a user runs an UPDATE statement to change a salary from 1000 to 1100, then Oracle Database stores the value 1000 in the undo data.

Undo data is persistent and survives a database shutdown. By using flashback features, you can use undo data to query past data or recover from logical damage. Besides using it in flashback features, Oracle Database uses undo data to perform these actions:
- Roll back active transactions
- Recover terminated transactions by using database or process recovery
- Provide read consistency for SQL queries

Oracle Flashback can address and rewind data that is compromised due to various human or operator errors that inadvertently or maliciously change data, cause bad installations and upgrades, and result in logical errors in applications. These problems are addressed in the following phases, and use features such as flashback transaction, flashback drop, flashback table, and flashback database.

Phase 1: Detection of logical failure, which is usually done by the application.
Phase 2: Error investigation using features such as flashback query, flashback version query, and flashback transaction query and the DBMS_FLASHBACK package.
Phase 3: Error recovery.

### 3.13.1 Oracle Flashback Query

Oracle Flashback Query (Flashback Query) provides the ability to view data as it existed in the past by using the Automatic Undo Management system to obtain metadata and historical data for transactions. Undo data is persistent and survives a database malfunction or shutdown. The unique features of Flashback Query not only provide the ability to query previous versions of tables, they also provide a powerful mechanism to recover from erroneous operations.

Uses of Flashback Query include:
- Recovering lost data or undoing incorrect, committed changes. For example, rows that were deleted or updated can be immediately repaired even after they were committed.
- Comparing current data with the corresponding data at some time in the past. For example, by using a daily report that shows the changes in data from yesterday, it is possible to compare individual rows of table data, or find intersections or unions of sets of rows.
- Checking the state of transactional data at a particular time, such as verifying the account balance on a certain day.
- Simplifying application design by removing the need to store certain types of temporal data. By using Flashback Query, it is possible to retrieve past data directly from the database.
- Applying packaged applications, such as report generation tools, to past data.
- Providing self-service error correction for an application, enabling users to undo and correct their errors.

See Also: Oracle Database Development Guide

### 3.13.2 Oracle Flashback Version Query

Oracle Flashback Version Query is an extension to SQL that you can use to retrieve the versions of rows in a given table that existed at a specific time interval. Oracle Flashback Version Query returns a row for each version of the row that existed in the specified time interval. For any given table, a new row version is created each time the COMMIT statement is executed.

Oracle Flashback Version Query is a powerful tool that database administrators (DBAs) can use to run analysis to determine the source of problems. Additionally,
application developers can use Oracle Flashback Version Query to build customized applications for auditing purposes.

**See Also:** *Oracle Database Development Guide*

### 3.13.3 Oracle Flashback Transaction

Oracle Flashback Transaction backs out a transaction and its dependent transactions. The `DBMS_FLASHBACK.TRANSACTION_BACKOUT()` procedure rolls back a transaction and its dependent transactions while the database remains online. This recovery operation uses undo data to create and execute the compensating transactions that return the affected data to its original state. You can query the `DBA_FLASHBACK_TRANSACTION_STATE` view to see whether the transaction was backed out using dependency rules or forced out by either:

- Backing out nonconflicting rows
- Applying undo SQL

Oracle Flashback Transaction increases availability during logical recovery by quickly backing out a specific transaction or set of transactions and their dependent transactions. You use one command to back out transactions while the database remains online.

**See Also:**

- *Oracle Database Development Guide*
- *Oracle Database PL/SQL Packages and Types Reference*

### 3.13.4 Oracle Flashback Transaction Query

Oracle Flashback Transaction Query provides a mechanism to view all of the changes made to the database at the transaction level. When used in conjunction with Oracle Flashback Version Query, it offers a fast and efficient means to recover from a human or application error. Oracle Flashback Transaction Query increases the ability to perform online diagnosis of problems in the database by returning the database user that changed the row, and performs analysis and audits on transactions.

**See Also:** *Oracle Database Development Guide*

### 3.13.5 Oracle Flashback Table

Oracle Flashback Table recovers a table to a previous point in time. It provides a fast, online solution for recovering a table or set of tables that were changed by a human or application error. In most cases, Oracle Flashback Table alleviates the need for administrators to perform more complicated point-in-time recovery operations. The data in the original table is not lost when you use Oracle Flashback Table because you can return the table to its original state.

**See Also:** *Oracle Database Backup and Recovery User’s Guide*

### 3.13.6 Oracle Flashback Drop

Dropping objects by accident is a problem for database users and database administrators. Although there is no easy way to recover dropped tables, indexes, constraints, or triggers, Oracle Flashback Drop provides a safety net when you are dropping objects. When you drop a table, it is automatically placed into the Recycle
Bin. The Recycle Bin is a virtual container where all dropped objects reside. You can continue to query data in a dropped table.

See Also: *Oracle Database Backup and Recovery User’s Guide*

### 3.13.7 Oracle Flashback Restore Points

When an Oracle Flashback recovery operation is performed on the database, the DBA must determine the point in time—identified by the system change number (SCN) or time stamp—to which you can later flash back the data. Oracle Flashback restore points are labels that you can define to substitute for the SCN or transaction time used in Flashback Database, Flashback Table, and Oracle Recovery Manager (RMAN) operations. Furthermore, a database can be flashed back through a previous database recovery and opened with an `OPEN RESETLOGS` command by using guaranteed restore points. Guaranteed restore points allow major database changes—such as database batch jobs, upgrades, or patches—to be quickly undone by ensuring that the undo required to rewind the database is retained.

Using the Oracle Flashback restore points feature provides the following benefits:

- The ability to quickly restore to a consistent state, to a time before a planned operation that has gone awry (for example, a failed batch job, an Oracle software upgrade, or an application upgrade)
- The ability to resynchronize a snapshot standby database with the primary database
- A quick mechanism to restore a test or cloned database to its original state

See Also: *Oracle Database Backup and Recovery User’s Guide*

### 3.13.8 Oracle Flashback Database

Oracle Flashback Database is the equivalent of a fast rewind button, quickly returning a database to a previous point in time without requiring a time consuming restore and roll forward using a backup and archived logs. The larger the size of the database, the greater the advantage of using Oracle Flashback Database for fast point in time recovery.

Enabling Oracle Flashback Database provides the following benefits:

- Fast point in time recovery to repair logical corruptions, such as those caused by administrative error.
- Useful for iterative testing when used with Oracle restore points. A restore point can be set, database changes implemented, and test workload run to assess impact. Oracle Flashback Database can then be used to discard the changes and return the database to the original starting point, different modifications can be made, and the same test workload run a second time to have a true basis for comparing the impact of the different configuration changes.
- Data Guard uses Oracle Flashback Database to quickly reinstantiate a failed primary database as a new standby (after a failover has occurred), without requiring the failed primary to be restored from a backup.

See Also:

- *Oracle Database Backup and Recovery User’s Guide*
- *Oracle Database SQL Language Reference*
3.13.9 Block Media Recovery Using Flashback Logs or Physical Standby Database

After attempting to automatically repair corrupted blocks, block media recovery can optionally retrieve a more recent copy of a data block from the flashback logs to reduce recovery time. Automatic block repair allows corrupt blocks on the primary database to be automatically repaired as soon as they are detected, by using good blocks from a physical standby database.

Furthermore, a corrupted block encountered during instance recovery does not result in instance recovery failure. The block is automatically marked as corrupt and added to the RMAN corruption list in the \texttt{V$DATABASE\_BLOCK\_CORRUPTION} table. You can subsequently issue the RMAN \texttt{RECOVER BLOCK} command to fix the associated block. In addition, the RMAN \texttt{RECOVER BLOCK} command restores blocks from a physical standby database, if it is available.

See Also:

- \textit{Oracle Database Backup and Recovery User's Guide} for block media repair
- \textit{Oracle Database Backup and Recovery Reference} for the RMAN \texttt{RECOVER BLOCK} command

3.13.10 Flashback Data Archive

The Flashback Data Archive is stored in a tablespace and contains transactional changes to every record in a table for the duration of the record’s lifetime. The archived data can be retained for a much longer duration than the retention period offered by an undo tablespace, and used to retrieve very old data for analysis and repair.

See Also: \textit{Oracle Database Development Guide}

3.14 Oracle Data Pump and Data Transport

Oracle Data Pump technology enables very high-speed movement of data and metadata from one database to another. Data Pump is used to perform the following planned maintenance activities:

- Database migration to a different platform
- Database migration to pluggable databases
- Database upgrade

See Section 5.4, "Oracle High Availability Solutions for System and Software Maintenance" for more information about using this technology for planned maintenance.

The Data Pump features that enable the planned maintenance activities listed above are the following:

- Full transportable export/import to move an entire database to a different database instance
- Transportable tablespaces to move a set of tablespaces between databases

3.15 Oracle Replication Technologies for Non-Database Files

Table 3–1 describes the Oracle replication technologies for non-database files.
Oracle DBFS provides the following benefits:

Table 3–1  Oracle Replication Technologies for Non-Database Files

<table>
<thead>
<tr>
<th>Technology</th>
<th>Recommended Usage</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database File System</td>
<td>Recommended for Exadata Database Machine systems or when you need full synchronization between database and non-database systems</td>
<td>Can be integrated with the database to maintain complete consistency between the database changes and the file system changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All data stored in the database and can be used with Oracle Active Data Guard to provide both disaster recovery and read-only access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can take advantage all of the Oracle database features</td>
</tr>
<tr>
<td>Oracle ASM Cluster File System</td>
<td>Recommended to provide a single-node and cluster-wide file system solution integrated with Oracle ASM, Oracle Clusterware, and Oracle Enterprise Manager technologies</td>
<td>Oracle ACFS establishes and maintains communication with the Oracle ASM instance to participate in Oracle ASM state transitions including Oracle ASM instance and disk group status updates and disk group rebalancing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Supports many database and application files, including executables, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and other general-purpose application file data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can provide near-time consistency between database changes and file system changes when point-in-time recovery happens</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can be exported and accessed by remote clients using standard NAS File Access Protocols such as NFS and CIFS.</td>
</tr>
<tr>
<td>Oracle Solaris ZFS Storage Appliance Replication</td>
<td>Recommended for disaster recovery protection for non-database files, and specifically for Oracle Fusion Middleware critical files stored outside of the database.</td>
<td>Replicates all non-database objects, including Oracle Fusion Middleware binaries configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Can provide near time consistency between database changes and file system changes when point-in-time recovery happens</td>
</tr>
</tbody>
</table>

3.15.1 Oracle Database File System

Oracle Database File System (DBFS) takes advantage of the features of the database to store files, and the strengths of the database in efficiently managing relational data, to implement a standard file system interface for files stored in the database. With this interface, storing files in the database is no longer limited to programs specifically written to use BLOB and CLOB programmatic interfaces. Files in the database can now be transparently accessed using any operating system (OS) program that acts on files. For example, extract, transform, and load (ETL) tools can transparently store staging files in the database.

Oracle DBFS provides the following benefits:
- Full stack integration recovery and failover: By storing file system files in a database structure, it is possible to easily perform point-in-time recovery of both database objects and file system data.

- Disaster Recovery System Return on Investment (ROI): All changes to files contained in DBFS are also logged through the Oracle database redo log stream and thus can be passed to a Data Guard physical standby database. Using Oracle Active Data Guard technology, the DBFS file system can be mounted read-only using the physical standby database as the source. Changes made on the primary are propagated to the standby database and are visible once applied to the standby.

- File system backups: Because DBFS is stored in the database as database objects, standard RMAN backup and recovery functionality can be applied to file system data. Any backup, restore, or recovery operation that can be performed on a database or object within a database can also be performed against the DBFS file system.

### 3.15.2 Oracle ASM Cluster File System

Oracle ASM Cluster File System (ACFS) is a multiplatform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support customer files maintained outside of Oracle Database. Oracle ACFS supports many database and application files, including executables, database trace files, database alert logs, application reports, BFILEs, and configuration files. Other supported files are video, audio, text, images, engineering drawings, and other general-purpose application file data.

Oracle ACFS takes advantage of the following Oracle ASM functionality:

- Oracle ACFS dynamic file system resizing
- Maximized performance through direct access to Oracle ASM disk group storage
- Balanced distribution of Oracle ACFS across Oracle ASM disk group storage for increased I/O parallelism
- Data reliability through Oracle ASM mirroring protection mechanisms

An additional feature of Oracle ACFS is Oracle ACFS Replication which, similar to Data Guard for the database, enables replication of Oracle ACFS file systems across the network to a remote site, providing disaster recovery capability for the file system. Oracle ACFS replication captures file system changes written to disk for a primary file system and records the changes in files called replication logs. These logs are transported to the site hosting the associated standby file system where background processes read the logs and apply the changes recorded in the logs to the standby file system. After the changes recorded in a replication log are successfully applied to the standby file system, the replication log is deleted from the sites hosting the primary and standby file systems.

### 3.15.3 Oracle Solaris ZFS Storage Appliance Replication

The Oracle Solaris ZFS Storage Appliance series supports snapshot-based replication of projects and shares from a source appliance to any number of target appliances manually, on a schedule, or continuously for the following use cases:

- Disaster recovery: Replication can be used to mirror an appliance for disaster recovery. In the event of a disaster that impacts the service of the primary appliance (or even an entire data center), administrators activate the service at the disaster recovery site, which takes over using the most recently replicated data.
When the primary site is restored, data changed while the disaster recovery site was in service can be migrated back to the primary site, and normal service is restored. Such scenarios are fully testable before a disaster occurs.

- Data distribution: Replication can be used to distribute data (such as virtual machine images or media) to remote systems across the world in situations where clients of the target appliance would not ordinarily be able to reach the source appliance directly, or such a setup would have prohibitively high latency. One example uses this scheme for local caching to improve latency of read-only data (such as documents).

- Disk-to-disk backup: Replication can be used as a backup solution for environments in which tape backups are not feasible. Tape backup might not be feasible, for example, because the available bandwidth is insufficient or because the latency for recovery is too high.

- Data migration: Replication can be used to migrate data and configuration between Oracle Solaris ZFS Storage appliances when upgrading hardware or rebalancing storage. Shadow migration can also be used for this purpose.

The architecture of Oracle Solaris ZFS Storage Appliance also makes it an ideal platform to complement Data Guard for disaster recovery of Oracle Fusion Middleware. Oracle Fusion Middleware has a number of critical files that are stored outside of the database. These binaries, configuration data, metadata, logs and so on also require data protection to ensure availability of the Oracle Fusion Middleware. For these, the built-in replication feature of the ZFS Storage Appliance is used to move this data to a remote disaster recovery site.

Benefits of the Oracle Solaris ZFS Storage Appliance when used with Oracle Fusion Middleware include:

- Leverages remote replication for Oracle Fusion Middleware
- Provides ability to quickly create clones and snapshots of databases to increase ROI of DR sites

### 3.16 Client and Application Failover

A highly available architecture requires the application tier to transparently fail over to a surviving instance or database advertising the required service. This ensures that applications are generally available or minimally impacted in the event of node failure, instance failure, data corruption, or database failures. Transparent client failover enables applications to fail over to another available Oracle RAC instance or to another database (such as in the case of a Data Guard role transition or Oracle GoldenGate).

Client failover encompasses failure notification, connection cleanup, and automatic reconnection and retries of database service residing in another Oracle RAC instance or database and possibly query retry.

At a high level, the following components are used to provide for seamless client failover:

- **Services**

  Oracle Database provides a powerful automatic workload management facility, called services, to enable the enterprise grid vision. Services are entities that you can define in Oracle databases that enable you to group database workloads, route work to the optimal instances that are assigned to offer the service, and achieve high availability for planned and unplanned actions.
■ **High Availability Framework**
An Oracle RAC component that enables Oracle Database to maintain components in a running state.

■ **Fast Application Notification (FAN)**
FAN is a high availability notification mechanism that Oracle RAC uses to notify other processes about configuration-level and service-level information that includes service status changes, such as UP or DOWN events. The Oracle client drivers and Oracle connection pools respond to FAN events and take immediate action. FAN UP and DOWN events can apply to instances, services, and nodes.

■ **Transaction Guard**
Transaction Guard is a tool that provides a protocol and an API for at-most-once execution of transactions in case of unplanned outages and duplicate submissions.

■ **Application Continuity**
Application Continuity provides a general purpose infrastructure that replays the in-flight request when a recoverable error is received, masking many system, communication, and storage outages and hardware failures. Unlike existing recovery technologies, this feature attempts to recover the transactional and non-transactional session states beneath the application, so that the outage appears to the application as a delayed execution.

■ **Connection Load Balancing**
Connection Load Balancing is a feature of Oracle Net Services that balances incoming connections across all of the instances that provide the requested database service.

■ **Fast Connection Failover**
Fast Connection Failover is the ability of Oracle Clients to provide rapid failover of connections by subscribing to FAN events.

■ **Transparent Application Failover (TAF)**
Transparent Application Failover is a run-time failover for high availability environments that refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and, optionally, resume a SELECT statement that was in progress. This reconnection happens automatically from within the Oracle Call Interface (OCI) library.

■ **Single Client Access Name (SCAN)**
SCAN provides a single name to the clients connecting to Oracle RAC that does not change throughout the life of the cluster, even if you add or remove nodes from the cluster. Clients connecting with SCAN can use a simple connection string, such as a thin JDBC URL or EZConnect, and still achieve the load balancing and client connection failover.

■ **Global Data Services**
Global Data Services (GDS) is a new capability of Oracle Database that extends the concept of services to a globally replicated configuration involving a combination of single-instance, Oracle RAC, Oracle Active Data Guard, and Oracle GoldenGate. This enables services to be deployed anywhere within this globally replicated configuration, supporting load balancing, high availability, database affinity, and so on.
Connection Time Failover

Oracle Net supports connect descriptors with multiple lists of addresses, each with its own characteristics. Connection time failover allows for a new connection attempt to fail over to a different address if the connection to the first address fails.

See Also:
- Oracle Database Concepts for information about how the database processes transactions
- Oracle Real Application Clusters Administration and Deployment Guide for information about Dynamic Database Services
- Oracle Database 2 Day + Real Application Clusters Guide for information about Dynamic Database Services
- Oracle Database Global Data Services Concepts and Administration Guide

3.16.1 Client Failover Processing for Connections

At a high level, automating client failover in an MAA environment includes relocating database services to available resources, notifying clients that a failure has occurred, potentially breaking them out of TCP timeout, and redirecting application connections to available resources where the database service is active. The components described in the introduction to this chapter that are used to process the failover of application connections depend on the configuration of your MAA environment.

Table 3–2  Client Failover Processing for Connections

<table>
<thead>
<tr>
<th>MAA Configuration</th>
<th>Service Relocation</th>
<th>Application Notification</th>
<th>Session Failover and Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Instance with Data Guard</td>
<td>■ Start service after Data Guard failover using a trigger written on the DB_ROLE_CHANGE system event  ■ Use Global Data Services</td>
<td>Configure your operating system for efficient TCP timeouts on the hosts that run the application layer</td>
<td>Configure Transparent Oracle Failover (TAF) for OCI clients. If not using TAF, you can include Transaction Guard in your application for OCI, JDBC Thin, or ODP.</td>
</tr>
<tr>
<td>Oracle RAC Database or Oracle RAC One Node</td>
<td>■ Use services managed by Oracle Clusterware</td>
<td>Configure for Fast Application Notification</td>
<td>Configure Transparent Oracle Failover (TAF) for OCI clients. Configure Application Continuity for Thin JDBC Clients., If not using these, you may include Transaction Guard in your application for OCI, JDBC Thin, or ODP. (TAF and AC include Transaction Guard)</td>
</tr>
</tbody>
</table>
The following sections provide more information about service relocation and application notification.

### 3.16.1.1 Services

A service name is a logical representation of a service used for client connections. When a client connects to a listener, it requests a connection to a service. When a database instance starts, it registers itself with a listener as providing one or more services by name. A single service, as known by a listener, can identify one or more database instances in an Oracle RAC or Data Guard environment. A single database instance can register one or more services with a listener.

#### 3.16.1.1.1 Service Usage in a Single-Instance Database and Data Guard Environment

The application should connect to the database using a primary specific service name, that is a user-created service that is only active on the primary database. In the event of a Data Guard failover, this service migrates to any database that currently holds the primary role. This can be accomplished in single-instance environments that do not have Oracle Clusterware installed by creating a trigger that executes based on the ON\_STARTUP system event. This trigger should check the DATABASE_ROLE value of the V\$DATABASE view, and if the value is PRIMARY, then start the user created service.

#### 3.16.1.1.2 Service Usage in an Oracle RAC Database Environment

Resource profiles are automatically created when you define a service. A resource profile describes how Oracle Clusterware should manage the service and which instance the service should failover to if the preferred instance stops. Resource profiles also define service dependencies for the instance and the database. Due to these dependencies, if you stop a database, then the instances and services are automatically stopped in the correct order.

When you define a service for an administrator-managed database, you define which instances usually support that service using SRVCTL with the -preferred parameter.
These are known as the preferred instances. You can also define other instances to support a service if the service’s preferred instance fails using SRVCTL with the -available parameter. These are known as available instances.

When you specify preferred instances, you are specifying the number of instances on which a service usually runs. This is the maximum cardinality of the service. Oracle Clusterware attempts to ensure that the service runs on the number of instances for which you have configured the service. Afterward, due to either instance failure or planned service relocations, a service may be running on an available instance.

If an instance fails, then you cannot control to which available instance Oracle Clusterware relocates the services if there are multiple instances in the list. During a planned operation, however, you can manually direct the service to any instance in either the preferred or the available list not currently offering the service.

3.16.1.1.3 Service Usage in an Oracle RAC Database and Data Guard Environment

If you configured Data Guard in your Oracle RAC environment, then you can define a role for each service using SRVCTL with the -l parameter. When you specify a role for a service, Oracle Clusterware automatically starts the service only when the database role matches the role you specified for the service. Valid roles are PRIMARY, PHYSICAL_STANDBY, LOGICAL_STANDBY, and SNAPSHOT_STANDBY and you can specify more than one role for a service.

If multiple databases in the cluster offer the same service name, then Oracle RAC balances connections to that service across all such databases. This is useful for standby and active Data Guard databases, but if you want client connections to a service to be directed to a particular database, then the service name must be unique within the cluster (not offered by any other database).

See Also: Oracle Data Guard Concepts and Administration for more information about database roles.

3.16.1.1.4 Service Usage in a Replicated Environment or Oracle Active Data Guard Environment

The Global Data Services framework is the software infrastructure for global services. This framework automates and centralizes configuration, maintenance, and monitoring of a database cloud, and enables load balancing and failover for global services. The framework manages these virtualized resources with minimal administrative overhead, enabling the cloud to handle additional client requests.

The Global Data Services framework is built around the following preexisting Oracle Database technologies:

- Oracle Active Data Guard
  Enables high-performance farms of read-only databases.

- Data Guard Broker
  Enables creation, management, and monitoring of Data Guard configurations that include a primary database and up to 30 standby databases.

- Oracle GoldenGate
  Enables replication updates among multiple databases.

3.16.2 Fast Application Notification

With FAN, the continuous, dynamic database services built into Oracle RAC, Data Guard, and Global Data Services are extended to applications and mid-tier servers. When the state of a database service changes (for example, up, down, or not
restarting), the new status is posted to interested subscribers through FAN events. Oracle drivers and applications use these events to achieve very fast detection of failures, balancing of connection pools following failures, and balancing of connection pools again when the failed components are repaired. For example, when the service at an instance starts, the FAN event is used immediately to route work to that resource. When the service at an instance or node fails, the FAN event is used immediately to interrupt applications to recover.

To solve the high availability problems with database connections, Oracle Clusterware and Data Guard Broker post a FAN event, and also executes server-side callouts, immediately when a service changes state. The event payload contains the relevant information that describes the status of the service on Oracle RAC. On receipt of the FAN event, applications can terminate sessions in communication with the failed instance or node, notify sessions waiting to resume operation, and reorganize in coming work when additional resources are available. To know which sessions to process, every session using Oracle Database has a unique connection signature. The session signatures match the FAN payload.

For planned outages, use any connection pool with FAN configured: OCI, UCP, ICC, WebLogic Server Active Grid Link, or ODP.Net. The FAN planned event drains the work at request boundaries. Immediately, the FAN event is received for a planned down, the idle connections are removed from the pool for that service or instance, and the active (borrowed) connections are marked for release when they are returned to the pool. This effectively drains the work for planned outages with no interruption to the users.

FAN is also used for posting advisories for runtime connection load balancing, Web Affinity, and Data Dependent Routing.

See Also: Oracle Real Application Clusters Administration and Deployment Guide for information about Dynamic Database Services

3.16.1.2.1 Enabling FAN for Oracle Clients

Oracle integrated FAN with many of the common client application environments that are used to connect to Oracle RAC databases. Therefore, the easiest way to use FAN is to use an integrated Oracle client.

Due to the integration with FAN, Oracle integrated clients are more aware of the current status of an Oracle RAC cluster. This prevents client connections from waiting or trying to connect to instances or services that are no longer available. When instances start, Oracle RAC uses FAN to notify the connection pool so that the connection pool can create connections to the recently started instance and take advantage of the additional resources that this instance provides.

Oracle client drivers that are integrated with FAN can:

- Remove terminated connections immediately when a service is declared as DOWN at an instance, and immediately when nodes are declared as DOWN
- Report errors to clients immediately when Oracle Database detects the NOT RESTARTING state, instead of making the client wait while the service repeatedly attempts to restart

Oracle connection pools that are integrated with FAN can:

- Balance connections across all of the Oracle RAC instances when a service starts; this is preferable to directing the sessions that are defined for the connection pool to the first Oracle RAC instance that supports the service
- Balance work requests at run time using load balancing advisory events
3.16.1.2.2 Considerations for Applications That Cannot Use FAN

Configure your operating system for efficient TCP timeouts on the hosts that run the application layer. The OS TCP timeouts should be set to the amount of time it takes for the database layer to failover and the database services to be started. Consult your operating system manuals for how to properly configure TCP timeout.

Configure reconnection logic within the application to respond appropriately in the event of an exception. For example, when a session from the connection pool receives an exception that results in a disconnection (such as an ORA-3113 error), the application should automatically attempt to reconnect that session. The reconnection attempts should be configured such that they will continue for the length of time that it takes to failover the database layer and bring the application services online.

3.16.2 Transaction Failover and Protection

Transaction failover and protection technologies include Transaction Guard and Application Continuity.

3.16.2.1 Transaction Guard

Transaction Guard is a generic tool for applications to provide a reliable, known outcome for transactions following planned and unplanned outages. Applications use a new concept called the logical transaction ID to determine the outcome of the last transaction open in a database session following an outage. Without using Transaction Guard, applications that attempt to retry operations following outages can cause logical corruption by committing duplicate transactions.

Failing to recognize that the last submission has committed, will commit sometime soon, or has not run to completion can lead applications that attempt to replay to cause duplicate transaction submissions and other forms of logical corruption because the software might try to re-issue already persisted changes.

Without Transaction Guard, if a transaction was started and a commit was issued, the commit message that is sent back to the client is not durable. The client is left not knowing whether the transaction committed or not. The transaction cannot be validly resubmitted if the non-transactional state is incorrect or if it is already committed. In the absence of guaranteed commit and completion information, resubmission can lead to transactions applied more than once and in the incorrect state.

See Also:
- Oracle Database 2 Day + Real Application Clusters Guide for information about Transaction Guard with Oracle RAC and Dynamic Database Services
- Oracle Database Development Guide "SQL Processing for Application Developers" and "Using Transaction Guard"
- Oracle Database PL/SQL Packages and Types Reference DBMS_APP_CONT package

3.16.2.2 Application Continuity

A highly available architecture requires the ability of the application tier to transparently fail over to a surviving instance or database advertising the required
service. This ensures that applications are generally available or minimally impacted in the event of node failure, instance failure, data corruption, or database failures. Application Continuity for Java attempts to mask recoverable outages by replaying the request at another available Oracle RAC instance or to another database (such as in the case of a Data Guard role transition).

Application Continuity encompasses:

- FAN: failure notification
- Connection cleanup
- Automatic reconnection and retries of database service residing in another Oracle RAC instance or database
- Replay of the in-flight request

Masking outages of the database session is a complex task for application development and, as a result, errors and timeouts are often exposed to the users. Application Continuity attempts to mask outages from users and applications by recovering the database session following recoverable outages, unplanned and planned. Application Continuity performs this recovery beneath the application so that the outage appears to the application as a delayed execution. For the recovery to succeed, the data and messages restored to the client by Application Continuity must be the same as those that the application has seen and potentially made decisions on.

Application Continuity is started for outages that are recoverable, typically related to underlying software, foreground, hardware, communications, network, or storage layers. Application Continuity is used to improve the user experience when handling both unplanned outages and planned outages.

With Oracle Database 12c release 1, Application Continuity for Java is available for general use with:

- JDBC-Thin Oracle driver
- JDBC Universal Connection Pool
- WebLogic Server Active Grid Link

See Also:

- Oracle Database Concepts for information about transactions
- Oracle Database Development Guide for information about transactions
- Oracle Real Application Clusters Administration and Deployment Guide for information about Dynamic Database Services
- Oracle Database 2 Day + Real Application Clusters Guide for information about Dynamic Database Services

3.16.3 Oracle Database with Global Data Services

Global Data Services enables administrators to automatically and transparently manage client workloads across replicated databases that offer common services. A database service is a named representation of one or more database instances. Services enable you to group database workloads and route a particular work request to an appropriate instance. A global service is a service provided by multiple databases synchronized through data replication.

Global Data Services provides 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 interrelated through Oracle Data Guard, Oracle GoldenGate, or any other replication technology.

The benefits of Global Data Services include the following:

- Enables you to centrally manage global resources, including globally distributed multiple database configurations
- Provides global scalability, availability, and runtime load balancing
- Supports seamless failover
- Enables you to dynamically add databases to the GDS configuration and dynamically migrate global services
- Enables optimal resource utilization

The global services management framework is the software infrastructure for global services. This framework automates and centralizes configuration, maintenance, and monitoring of a GDS configuration, and enables load balancing and failover for services. The framework manages these virtualized resources with minimal administrative overhead, enabling the configuration to handle additional client requests.

The global services management framework is built around the following preexisting Oracle Database technologies:

- Oracle Real Application Clusters (Oracle RAC)
  Enables dynamic load balancing and workload management in a cluster
- Oracle Active Data Guard
  Enables high-performance farms of read-only databases
- Data Guard Broker
  Enables creation, management, and monitoring of Data Guard configurations that include a primary database and up to 30 standby databases
- Oracle GoldenGate
  Enables replication updates among multiple databases
Figure 3–2  Global Data Services Architecture

See Also:  Oracle Database Global Data Services Concepts and Administration Guide
Oracle Database offers an integrated suite of high availability solutions that increase availability and eliminate or minimize both planned and unplanned downtime. These solutions help enterprises maintain business continuity 24 hours a day, 7 days a week. However, the Oracle high availability solutions go beyond reducing downtime by providing solutions to increase system use on the primary and secondary systems and to help improve overall performance, scalability, and manageability.

Table 4–1 describes the various Oracle high availability solutions for unplanned downtime. The table shows how the features discussed in the subsequent sections can be used to address various causes of unplanned downtime. Where several Oracle solutions are listed, the MAA recommended solution is indicated in the Oracle Solution column.

See Table 7–5 for a summary of the attainable recovery times for all of the types of unplanned downtime for each Oracle high availability architecture.

<table>
<thead>
<tr>
<th>Outage Scope</th>
<th>Oracle Solution</th>
<th>Benefits</th>
</tr>
</thead>
</table>
| Site failures | **Oracle Data Guard** (MAA recommended) and Oracle Application Failover solution | ■ Integrated client and application failover  
■ Fastest and simplest database replication  
■ Supports all data types  
■ Zero data loss by eliminating propagation delay  
■ Oracle Active Data Guard supports read-only services and DML on global temporary tables and sequences to off-load more work from the primary |
| | **Oracle GoldenGate** | ■ Flexible logical replication solution (target is open read/write)  
■ Active-active high availability (with conflict resolution)  
■ Heterogeneous platform and heterogeneous database support |
| | **Recovery Manager and Oracle Secure Backup** | ■ Fully managed database recovery and integration with Oracle Secure Backup  
■ Non-real-time recovery |
### Table 4–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

<table>
<thead>
<tr>
<th>Outage Scope</th>
<th>Oracle Solution</th>
<th>Benefits</th>
</tr>
</thead>
</table>
| Instance or computer failures | Oracle Real Application Clusters and Oracle Clusterware (MAA recommended) | ■ Integrated client and application failover  
■ Automatic recovery of failed nodes and instances  
■ Lowest application brownout with Oracle Real Application Clusters |
|                      | Oracle RAC One Node                                                            | ■ Integrated client and application failover  
■ Online database relocation migrates connections and instances to another node  
■ Better database availability than traditional cold failover solutions |
|                      | Oracle Data Guard                                                              | ■ Integrated client and application failover  
■ Fastest and simplest database replication  
■ Supports all data types  
■ Zero data loss by eliminating propagation delay  
■ Oracle Active Data Guard supports read-only services and DML on global temporary tables and sequences to off-load more work from the primary |
|                      | Oracle GoldenGate                                                              | ■ Flexible logical replication solution (target is open read/write)  
■ Active-Active high availability (with conflict resolution)  
■ Heterogeneous platform and heterogeneous database support |
| Storage failures      | Oracle Automatic Storage Management (MAA recommended)                          | ■ Mirroring and online automatic rebalancing places redundant copies of the data in separate failure groups. |
|                      | Oracle Data Guard (MAA recommended)                                            | ■ Integrated client and application failover  
■ Fastest and simplest database replication  
■ Supports all data types  
■ Zero data loss by eliminating propagation delay  
■ Oracle Active Data Guard supports read-only services and DML on global temporary tables and sequences to off-load more work from the primary |
|                      | Recovery Manager with Fast Recovery Area and Oracle Secure Backup (MAA recommended) | ■ Fully managed database recovery and managed disk and tape backups |
|                      | Oracle GoldenGate                                                              | ■ Flexible logical replication solution (target is open read/write)  
■ Active-active high availability (with conflict resolution)  
■ Heterogeneous platform and heterogeneous database support |
| Data corruption       | Corruption Prevention, Detection, and Repair (MAA recommended)                  | ■ Different levels of block corruption prevention and detection at the database level |

| Database initialization settings such as DB_BLOCK_CHECKING, DB_BLOCK_CHECKSUM, and DB_LOST_WRITE_PROTECT |
Data corruption Oracle Data Guard (MAA recommended) Oracle Active Data Guard Automatic Block Repair DB_LOST_WRITE_PROTECT initialization parameter

- In a Data Guard configuration having an Oracle Active Data Guard standby, physical block corruptions detected by Oracle at a primary database are automatically repaired using a good copy of the block retrieved from the standby, and vice versa. The repair is transparent to the user and application.
- If a lost write that occurred on the primary database is detected either by the physical standby database or during media recovery of the primary database, recovery is stopped to preserve the consistency of the database. However, failing over to the standby database using Data Guard will result in some data loss.
- If a lost write is detected on the standby database, you can restore the affected file and restart Redo Apply if the lost write is isolated and the hardware problem is corrected.
- DB_LOST_WRITE_PROTECT initialization parameter provides lost write detection.

Note: Lost writes can corrupt the entire database, which may require that you rebuild the affected database after resolving the hardware issue.

Data Recovery Advisor and Recovery Manager with Fast Recovery Area (MAA recommended)

- Data Recovery Advisor automatically detects data corruptions and recommends the best recovery plan.
- RMAN online block-media recovery time is very fast because RMAN can use flashback logs, backups, or the standby database to recover from physical block corruptions.

Oracle Exadata Database Machine and Oracle Automatic Storage Management (MAA recommended)

- If Oracle ASM detects a corruption and has a good mirror, Oracle ASM returns the good block and repairs the corruption during a subsequent write I/O.
- Exadata Cell provides implicit HARD enabled checks to prevent data corruptions caused by bad or misdirected storage I/O.

Oracle GoldenGate

- Flexible logical replication solution (target is open read/write). Logical replica can be used as a failover target if partner replica is corrupted.
- Active-active high availability (with conflict resolution)
- Heterogeneous platform and heterogeneous database support

<table>
<thead>
<tr>
<th>Outage Scope</th>
<th>Oracle Solution</th>
<th>Benefits</th>
</tr>
</thead>
</table>
| Data corruption | Oracle Data Guard (MAA recommended) Oracle Active Data Guard Automatic Block Repair DB_LOST_WRITE_PROTECT initialization parameter | - In a Data Guard configuration having an Oracle Active Data Guard standby, physical block corruptions detected by Oracle at a primary database are automatically repaired using a good copy of the block retrieved from the standby, and vice versa. The repair is transparent to the user and application.  
- If a lost write that occurred on the primary database is detected either by the physical standby database or during media recovery of the primary database, recovery is stopped to preserve the consistency of the database. However, failing over to the standby database using Data Guard will result in some data loss.  
- If a lost write is detected on the standby database, you can restore the affected file and restart Redo Apply if the lost write is isolated and the hardware problem is corrected.  
- DB_LOST_WRITE_PROTECT initialization parameter provides lost write detection.  

Note: Lost writes can corrupt the entire database, which may require that you rebuild the affected database after resolving the hardware issue. |
| Data Recovery Advisor and Recovery Manager with Fast Recovery Area (MAA recommended) | Data Recovery Advisor automatically detects data corruptions and recommends the best recovery plan.  
- RMAN online block-media recovery time is very fast because RMAN can use flashback logs, backups, or the standby database to recover from physical block corruptions. |
| Oracle Exadata Database Machine and Oracle Automatic Storage Management (MAA recommended) | - If Oracle ASM detects a corruption and has a good mirror, Oracle ASM returns the good block and repairs the corruption during a subsequent write I/O.  
- Exadata Cell provides implicit HARD enabled checks to prevent data corruptions caused by bad or misdirected storage I/O. |
| Oracle GoldenGate | Flexible logical replication solution (target is open read/write). Logical replica can be used as a failover target if partner replica is corrupted.  
- Active-active high availability (with conflict resolution)  
- Heterogeneous platform and heterogeneous database support |
Chapter 5, "Oracle Database High Availability Solutions for Planned Downtime" provides a summary of the key high availability solutions that address different types of planned downtime along with the recovery time for each solution.
Planned downtime can be just as disruptive to operations as unplanned downtime. This is especially true for global enterprises that must support users in multiple time zones, or for those that must provide Internet access to customers 24 hours a day, 7 days a week.

In the past, planned downtime was necessary to perform the following activities:

- Periodic maintenance—such as patching or reconfiguring the system to update a database, application, operating system, middleware, or network
- New deployments—such as to perform major upgrades or new rollouts of the hardware, database, application, operating system, middleware, or network

This chapter contains the following topics:

- High Availability Solutions for Migration
- Dynamic and Online Resource Provisioning
- Online Reorganization and Redefinition
- Oracle High Availability Solutions for System and Software Maintenance
- Online Application Maintenance and Upgrades

5.1 High Availability Solutions for Migration

Table 5–1 describes at a high level the high availability solutions for migration. Each solution is described in the sections following the table.
5.1.1 Platform Migration

Migrating a database to a different platform is required when you move an existing database to a system that runs a different operating system than the current system. For example, database migration is required when moving a database from Microsoft Windows to Linux, or from AIX or HP-UX to Oracle Exadata Database Machine running Oracle Linux. Database migration to a different platform is accomplished with one of the following solutions:

- Data Guard Heterogeneous Physical Standby
- Data Pump full transportable export/import
- Data Pump tablespace transportable export/import

The following features can be used in combination with the migration solutions previously described to reduce database migration downtime:

- Recovery Manager Cross-Platform Inconsistent Tablespace Transportation
- Oracle GoldenGate

5.1.1.1 Migrating a Database to Oracle Exadata Database Machine or SPARC SuperCluster

Database migration to Oracle Exadata Database Machine or SPARC SuperCluster uses the same methods as a database migration across platforms, as described in this section. The target platform for Oracle Exadata and SPARC SuperCluster is described in Table 5–2.
High Availability Solutions for Migration

Table 5–2  Platform Migration to an Engineered System

<table>
<thead>
<tr>
<th>Engineered System</th>
<th>Database Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Exadata Database Machine</td>
<td>Oracle Linux x86 64-bit (little endian) or Oracle Solaris x86-64 (little endian)</td>
</tr>
<tr>
<td>SPARC SuperCluster</td>
<td>Oracle Solaris SPARC (big endian)</td>
</tr>
</tbody>
</table>

See Also:  MAA white paper "Best Practices for Migrating to Exadata Database Machine" at http://www.oracle.com/goto/maa

5.1.1.2 Platform Migration Solutions

Table 5–3 lists the recommended solutions to use for the database migration scenarios. Each solution is described in the sections following the table.

Table 5–3  Database Migration Scenarios and Solutions

<table>
<thead>
<tr>
<th>Database Migration Scenario</th>
<th>Solution to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate to a platform that is the same endian format</td>
<td>1.  Heterogeneous Data Guard Configurations</td>
</tr>
<tr>
<td></td>
<td>2.  Data Pump Full Transportable Export/Import if Data Guard Heterogeneous Physical Standby cannot be used</td>
</tr>
<tr>
<td></td>
<td>3.  Data Pump tablespace transportable export/import if Data Pump full transportable export/import cannot be used</td>
</tr>
<tr>
<td></td>
<td>4.  Oracle Golden Gate if Data Guard Heterogeneous Physical Standby cannot be used, and lower downtime than Data Pump can provide is required</td>
</tr>
<tr>
<td>Migrate to a platform that is a different endian format</td>
<td>1.  Data Pump Full Transportable Export/Import</td>
</tr>
<tr>
<td></td>
<td>2.  Data Pump tablespace transportable export/import if Data Pump full transportable export/import cannot be used</td>
</tr>
<tr>
<td></td>
<td>3.  Oracle Golden Gate if lower downtime than Data Pump can provide is required</td>
</tr>
</tbody>
</table>

5.1.1.2.1  Heterogeneous Data Guard Configurations

Data Guard supports running a physical standby database on a different platform than the primary system for a limited number of platform combinations (for example, Windows to Linux). Migration between platforms that support a heterogeneous primary/standby combination is accomplished with a simple Data Guard switchover operation. The following criteria must be met to use this method:

- The platform combination must be listed as supported in My Oracle Support Note 413484.1.
- The source database and target database must be the same Oracle Database release.

See Also:  My Oracle Support Note 413484.1 at http://support.oracle.com/

5.1.1.2.2  Data Pump Full Transportable Export/Import

You can use the full transportable export/import feature to copy an entire database from one platform to another. You can use Data Pump to produce an export dump file,
transport the dump file and the data files for user-defined tablespaces to the target database if necessary, and then import the export dump file. Full transportable exports are supported from a source database running Oracle Database 11g release 2 (11.2.0.3) or later.

See "Transporting Data" in the Oracle Database Administrator’s Guide for information about the general limitations of transporting data and limitations specific to full transportable export/import.

A full transportable export exports all objects and data necessary to create a complete copy of the database. A mix of data movement methods is used:

- Objects residing in transportable tablespace have only their metadata unloaded into the dump file set; the data itself is moved when you copy the data files to the target database. The data files that must be copied are listed at the end of the log file for the export operation.
- Objects residing in non-transportable tablespaces (for example, SYSTEM and SYSAUX) have both their metadata and data unloaded into the dump file set, using direct path unload and external tables.

The length of time required to migrate a database to a new platform depends on the following factors:

- Data size
- Metadata size

The high-level steps to migrate a database are as follows:

1. Create a new, empty database on the target platform.
2. Stop the application (read-only access to the data is still permitted.)
3. Make the user tablespaces read only in the source database.
4. Perform full transportable export of the source database.
5. Transfer export dump file and data files for user tablespaces to the destination system.
6. Use RMAN to convert the data files to the endian format of the destination system (if necessary).
7. Perform full transportable import into the target database.
8. Make user tablespaces read/write in the target database.
9. Start the application, connecting to the target database.

To reduce migration downtime use Recovery Manager cross-platform inconsistent tablespace transportation in conjunction with Data Pump full transportable export/import.

See Also: See Section 5.1.1.4, "Methods to Reduce Database Migration Downtime” for additional information.

5.1.1.3 Data Pump Tablespace Transportable Export/Import

You can use the tablespace transportable export/import feature to copy all user-defined tablespaces from a database on one platform to a database running on another. A tablespace transportable export exports only the metadata for the tables (and their dependent objects) within a specified set of user-defined tablespaces. The tablespace data files are copied in a separate operation. Then, a transportable tablespace import is performed to import the dump file containing the metadata and
to specify the data files to use. Tablespace transportable exports are supported between different platforms for version 10.0 compatible or later source and target databases.

See “Transporting Data” in Oracle Database Administrator’s Guide for information about the general limitations of transporting data and limitations specific to tablespace transportable export/import.

The length of time required to migrate a database to a new platform depends on the following factors:
- Data size
- Metadata size

The high-level steps are as follows:
1. Create a new, empty database on the target platform.
2. Stop the application (read-only access to the data is still permitted).
3. Import objects required for transport operations into the target database.
4. Make the user tablespaces read only in the source database.
5. Perform full transportable export of the source database.
6. Transfer export dump file and data files for user tablespaces to the destination system.
7. Use RMAN to convert the data files to the endian format of the destination system (if necessary).
8. Perform tablespace transportable import of all user tablespaces.
9. Export and import database objects that could not be transported.
10. Make user tablespaces read/write in the target database.
11. Start the application, connecting to the target database.

To reduce migration downtime use Recovery Manager cross-platform inconsistent tablespace transportation in conjunction with Data Pump tablespace transportable export/import.

See Also: Section 5.1.1.4, "Methods to Reduce Database Migration Downtime"

5.1.1.4 Methods to Reduce Database Migration Downtime

The methods described in the following sections can be used in combination with the migration methods previously described to reduce database migration downtime:
- Recovery Manager Cross-Platform Inconsistent Tablespace Transportation
- Oracle GoldenGate

5.1.1.4.1 Recovery Manager Cross-Platform Inconsistent Tablespace Transportation

The downtime required to migrate a database using Data Pump full or tablespace transportable export/import is primarily determined by the following two factors:
- Data size
- Metadata size
To reduce migration downtime use Recovery Manager cross-platform inconsistent tablespace transportation in conjunction with Data Pump full or tablespace transportable export/import. Migration downtime is reduced by allowing most data to be moved while the source database remains online. When you use Recovery Manager cross-platform inconsistent tablespace transportation in conjunction with Data Pump full or tablespace transportable export/import, the downtime required is primarily determined by the following:

- Data change rate
- Metadata size

RMAN enables you to transport the majority of the database to the target system while the database on the source system remains online by creating an inconsistent backup of the user-defined tablespaces on the source system and restoring it on the target system. Because the time for the initial backup/restore operation can be significant, the data files produced by the inconsistent backup can be rolled forward one or more times using a cross-platform incremental backup. To make the data files consistent in order to complete the transportation, you then apply a final cross-platform incremental backup, taken when the tablespaces are in read-only mode. The final step is to complete the migration using Data Pump full or tablespace transportable export/import.

The high-level steps to transport the database to the target system are as follows:

**Phase 1: Prepare phase**

1. Create an RMAN cross-platform inconsistent backup of all user-defined tablespaces.
2. Restore the cross-platform inconsistent backup on the target system. The target data files created on the target system are called foreign data files.

**Phase 2: Roll forward phase**

1. Create an RMAN cross-platform incremental backup of all user-defined tablespaces.
2. Recover the foreign data files on the target system by applying the cross-platform incremental backup.

   The roll forward phase is repeated as many times as necessary to catch foreign data files up to the source database.

**Phase 3: Transport phase**

1. Stop the application.
2. Make the user-defined tablespaces read-only.
3. Repeat Phase 2 (Roll forward phase) one final time.
4. Migrate the database using Data Pump full transportable export/import, or migrate the user-defined tablespaces using Data Pump tablespace transportable export/import.
5. Start the application, connecting to the target database.

See Also: Oracle Database Backup and Recovery User’s Guide

5.1.1.4.2 Oracle GoldenGate

Use Oracle GoldenGate to reduce migration downtime. Migration downtime is reduced by allowing the target database to be created and kept synchronized while the
source database remains online. When you use Oracle GoldenGate the downtime required is the length of time it takes to reconnect the application to the target database.

The high-level steps are as follows:

1. Start a change-synchronization Extract group to extract ongoing data changes.
2. Create the target database using Data Pump full transportable export/import or Data Pump tablespace transportable export/import.
3. Start the change-synchronization Replicat group to re-synchronize rows that were changed while the target database was being created.
4. Stop the application.
5. Start the application, connecting to the target database.

See Also: Section 3.2, "Oracle GoldenGate"

5.1.2 Database Migration to a Different Character Set

Use Oracle GoldenGate to reduce character set migration downtime. Character set migration downtime is reduced by allowing the target database to be created and kept synchronized while the source database remains online. When you use Oracle GoldenGate the downtime required is the length of time it takes to reconnect the application to the target database.

The high-level steps are as follows:

1. Create the empty target database with the desired character set.
2. Start a change-synchronization Extract group to extract ongoing data changes.
3. Perform a Data Pump full non-transportable export/import. The data is automatically converted to the new character set during the import process.
4. Start the change-synchronization Replicat group to resynchronize rows that were changed while the target database was being created.
5. Stop the application.
6. Start the application, connecting to the target database.

See Also: Section 3.2, "Oracle GoldenGate"

5.1.3 Migrating to Multitenant Architecture

Migrating a non-container database (non-CDB), or an unplugged pluggable database (PDB), to a PDB in a target CDB is accomplished with one of the following solutions:

- CREATE PLUGGABLE DATABASE statement
- Data Pump full transportable export/import
- Data Pump tablespace transportable export/import

<table>
<thead>
<tr>
<th>Table 5-4  Pluggable Database Migration Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solution</strong></td>
</tr>
<tr>
<td>CREATE PLUGGABLE DATABASE statement</td>
</tr>
</tbody>
</table>

See Also: Section 3.2, "Oracle GoldenGate"
5.1.4 Migration to Oracle ASM Storage

5.1.4.1 Migrate to Oracle ASM-Managed Storage Using Data Guard
If a database currently does not use Oracle ASM to manage storage, then you can
migrate all or part of the database into Oracle ASM, thereby simplifying database
administration. Use Data Guard to minimize downtime when migrating to Oracle
ASM. The high-level steps are as follows:

1. Create a standby database using Oracle ASM storage
2. Perform a Data Guard switchover

See Also:
- MAA white paper "Minimal Downtime Migration to ASM" at
  http://www.oracle.com/goto/maa
- "Performing Oracle ASM Data Migration with RMAN" in Oracle
  Automatic Storage Management Administrator’s Guide

5.1.4.2 Migrate to New Storage Using Oracle ASM
If an existing storage device is already managed by Oracle ASM, and it will be
replaced with new storage, and the new storage is connected to the existing database
server or cluster, then use Oracle ASM to perform the storage migration. Oracle ASM
enables you to add all disks from new storage and drop all disks from existing storage.
Oracle ASM automatically rebalances and migrates data to the new storage while
the database remains operational. Before removing the existing storage device, ensure that
the rebalancing is complete.

The high-level steps are as follows:

1. Connect and configure the new storage on the existing system.
2. Add the new storage to Oracle ASM and drop the original storage from Oracle
   ASM using Oracle ASM commands.
3. Wait for the Oracle ASM rebalance operation that moves the data to the new storage to complete.
4. Disconnect the original storage device.

See Also: Oracle Automatic Storage Management Administrator’s Guide

5.1.5 Migrating a Database from a Single-Instance System to an Oracle RAC Cluster

Use Data Guard when migrating from a non-clustered system running single-instance Oracle Database to a clustered environment running Oracle RAC. The time required to perform the switchover is the only downtime incurred.

See Also: MAA white paper "Rapid Oracle RAC Standby Deployment" at http://www.oracle.com/goto/maa

5.2 Dynamic and Online Resource Provisioning

For system and database changes, use the dynamic resource provisioning features that are discussed in the following sections:

- Renaming and Relocating Online Datafiles
- Dynamic Reconfiguration of the Database
- Automatic Tuning of Memory Management
- Automated Distribution of Data Files, Control Files, and Log Files

5.2.1 Renaming and Relocating Online Datafiles

Every data file is either online (available) or offline (unavailable). You can alter the availability of individual data files or temporary files by taking them offline or bringing them online. Offline data files cannot be accessed until they are brought back online.

Starting in Oracle Database 12c Release 1 (12.1) you can use SQL to move an online data file from one physical file to another while the database is open and accessing the file.

You can use the ALTER DATABASE MOVE DATAFILE SQL statement to rename or relocate online datafiles. This statement enables you to rename or relocate a datafile while the database is open and users are accessing the data file.

When you rename or relocate online data files, the pointers to the data files, as recorded in the database control file, are changed. The files are also physically renamed or relocated at the operating system level.

You might rename or relocate online data files because you want to allow users to access the data files when you perform one of the following tasks:

- Move the data files from one type of storage to another
- Move data files that are accessed infrequently to lower cost storage
- Make a tablespace read-only and move its data files to write-once storage
- Move a database into Oracle Automatic Storage Management (Oracle ASM)
- Rename a datafile to a more descriptive name

See Also: Oracle Database Administrator’s Guide to learn how to rename or relocate online data files.
5.2.2 Dynamic Reconfiguration of the Database

Oracle continues to broaden support for dynamic reconfiguration of the database, enabling it to adapt to changes in hardware demands without any service interruptions. Oracle Database dynamically accommodates various changes to hardware and database configurations by providing the ability to:

- Add and remove processors from a symmetric multiprocessing (SMP) server
- Add and remove nodes and instances in an Oracle RAC environment
- Dynamically increase and decrease its shared memory allocation and automatically tune memory online using automatic shared memory management
- Add and remove database disks online without disturbing database activities using Oracle ASM
- Add and remove storage arrays or Exadata Cells online without disturbing database activities using Oracle ASM
- Expand your Exadata system by upgrading to a Quarter Rack, Half Rack, Full Rack, and all the way up to Multiple Full Racks without downtime
- Automatically rebalance the I/O load across the database storage using Oracle ASM
- Move data files online when adding or dropping disks using Oracle ASM, which automatically rebalances database storage whenever the storage configuration is changed
- Dynamically control database session resource consumption using Resource Manager consumer groups and plans
- Change almost all initialization parameters without shutting down the instance, by using either of the following SQL*Plus statements:
  - The ALTER SESSION statement changes the value of a parameter during a session.
  - The ALTER SYSTEM statement changes the value of a parameter in all sessions of an instance for the duration of the instance.

These capabilities provide no-cost system changes and capacity on-demand provisioning, both of which are fundamental requirements of enterprise grid computing.

See Also: Oracle Database Administrator’s Guide for information about platforms that support Automatic Memory Management.

5.2.3 Automatic Tuning of Memory Management

Two memory management initialization parameters, MEMORY_TARGET and MEMORY_MAX_TARGET, enable automatic management of the System Global Area (SGA), Program Global Area (PGA), and other memory required to run Oracle Database.

The MEMORY_MAX_TARGET parameter specifies the maximum value to which the MEMORY_TARGET can grow dynamically.

---

Oracle Database uses a noncentralized policy to free and acquire memory in each subcomponent of the SGA and the PGA. Oracle Database automatically tunes memory by prompting the operating system to transfer granules of memory from less needy to more needy components. The granularity of the memory transfer is dependent on the current free memory and the amount of memory the operating system requires to maintain a basic level of service.

### Note:
Automatic memory management with the `MEMORY_TARGET` and `MEMORY_MAX_TARGET` initialization parameters is supported on Linux, Windows, Solaris, HP-UX, and AIX. See Oracle Database Concepts and Oracle Database Administrator’s Guide for more information about all supported platforms.

<table>
<thead>
<tr>
<th>if ...</th>
<th>And ...</th>
<th>Then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>You omit <code>MEMORY_MAX_TARGET</code></td>
<td>You omit <code>MEMORY_TARGET</code></td>
<td>The initialization parameters are left at their default values (0), and Oracle Database does not automatically tune memory.</td>
</tr>
<tr>
<td>You omit <code>MEMORY_MAX_TARGET</code></td>
<td>Include a value for <code>MEMORY_TARGET</code></td>
<td>The database automatically sets <code>MEMORY_MAX_TARGET</code> to the value of <code>MEMORY_TARGET</code>.</td>
</tr>
<tr>
<td>You omit <code>MEMORY_TARGET</code></td>
<td>Include a value for <code>MEMORY_MAX_TARGET</code></td>
<td>The <code>MEMORY_TARGET</code> parameter defaults to zero.</td>
</tr>
</tbody>
</table>

Oracle Database uses a noncentralized policy to free and acquire memory in each subcomponent of the SGA and the PGA. Oracle Database automatically tunes memory by prompting the operating system to transfer granules of memory from less needy to more needy components. The granularity of the memory transfer is dependent on the current free memory and the amount of memory the operating system requires to maintain a basic level of service.

5.2.4 Automated Distribution of Data Files, Control Files, and Log Files

Oracle ASM automatically distributes data files, control files, and log files across all available disks. Database storage is rebalanced whenever the storage configuration changes, including adding and removing disks, Exadata Cells, or storage arrays. Oracle ASM provides redundancy through the mirroring of database files, and provides optimal performance by automatically striping database files across available disks.

**See Also:** For more information about Oracle ASM:
- Oracle Database Concepts
- Oracle Automatic Storage Management Administrator’s Guide

5.3 Online Reorganization and Redefinition

One way to enhance availability and manageability is to allow user access to the database during a data reorganization operation. The Online Reorganization and Redefinition feature in Oracle Database offers administrators significant flexibility to modify the physical attributes of a table and transform both data and table structure while allowing user access to the database. This capability improves data availability, query performance, response time, and disk space usage. All of these are important in a mission-critical environment and make the application upgrade process easier, safer, and faster.

The Online Reorganization and Redefinition architecture provides the following benefits:
- Online table reorganization and redefinition:
Online Reorganization and Redefinition

- Change any physical attribute of the table online, including moving the table to a new location, partitioning the table, and converting the table from one organization (such as heap-organized) to another (such as index-organized).

- Change many logical attributes such as column names, types, and sizes. Columns can be added, deleted, or merged. However, you cannot modify the primary key of the table.

- REDEF_TABLE procedure, which automates online table reorganization of a single table in one command (new in Oracle Database 12c).

- Set an unused column online (new in Oracle Database 12c).

■ Online index operations:

  - Create indexes online and analyze them simultaneously. You can also use online repair of the physical guess component of logical row IDs (used in secondary indexes and in the mapping table for index-organized tables).

  - Reorganize an index-organized table and secondary indexes online to eliminate the reorganization maintenance window. Secondary indexes support efficient use of block hints (physical guesses). You can also perform online repair of invalid physical guesses of logical row IDs stored in secondary indexes on an index-organized table.

  - Reorganize an index-organized table or table partition without rebuilding its secondary indexes, resulting in a short reorganization maintenance window.

  - New in Oracle Database 12c: drop index online, alter index visible/invisible, alter index unusable online, and drop constraint online.

■ Online moves of partitioned tables

■ Online reorganization support for advanced queues, clustered tables, materialized views, and abstract data types (objects)

■ Fast ADD COLUMN operations with default value (does not need to update all rows to a default value)

■ Speedier application migration and testing with invisible indexes:

  - Speeds up migration with explicit hints, then drops when finished

  - Prevents premature use of newly created indexes

  - Tests effects of DROP INDEX, making the index visible if needed, thus there is no need for an index rebuild

■ Online index builds with no pause to perform DML operations (no exclusive DML locks are required)

■ Easier table DDL operations online (there is an option to wait for active DML operations instead of stopping)

■ Redefinition of multiple partitions in a single redefinition session to reduce the completion time to redefine multiple partitions (new in Oracle Database 12c).

■ Redefinition of tables that have Virtual Private Database (VPD) policies defined on them to eliminate downtime for redefining these tables (new in Oracle Database 12c).

■ Improved SYNC_INTERIM_TABLE performance with optimized Materialized View Log processing (new in Oracle Database 12c).

■ Improved resilience of FINISH_REDEF_TABLE with better lock management (new in Oracle Database 12c).
The ability to modify physical table attributes and transform both data and table structure has been available since the Oracle8i release. Table 5–6 provides a comprehensive list of data reorganization capabilities.
<table>
<thead>
<tr>
<th>Action</th>
<th>Oracle Database 9i</th>
<th>Oracle Database 10g Release 1</th>
<th>Oracle Database 10g Release 2</th>
<th>Oracle Database 11g</th>
<th>Oracle Database 12c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online reorganization using the package</td>
<td>Modify table storage parameters</td>
<td>Reorganize a single partition</td>
<td>Table with materialized view logs or materialized views</td>
<td>Redefinition of multiple partitions in a single redefinition session to reduce the completion time to redefine multiple partitions.</td>
<td>Redefinition of tables that have Oracle Virtual Private Database policies defined on them to eliminate downtime for redefining these tables.</td>
</tr>
<tr>
<td>DBMS_REDEFINITION</td>
<td>Move the table to a different tablespace</td>
<td>Advanced queue and clustered tables</td>
<td>No recompilation of dependent objects when redefinition does not logically affect objects</td>
<td>Drop index online (create/rebuild index online in release 10g and 11g)</td>
<td>Drop index online (create/rebuild index online in release 11g)</td>
</tr>
<tr>
<td></td>
<td>Add support for parallel queries</td>
<td>Table containing an ADT</td>
<td>Alter index invisible / invisible; Alter index unusable online</td>
<td>Alter index visible / invisible; Alter index unusable online</td>
<td>Alter index visible / invisible; Alter index unusable online</td>
</tr>
<tr>
<td></td>
<td>Add or drop partitioning support</td>
<td>Retain and clone statistics</td>
<td>Drop constraint online (create constraint online in release 11g)</td>
<td>Drop constraint online (create constraint online in release 11g)</td>
<td>Drop constraint online (create constraint online in release 11g)</td>
</tr>
<tr>
<td></td>
<td>Re-create the table to avoid fragmentation</td>
<td>Clone check and not null constraints</td>
<td>Set unused column online (add column online in release 11g)</td>
<td>Set unused column online (add column online in release 11g)</td>
<td>Set unused column online (add column online in release 11g)</td>
</tr>
<tr>
<td></td>
<td>Change from a table to an index-organized table, or vice-versa</td>
<td>Copies dependent objects for nested tables</td>
<td>Online move partition (utilizes online redefinition internally)</td>
<td>Online, multi-partition redefinition in single session</td>
<td>Online, multi-partition redefinition in single session</td>
</tr>
<tr>
<td></td>
<td>Add or drop a column</td>
<td></td>
<td>Online redefinition of tables with Oracle Virtual Private Database policies</td>
<td>Online redefinition of tables with Oracle Virtual Private Database policies</td>
<td>Online redefinition of tables with Oracle Virtual Private Database policies</td>
</tr>
<tr>
<td></td>
<td>Transform a column using a function</td>
<td></td>
<td>Single command redefinition with new REDEF_TABLE procedure</td>
<td>Single command redefinition with new REDEF_TABLE procedure</td>
<td>Single command redefinition with new REDEF_TABLE procedure</td>
</tr>
</tbody>
</table>
Oracle High Availability Solutions for System and Software Maintenance

### 5.4 Oracle High Availability Solutions for System and Software Maintenance

Oracle provides high availability solutions to prevent, tolerate, and reduce downtime for all types of planned maintenance. Table 5–7 describes the various Oracle high availability solutions for planned downtime, along with the outage time that can be attained with each solution.

---

Table 5–7 (Cont.) New Data Reorganization Capabilities by Release

<table>
<thead>
<tr>
<th>Action</th>
<th>Oracle Database 9i</th>
<th>Oracle Database 10g Release 1</th>
<th>Oracle Database 10g Release 2</th>
<th>Oracle Database 11g</th>
<th>Oracle Database 12c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reclaiming unused space</td>
<td>Not applicable</td>
<td>Use the SHRINK SPACE clause on the following statements: ALTER TABLE</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER INDEX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER MATERIALIZED VIEW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER MATERIALIZED VIEW LOG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index create online</td>
<td>CREATE INDEX emp.ename.idx ON emp(ename) ONLINE;</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>DML lock-free online index creation, allowing transparent creation with no dependency on workload</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>■ Parallel operations supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Partitions supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ All index types except cluster</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index coalesce online</td>
<td>ALTER INDEX emp.ename_idx COALESCE;</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>■ Parallel operations supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Partitions supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ All index types</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index-organized table move online</td>
<td>ALTER TABLE emp MOVE ONLINE;</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>■ Parallel operations not supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Partitions supported</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Index-Organized table only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also: Oracle Database Administrator’s Guide
In all cases, Oracle recommends that you extensively test all procedures before conducting planned maintenance operations. See Table 7–8 for a summary of the attainable recovery times for all types of planned downtime for each Oracle high availability architecture.

**Table 5–7  Oracle High Availability Solutions for System and Software Maintenance**

<table>
<thead>
<tr>
<th>Maintenance Type</th>
<th>Oracle Recommended Solution</th>
<th>Solution Description</th>
<th>Outage Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system and hardware upgrades</td>
<td>Oracle Real Application Clusters and Oracle Clusterware, Oracle RAC One Node, or Data Guard Standby-First Patch Apply</td>
<td>Section 5.4.1, &quot;Operating System Upgrades and Hardware Upgrades&quot;</td>
<td>No downtime for Oracle RAC and Oracle RAC One Node. Seconds to minutes for Standby-First Patch Apply</td>
</tr>
<tr>
<td>Oracle interim patches or diagnostic patches</td>
<td>Oracle Real Application Clusters and Oracle Clusterware, Oracle RAC One Node, or Online Patching</td>
<td>Section 5.4.2, &quot;Online Patching&quot;</td>
<td>No downtime¹</td>
</tr>
<tr>
<td>Oracle Database and Oracle Grid Infrastructure bundle patches, Patch Set Updates (PSU), Critical Patch Updates (CPU)</td>
<td>Data Guard Standby-First Patch Apply, Oracle Real Application Clusters and Oracle Clusterware, or Oracle RAC One Node</td>
<td>Section 5.4.3, &quot;System and Cluster Upgrades Using Data Guard&quot;</td>
<td>Seconds to minutes with Standby-First Patch Apply. No downtime for Oracle RAC and Oracle RAC One Node</td>
</tr>
<tr>
<td>Oracle Database and Oracle Grid Infrastructure Patch Set (for example, Oracle Database 11.2.0.2 or 11.2.0.3) and Major Upgrade (for example, Oracle Database 11.2 to 12.1)</td>
<td>Oracle Data Guard</td>
<td>Section 5.4.8, &quot;Database Rolling Upgrade with Data Guard&quot;</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Upgrading Exadata storage</td>
<td>The Exadata PatchMgr utility</td>
<td>Section 5.4.7, &quot;Rolling Upgrade of Exadata Storage Server Software&quot;</td>
<td>No downtime</td>
</tr>
<tr>
<td>Application upgrades</td>
<td>Online Application Maintenance and Upgrades</td>
<td>Section 5.5, &quot;Online Application Maintenance and Upgrades&quot;</td>
<td>No downtime</td>
</tr>
</tbody>
</table>

¹ Patches that cannot be applied by performing a rolling upgrade can be applied with the MINIMIZE_DOWNTIME option of the OPatch utility to reduce the availability impact of the patch application.

**See Also:**

- Oracle Data Guard Concepts and Administration for more information about using Data Guard with SQL Apply to upgrade an Oracle database
- The MAA white papers about rolling upgrade best practices at http://www.oracle.com/goto/maa
5.4.1 Operating System Upgrades and Hardware Upgrades

Using Oracle RAC is the recommended solution for avoiding downtime during system and hardware upgrades. For a single-instance Oracle RAC database, you can use Oracle RAC One Node.

If you cannot perform the upgrade using Oracle RAC or Oracle RAC One Node, then the recommended solution is to use Data Guard and physical standby databases as described in Section 5.4.3. Alternatively, you can use cold cluster failover with Oracle Clusterware as described in Section 5.4.5.

The following list provides a high-level overview of the steps when upgrading using Oracle RAC:

1. Perform the following prerequisite checks:
   - Ensure that the planned maintenance can be performed in a rolling manner from an operating system perspective.
   - Ensure that the database and clusterware versions are certified with the new system and hardware changes.

2. Stop the application service if the application service runs on more than one instance in the cluster. If the application service runs on only the instance being upgraded, then relocate the service to another node in the cluster.
   
   Stopping the application service implicitly redirects connections off of the destination instance when using fast application notification (FAN).

3. Shut down the destination instance or instances with the IMMEDIATE option.

4. Shut down and disable Oracle Clusterware.
   
   Disabling Oracle Clusterware prevents it from starting automatically.

5. Perform maintenance.

6. Enable and start Oracle Clusterware.
   
   This step implicitly starts the database instances.

7. Start the application service.
   
   This step implicitly redirects connections to the destination instance when using FAN.

8. Repeat all steps on the next node.

See your operating system-specific Oracle Real Application Clusters installation guide.

5.4.2 Online Patching

Typically, interim and diagnostic patches are applied to one node at a time in a rolling manner. During patch application to a software home, the software (for example, a database instance) running from the home is shut down. If, however, there is an urgent need for the patch to be installed and software cannot be shut down at the current time, then qualified interim and diagnostic patches can be applied online while software remains running.

The only time a patch should be applied in an online manner is when:

- The patch README indicates that it can be applied in an online manner.
- The patch needs to be applied urgently and database instances cannot be shut down to apply the regular (offline) version of the patch.
You can perform online patching with any Oracle database using the OPatch command-line utility.

Use the following considerations when performing online patching:

- Oracle provides qualified interim and diagnostic patches as combination patches, which contain both an online patch and an offline patch for the same bug fix. Therefore, you can apply the online patch initially to avoid unplanned downtime. However, because online patches have memory overhead, you should roll back the online patch, and apply the offline patch during scheduled downtime.

- Applying an online patch increases memory consumption on the system because each Oracle process uses more memory from the Program Global Area (PGA) during the patch application. Consider memory requirements before you begin applying an online patch. Each online patch is unique, and the memory requirements are patch-specific. Apply the patch on your test system first so that you can assess the effect of the online patch on your production system and estimate any additional memory usage.

See Also:

- "RDBMS Online Patching Aka Hot Patching” in My Oracle Support Note 761111.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=761111.1
- Oracle Universal Installer NextGen Installation Guide for information about online patching and the OPatch utility
- Oracle Database Upgrade Guide for an overview of rolling upgrades and rolling patches

5.4.3 System and Cluster Upgrades Using Data Guard

Data Guard and physical standby databases are the recommended solution for performing system and cluster upgrades that you cannot upgrade using Oracle RAC rolling upgrades.

Data Guard is also recommended for migrations to Oracle ASM, Oracle RAC, 64-bit systems, Windows to Linux, or Linux to Windows, or the same processor architecture platforms. For example:

- Use Data Guard for system upgrades that cannot be upgraded using Oracle RAC rolling upgrades due to system restrictions.

- Use Data Guard when migrating to Oracle ASM, from a noncluster environment to Oracle RAC, to a different platform with the same endian format, or to a different platform with the same processor architecture. The time required to perform the switchover is the only downtime incurred. For more information, see "Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration” in My Oracle Support Note 413484.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1

In general, upgrade the physical standby database first and then perform a Data Guard switchover to the physical standby database.

5.4.3.1 Upgrading the Physical Standby Database

To upgrade the physical standby database and perform a switchover:
1. Upgrade the system or change the physical standby database system to your destination environment.

For example, you can convert the standby database from a single-instance database to an Oracle RAC database by using Oracle ASM, without any effect on the primary database. Then, restart the standby database, ensure that it matches your destination environment, and wait for Redo Apply to finish applying all redo data to the standby database.

2. Perform a Data Guard switchover. Optimally, the switchover should take only seconds to minutes.

3. Shut down the original primary database (now the standby database).

4. Upgrade or make system changes to the original primary database.

5. Restart the upgraded database as a standby database and allow recovery to automatically synchronize the databases.

6. Optionally, perform a Data Guard switchover to return the standby database to the primary database role.

---

**Note:** Conversion from 32-bit to 64-bit is automatic if you are applying an Oracle Database patch set or doing an Oracle Database upgrade at the same time. If you are upgrading only the operating system, then you may need to perform the additional post-upgrade steps that are described in the My Oracle Support Note 414043.1 at http://support.oracle.com/. See the Oracle Database Upgrade Guide for more information about upgrades.

---

### 5.4.3.2 Best Practices for System and Cluster Upgrades

Consider the following best practices and guidelines for system and cluster upgrades and migrations:

- For fastest switchover, configure the standby database to use real-time apply and, if possible, ensure there are no archive log gaps and that the databases are close to being synchronized before beginning the switchover operation.

- Use Data Guard and physical standby databases to perform system and cluster upgrades if Oracle RAC rolling upgrade or online patching is not possible. See Oracle Data Guard Concepts and Administration for more information.

### 5.4.4 Rolling Upgrade With Oracle Real Application Clusters

Oracle patches to database software are usually applied to implement known fixes for software problems, or to apply diagnostic patches to gather information about a problem. Plan to apply patches during a scheduled maintenance outage.

There are several types of patches including:

- **Interim patch**
  A single patch created to provide a specific fix between the release of patch sets.

- **Bundle patch**
  A collection of patches that is issued between patch sets. A patch bundle is usually cumulative. Microsoft Windows bug fixes for Oracle Database are generally issued in a patch bundle (as opposed to an interim patch).

- **Patch Set Update (PSU)**
A quarterly patch that contains the most critical fixes for the applicable product (including security fixes), enabling customers to apply one patch to avoid many problems.

- **Critical Patch Update (CPU)**
  A collection of high-priority fixes (usually for security issues) once a quarter. CPUs are cumulative with respect to prior security fixes but may contain other fixes in order to address patch conflicts with non-security patches (that is, reduce the need for merge requests).

- **Diagnostic patch**
  A patch created specifically to diagnose a problem and not to fix a bug.

### 5.4.4.1 Rolling Patch Installation with Oracle Real Application Clusters
To avoid downtime when applying Oracle database patches, perform rolling patch upgrades using Oracle RAC. You can apply approximately 90% of the new patches using Oracle RAC. Oracle provides the capability to perform rolling patch upgrades with Oracle RAC with little or no database downtime using the OPatch command-line utility. If it is not possible to use Oracle RAC, then use Data Guard and physical standby databases. See Section 5.4.3 for more information.

An Oracle RAC rolling upgrade enables all but one of the instances of the Oracle RAC installation to be available during the scheduled outage, further reducing the impact on the application downtime required for scheduled downtime. The Oracle OPatch utility enables you to apply the patch successively to the different instances in an Oracle RAC installation.

Performing a rolling upgrade is possible only for patches that are certified for rolling upgrades.

### 5.4.4.2 Rolling Patch Installation with Data Guard
If it is not possible to use Oracle RAC, then use Data Guard and physical standby databases. Data Guard Standby-First Patch Apply provides support for different software releases between a primary database and its physical standby database for the purpose of applying and validating Oracle patches in rolling manner.

See the README for the patch to determine if a target patch is certified as being a Standby First Patch.

See My Oracle Support Note 1265700.1 for additional information about Oracle Data Guard Standby First Patch Apply.

### 5.4.5 Rolling Upgrade with Oracle Clusterware
Performing rolling upgrades of Oracle Clusterware is the recommended solution for avoiding downtime when upgrading Oracle Clusterware. For single-instance Oracle RAC databases, consider using Oracle RAC One Node.

Rolling upgrades avoid downtime and ensure continuous availability of Oracle Clusterware while the software is upgraded to the new version. When you upgrade to Oracle Clusterware 12c, Oracle Clusterware and Oracle ASM binaries are installed as a single binary called the grid infrastructure. You can upgrade Oracle Clusterware in a rolling manner from Oracle Clusterware 10g and Oracle Clusterware 11g.

You can perform all upgrades to Oracle Clusterware in a rolling manner.
5.4.6 Rolling Upgrade with Oracle Automatic Storage Management

Performing rolling upgrades is the recommended solution for upgrading Oracle ASM. You can perform all upgrades starting with Oracle Database 11g (and later releases) in a rolling manner.

When you upgrade to Oracle Clusterware 12c, Oracle Clusterware and Oracle ASM binaries are installed as a single binary called the grid infrastructure. You can only upgrade Oracle ASM in a rolling manner from Oracle Database 11g release 1 (11.1).

For more information, see the Oracle Automatic Storage Management Administrator’s Guide.

5.4.7 Rolling Upgrade of Exadata Storage Server Software

During a rolling Exadata Storage Server Software upgrade, storage servers are patched one at a time until all of the servers are updated. Rolling patching takes advantage of Oracle ASM redundancy and automatic disk resynchronization to allow databases to continue to operate during patching. Rolling Exadata Storage Server Software upgrade orchestration is managed by the PatchMgr utility provided with the Exadata Storage Server Software.

See Also:

- My Oracle Support Note 888828.1 at http://support.oracle.com/ that includes:
  - The Oracle Exadata Storage Server website at http://www.oracle.com/exadata

5.4.8 Database Rolling Upgrade with Data Guard

Data Guard using SQL Apply is a recommended solution for performing patch set and database upgrades with minimal downtime. If the source database is using data types not natively supported by SQL Apply, you can use Extended Datatype Support (EDS) to accommodate several more advanced data types.

If the source database is using a software version not supported by SQL Apply rolling upgrade (earlier than Oracle Database release 10.1.0.3) or using EDS cannot sufficiently resolve SQL Apply data type conflicts, then consider using Rolling Upgrades using Oracle Active Data Guard, Database Upgrade Assistant (DBUA)\(^2\), transportable tablespace, or Oracle GoldenGate:

- Rolling Upgrades using Oracle Active Data Guard use a Data Guard physical standby database and the SQL Apply process. Section 5.4.8.1 describes the Rolling Upgrades using Oracle Active Data Guard feature.

---

\(^2\) DBUA incurs downtime. The amount of downtime is dependent on a number of factors. See Oracle Database High Availability Best Practices for additional considerations when choosing DBUA as an upgrade option. See Oracle Database Upgrade Guide for instructions on using DBUA to upgrade Oracle Database software.
■ DBUA provides a graphical user interface (GUI) utility that guides you through the upgrade process and is the simplest and recommended method of upgrading a database. However, if the time it takes DBUA to upgrade a database does not fit in the defined maintenance window, then consider using the transportable tablespace feature to perform a database upgrade in less than 1 hour.

■ Transportable tablespace is the solution if you cannot use SQL Apply but the maintenance window requires downtime to be less than 1 hour in duration, and the database being upgraded has a small number of simple schemas and data files that do not need to be transferred as part of the transport process (such as when the data files will be used in place). Section 5.4.8.3 describes the transportable tablespace solution.

■ Oracle GoldenGate provides the most flexibility when performing database upgrades and requiring additional data type support. Section 5.4.8.4 describes this solution.

Do not use Oracle RAC to perform rolling upgrades of patch sets. See your operating system-specific Oracle Real Application Clusters installation guide.

5.4.8.1 Performing Database Upgrades Using Data Guard and Physical Standby Databases

Rolling Upgrades using Oracle Active Data Guard provides new PL/SQL packages that automate much of the process of performing a database rolling upgrade (to a later Oracle Database release or to a new patch set, or when performing other database maintenance) using a physical standby database. You input an upgrade plan and PL/SQL packages automate three phases of the upgrade according to that plan: start, switchover, and finish.

During the upgrade, SQL Apply is used to synchronize the standby across versions, however, when the upgrade is complete, the Data Guard configuration is returned its original state of a primary database and a physical standby database.

Data protection is maintained during the Data Guard database rolling upgrade process by enabling the standby database that is the target of the upgrade to continue receiving primary database redo while the standby database is open in upgrade mode.

If errors are encountered during the process, then you can choose to either correct the errors and resume the upgrade or fall back to the original state of the configuration. This is supported for database rolling upgrades from Oracle Database 12c release 1 (12.1) onward.

The Oracle Database 12c release includes additional native redo-based replication for Data Guard SQL Apply to support database rolling upgrades (transient logical standby). Supported data types include Oracle Securefile, XML, Database File System (DBFS), XDB, Oracle Spatial, Oracle Text and Oracle Multimedia. For a full list of supported data types, see Appendix C in Oracle Data Guard Concepts and Administration.

Data Guard broker also supports database rolling upgrades.

Starting with Oracle Database 12c, Oracle Enterprise Manager Cloud Control (Cloud Control) provides options to perform a rolling upgrade of databases in a Data Guard configuration. The procedures are described in online help within Cloud Control.
5.4.8.2 Performing Database Upgrades Using Data Pump Full Transportable Export/Import

You can use full transportable export/import to upgrade a database from release 11.2.0.3 or later to Oracle Database 12c. To do so, install Oracle Database 12c and create an empty database. Next, use full transportable export/import to transport the release 11.2.0.3 database into the Oracle Database 12c database.

See Section 5.1.1.2.2, "Data Pump Full Transportable Export/Import" for the high-level steps.

See the Oracle Database Administrator’s Guide for information about the general limitations of transporting data and limitations specific to full transportable export/import.

5.4.8.3 Performing Database Upgrades Using Transportable Tablespace

If you cannot use SQL Apply because of data type conflicts, and testing shows that upgrading with DBUA cannot meet uptime requirements, then consider using the transportable tablespace solution to upgrade your database.

To use the transportable tablespace feature to upgrade an Oracle database:

1. Install Oracle Database software on the destination system and perform the initial steps on the source database to prepare for the transport process.

2. Prepare the source and destination databases:
   a. Gather information from the source database.
   b. Create the destination database with Database Configuration Assistant (DBCA).
   c. Prepare the destination database for Oracle Data Pump usage and to accept the tablespaces being transported.

3. Transport the user tablespaces:
   a. Ready the source database for transport by disconnecting users and restricting access to the source database, making all user tablespaces READ ONLY, and capturing sequence starting values from the source database.
   b. Transport the user tablespaces.

4. Verify that the destination database is complete and functional, and then back up the destination database.

Consider the following information when using the transportable tablespace feature:

The transportable tablespace feature is an option for performing a database upgrade in less than 1 hour for databases that have simple schemas and where the data files do not need to be transferred as part of the transport process (such as when the data files will be used in place). See the MAA white paper “Database Upgrade Using Transportable Tablespace” available on the MAA web site at http://www.oracle.com/goto/maa

See Also:

- Appendix C in Oracle Data Guard Concepts and Administration for a full list of supported data types
- Oracle Data Guard Broker
Using the transportable tablespace feature reduces database upgrade time by moving all user tablespaces from a database running an earlier software release to an empty destination database running a current software release. With transportable tablespace, tablespace data files are plugged in to the database by copying the data files to the destination database, then importing the object metadata into the destination database.

5.4.8.4 Performing Database Upgrades Using Oracle GoldenGate

Use Oracle GoldenGate to reduce database upgrade downtime. Database upgrade downtime is reduced by allowing the target database to be upgraded to the new version and kept synchronized while the source database remains online running the current version. When you use Oracle GoldenGate the downtime required is the length of time it takes to reconnect the application to the target database.

The high-level steps are:

1. Start a change-synchronization Extract group to extract ongoing data changes.
2. Create a duplicate target database. The ideal duplicate target database will begin as a physical standby database that is up-to-date.
3. Activate and upgrade the target database to the target version (or perform your maintenance action as described in Table 7–6).
4. Start the change-synchronization Replicat group to resynchronize rows that were changed while the target database was being created and upgraded.
5. Stop the application.
6. Start the application, connecting to the target database.

See Also:

- Oracle GoldenGate documentation for complete information about performing an online database upgrade. See: http://www.oracle.com/technetwork/middleware/goldengate/overview/index.html
- Oracle Database Backup and Recovery User’s Guide to learn about duplicating a database

5.4.8.5 Performing Database Upgrades Using Oracle GoldenGate and Data Guard

The configuration in Figure 5–1 shows how to configure Oracle GoldenGate and Data Guard to minimize downtime and risk for planned outages, such as for any upgrades and migrations that are not supported by a Data Guard database rolling upgrade. For example, this might include migrating to a different hardware architecture and operating system, or performing application upgrades that modify database objects. In this configuration, the physical standby databases provide disaster protection to prevent downtime or data loss before, during, and after the migration. This configuration also avoids any performance impact or operational risk by isolating the production database from any work required to perform the migration.
Oracle GoldenGate replication from the standby database (in the top right of Figure 5–1), to the new production database (bottom right), requires Oracle GoldenGate Archive Log Mode. If the requirements for Archive Log Mode cannot be met, then replicate directly from the original production database (represented by the database in the top left corner).

These requirements are achieved by creating a parallel environment on the new platform. Depending upon the type of migration planned, instantiating the new primary database may be as simple as restoring a backup of the existing standby database. For more complex migrations it may be necessary to use other Oracle technologies to instantiate the new primary database, such as Oracle Transportable Technologies or Oracle Data Pump. After instantiated, any additional changes are then implemented on what will become the new production system. When all of the changes are implemented, a new physical standby database is created to provide continuous data protection after cutover. Oracle GoldenGate heterogeneous replication (previously configured), is then used to synchronize the new production system with all transactions that occurred on the old system while the new environment was being implemented. When synchronization is complete, production is ready for cutover to the new environment. There is also the option of using Oracle GoldenGate heterogeneous replication after the cutover to keep the old environment synchronized with the new production system for a period of time, to provide a fast fall back option if any unanticipated problems arise.
5.5 Online Application Maintenance and Upgrades

For application changes, use the features described in the following list that can significantly reduce (or eliminate) the application downtime required to make changes to an application's database objects:

- Edition-Based Redefinition
- Oracle GoldenGate for Rolling Upgrades
- DDL with the WAIT Option
- ENABLE, DISABLE, and FOLLOWS Clauses for CREATE TRIGGER
- Enhanced ADD COLUMN Functionality
- Finer-Grained Dependencies
- Invisible Indexes
- Invisible Columns
- Multiple Indexes on the Same Set of Columns
- Dependent PL/SQL Recompilation After Online Table Redefinition

5.5.1 Edition-Based Redefinition

Edition-based redefinition enables you to upgrade the database component of an application while the application is in use, thereby minimizing or eliminating downtime. Your changes do not affect users of the application who continue to run the unchanged application until you make the upgraded application available to all users.

In favorable cases, rollover is possible. The preupgrade and the postupgrade editions can be used concurrently so that sessions that were started before the postupgrade edition was published can continue to use the preupgrade edition until they are terminated naturally while new sessions use the postupgrade edition. In less favorable cases, all preupgrade sessions must be terminated before new sessions can be allowed to use the postupgrade edition. In such cases, the application has a small amount of downtime.

The following sections describe the Editions, Editioning Views, and Cross-edition Triggers features of edition-based redefinition. For more information, see Oracle Database Development Guide.

5.5.1.1 Editions

Editions are nonschema objects; as such, they do not have owners. Editions are created in a single namespace, and multiple editions can coexist in the database. The edition feature enables you to copy database objects and redefine the copied objects in isolation.

Editions provide a privacy mechanism for installing new code and for making data changes so that the running production application does not see the changes. When all the required changes are made in private, they are published in a single atomic operation. Cutover depends simply on which edition a session uses.

5.5.1.2 Editioning Views

If you change the structure of one or more tables, you must also use the editioning view feature to insulate application code from changes made to the underlying table during online application upgrade. Tables are not editionable.
Columns are added to the underlying table and a new editioning view is created in the postupgrade edition to expose and to populate them.

Triggers may be created on an editioning view and its columns may be used in SQL hints. The defining subquery of an editioning view may only project or define aliases for selected columns. The SELECT list is used to project a subset of the table's columns and, typically, to rename them. It, therefore, defines a mapping of physical columns to logical columns.

### 5.5.1.3 Cross-edition Triggers
Cross-edition triggers are used as part of edition-based redefinition to keep the data in the preupgrade and postupgrade editions in step with each other. The preupgrade application remains in use concurrently while changes are applied, redefining the preupgrade edition to a postupgrade edition.

If users must be able to change data in the tables while you are changing the table structure, you also use forward cross-edition triggers. If you make the upgraded application available to some users while others continue to use the older version of the application, you also use reverse cross-edition triggers. Cross-edition triggers are not a permanent part of the application because you drop or disable them after you have made the upgraded application available to all users.

### 5.5.2 Oracle GoldenGate for Rolling Upgrades
Consider using Oracle GoldenGate for fast rolling upgrades. However, although Oracle GoldenGate upgrades might incur little or no database downtime, your ability to configure this solution requires some operational investment. See Section 3.2, "Oracle GoldenGate" and the Oracle GoldenGate documentation, as appropriate.

### 5.5.3 DDL with the WAIT Option
Data definition language (DDL) commands require exclusive locks on internal structures. If DDL commands are issued, then these locks may not be available causing the statement to immediately fail even though the DDL might have succeeded less than a second later. Specifying DDL commands with the WAIT option (the new default) resolves this issue. You specify the wait time instance-wide (in the initialization parameter file) and modify the wait time on a session level.

Specifying DDL commands with the WAIT option provides more flexibility to define grace periods for such commands to succeed instead of raising an error right away, thus requiring additional application logic to handle such errors. For more information, see Oracle Database Administrator’s Guide.

### 5.5.4 ENABLE, DISABLE, and FOLLOWS Clauses for CREATE TRIGGER
The states (ENABLE and DISABLE) and ordering (FOLLOWS) are triggers to control the firing of triggers. These additional states allow greater administrative control for triggers. You can use the CREATE TRIGGER statement in a disabled state to validate successful compilation before enabling. In addition, the trigger order can be controlled with the FOLLOWS clause. For more information, see Oracle Database Development Guide.

### 5.5.5 Enhanced ADD COLUMN Functionality
Default values of columns are maintained in the data dictionary for columns specified as NOT NULL.
Adding new columns with `DEFAULT` values and the `NOT NULL` constraint no longer requires the default value to be stored in all existing records. This enhancement not only enables a schema modification in less than a second and works independently of the existing data volume, but it also consumes no space. For more information, see Oracle Database Administrator’s Guide.

### 5.5.6 Finer-Grained Dependencies

Prior to Oracle Database 11g, metadata recorded mutual dependencies between objects with the granularity of the whole object. (For example, PL/SQL unit P depends on PL/SQL unit Q, or view V depends on table T.) In such cases, the dependent objects were sometimes needlessly invalidated. For example, if view V depends only on columns C1, C2, and C3 in table T and a new column, C99, is added, the validity of view V is not logically affected. Nevertheless, in earlier releases, V was invalidated by the addition of column C99.

Beginning with Oracle Database 11g release 1 (11.1), dependency metadata is recorded at a finer level of granularity, so that the addition of C99 does not invalidate view V. Similarly, if procedure P depends only on elements E1 and E2 in package PKG, then if element E99 is added to PKG, procedure P is not invalidated. (In Oracle Database 10g, this change to PKG would invalidate procedure P.)

By reducing the consequential invalidation of dependent objects in response to changes in the objects they depend upon, you can increase application availability. The benefit occurs both in the development environment and when an active application is parsed or upgraded. The benefit occurs when an Oracle Database patch set is applied because changes to schema objects must be compatible. For more information, see Oracle Database Development Guide.

### 5.5.7 Invisible Indexes

An invisible index provides an alternative to making an index unusable or even to dropping the index. An invisible index is maintained for any DML operation but is not used by the optimizer unless you explicitly specify the index with a hint.

Applications often require modification even when the complete application cannot be taken offline. Invisible indexes enable you to use temporary index structures for certain operations or modules of an application without affecting the overall application. Furthermore, you can use invisible indexes to test the removal of an index without dropping it right away, thus enabling a grace period for testing in production environments. For more information, see the Oracle Database Administrator’s Guide.

### 5.5.8 Invisible Columns

An invisible column is a user-specified column whose values are only visible when the column is explicitly specified by name. You can add an invisible column to a table without affecting existing applications, and make the column visible if necessary.

You might use invisible columns if you want to make changes to a table without disrupting applications that use the table. After you add an invisible column to a table, queries and other operations that must access the invisible column must refer to the column explicitly by name. When you migrate the application to account for the invisible columns, you can make the invisible columns visible.

See Oracle Database Administrator’s Guide for more information.
5.5.9 Multiple Indexes on the Same Set of Columns

In Oracle Database 12c, both B-tree and bitmap indexes can be created on the same set of columns. This feature enables an index to be created on the same set of columns as an existing index as long as some characteristic is different. This enables the type of an index to be changed in a patch edition while not disrupting an application. Only one of the multiple indexes can be a visible index at any time.

See “Creating Multiple Indexes on the Same Set of Columns” in Oracle Database Administrator’s Guide for more information.

5.5.10 Dependent PL/SQL Recompilation After Online Table Redefinition

This feature minimizes the need to recompile dependent PL/SQL packages after an online table redefinition. If the redefinition does not logically affect the PL/SQL packages, recompilation is not needed. This optimization is turned on by default.

If recompilation is needed, this feature reduces the time and effort to manually recompile a dependent PL/SQL package after an online table redefinition. The recompilation also includes views, synonyms, and other table-dependent objects (with the exception of triggers) that are not logically affected by the redefinition. For more information about redefining tables online, see Oracle Database Administrator’s Guide.
Operational Prerequisites to Maximizing Availability

Use operational best practices to provide a successful MAA implementation.

This chapter contains the following topics:

- Understand Availability and Performance SLAs
- Establish Test Practices and Environment
- Implement a High Availability Environment and Scalable Architecture
- Set up and Use Security Best Practices
- Establish Change Control Procedures
- Apply Recommended Patches and Software Periodically
- Execute Data Guard Role Transitions
- Establish Escalation Management Procedures
- Configure Monitoring and Service Request Infrastructure for High Availability
- Check the Latest MAA Best Practices

6.1 Understand Availability and Performance SLAs

Understand and document your high availability and performance service-level agreements (SLAs) and create an outage and solution matrix:

- Document the business's cost of downtime, Recovery Time Objectives (RTO or recovery time) and Recovery Point Objectives (RPO or data loss tolerance) for each unplanned outages described in Table 1–1.

- Document your allocated planned maintenance windows per quarter or per year for various planned maintenance activities described in Table 1–2.

- Build an outage and solution matrix similar those shown in Table 7–6, "Attainable Recovery Times for Planned System and Software Maintenance" and Table 7–7, "Data and Application Maintenance".

- Document current and future performance SLAs including overall throughput (for example, transaction rate and redo rate) and response time requirements for each strategic application services. If reduced performance is not allowed at any time, evaluate if sufficient resources are still available after a failure or after a Database role transition operation.
6.2 Implement a High Availability Environment and Scalable Architecture

Once you understand your high availability and performance requirements, then you can design, validate, and configure an architecture using the information in Chapter 7, "High Availability Architectures and Solutions" to meet your high availability requirements. You can test and ensure that you can meet your current and future performance SLAs.

Implement a high availability environment to achieve the optimal high availability architecture:

- Install or update your software with the latest certified patch sets
- Configure your software using best practices
- Document your choices and configuration

6.3 Establish Test Practices and Environment

Validate and automate repair operations to ensure that you meet your target high availability service-level agreements (SLAs). You should validate the backup, restore, and recovery operations and periodically evaluate all repair operations for various types of possible outages (see Table 4-1 and Table 5-7 for more information).

If you use Data Guard for disaster recovery and data protection, Oracle recommends that you perform periodic switchover operations or conduct full application and database failover tests. Also, periodically execute Application and Data Guard switchovers to fully validate all role transition procedures.

A good test environment and proper test practices are essential prerequisites in achieving the highest stability and availability in your production environment. By validating every change in your test environment thoroughly, you can proactively detect, prevent and avoid problems before applying the same change on production.

These practices involve the following:

- Configuring the Test System and QA Environments
- Performing Preproduction Validation Steps

6.3.1 Configuring the Test System and QA Environments

The test system should be a close replica of the production and standby environment. It's recommended to execute functional, performance and availability tests with a workload that mimics production. Evaluate if availability and performance SLAs (as described in Section 6.1) are maintained after each change and ensure that clear fallback or repair procedures are in place if things go awry while applying the change on the production environment.

With a properly configured test system, many problems can be avoided because changes are validated with an equivalent production and standby database configuration containing a full data set and using a workload framework to mimic production (for example, using Oracle Real Application Testing).

Do not try to reduce costs by eliminating the test system because that decision ultimately affects the stability and the availability of your production applications. Using only a subset of system resources for testing and QA has the tradeoffs shown in Table 6-1.
6.3.2 Performing Preproduction Validation Steps

Pre-production validation and testing of hardware, software, database, application or any changes is an important way to maintain stability. The high-level pre-production validation steps are:

1. Review the patch or upgrade documentation or any document relevant to that change. Evaluate the possibility of performing a rolling upgrade if your SLAs require zero or minimal downtime. Evaluate any rolling upgrade opportunities to

<table>
<thead>
<tr>
<th>Test Environment</th>
<th>Benefits and Tradeoffs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Replica of Production and Standby Systems</td>
<td>Validate all patches and software changes. Validate all functional tests. Full performance validation at production scale. Full high availability validation.</td>
</tr>
<tr>
<td>Full Replica of Production Systems</td>
<td>Validate all patches and software changes. Validate all functional tests. Full performance validation at production scale. Full high availability validation minus the standby system. No functional, performance, high availability and disaster recovery validation with standby database.</td>
</tr>
<tr>
<td>Standby System</td>
<td>Validate most patches and software changes. Validate all functional tests. Full performance validation if using Data Guard Snapshot Standby but this can extend recovery time if a failover is required. Role transition validation. Resource management and scheduling is required if standby and test databases exist on the same system.</td>
</tr>
<tr>
<td>Shared System Resource</td>
<td>Validate most patches and software changes. Validate all functional tests. This environment may be suitable for performance testing if enough system resources can be allocated to mimic production. Typically, however, the environment includes a subset of production system resources, compromising performance validation. Resource management and scheduling is required.</td>
</tr>
<tr>
<td>Smaller or Subset of the system resources</td>
<td>Validate all patches and software changes. Validate all functional tests. No performance testing at production scale. Limited full-scale high availability evaluations.</td>
</tr>
<tr>
<td>Different hardware or platform system resources but same operating system</td>
<td>Validate most patches and software changes. Limited firmware patching test. Validate all functional tests unless limited by new hardware features. Limited production scale performance tests. Limited full-scale high availability evaluations.</td>
</tr>
</tbody>
</table>

See Also: Oracle Database Testing Guide
minimize or eliminate planned downtime. Evaluate whether the patch or the change qualifies for Standby-First Patching.

Note: Standby-First Patch enables you to apply a patch initially to a physical standby database while the primary database remains at the previous software release (this applies to certain types of patches and does not apply to Oracle patch sets and major release upgrades; use the Data Guard transient logical standby method for patch sets and major releases). Once you are satisfied with the change, then perform a switchover to the standby database. The fallback is to switchback if required. Alternatively, you can proceed to the following step and apply the change to your production environment. For more information, see "Oracle Patch Assurance - Data Guard Standby-First Patch Apply" in My Oracle Support Note 1265700.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1265700.1

2. Validate the application in a test environment and ensure the change meets or exceeds your functionality, performance, and availability requirements. Automate the procedure and be sure to also document and test a fallback procedure. This requires comparing metrics captured before and after patch application on the test and against metrics captured on the production system. Real Application Testing may be used to capture the workload on the production system and replay it on the test system. AWR and SQL Performance Analyzer may be used to assess performance improvement or regression resulting from the patch.

Validate the new software on a test system that mimics your production environment, and ensure the change meets or exceeds your functionality, performance, and availability requirements. Automate the patch or upgrade procedure and ensure fallback. Being thorough during this step eliminates most critical issues during and after the patch or upgrade.

3. Optionally, use the Oracle Real Application Testing option that enables you to perform real-world testing of Oracle Database. Oracle Real Application Testing captures production workloads and assesses the impact of system changes before production deployment; thus, Oracle Real Application Testing minimizes the risk of instabilities associated with changes.

4. If applicable, perform final pre-production validation of all changes on a Data Guard standby database before applying them to production. Apply the change in a Data Guard environment, if applicable. For more information about Data Guard standby-first patch apply and transient logical standby method, see Section 3.1.2.1, "Data Guard Redo Apply and Standby-First Patching" and Section 3.1.2.2, "Data Guard Transient Logical Rolling Upgrades."

5. Apply the change in your production environment.
6.4 Set up and Use Security Best Practices

Corporate data can be at grave risk if placed on a system or database that does not have proper security measures in place. A well-defined security policy can help protect your systems from unwanted access and protect sensitive corporate information from sabotage. Proper data protection reduces the chance of outages due to security breaches. For more information, see the Oracle Database Security Guide.

6.5 Establish Change Control Procedures

Institute procedures that manage and control changes as a way to maintain the stability of the system and to ensure that no changes are incorporated in the primary database unless they have been rigorously evaluated on your test systems.

Review the changes and get feedback and approval from your change management team, which should include representatives for any component that affects the business requirements, functionality, performance, and availability of your system. For example, the team can include representatives for end-users, applications, databases, networks, and systems.

6.6 Apply Recommended Patches and Software Periodically

By periodically testing and applying the latest recommended patches and software versions, you ensure that your system has the latest security and software fixes required to maintain stability and avoid many known issues. Remember to validate all updates and changes on a test system before performing the upgrade on the production system.

Furthermore, Oracle health check tools such as raccheck (supporting single instance and Oracle RAC systems), odachk (supporting Oracle Data Appliance systems) and exachk (supporting Exadata Database Machine and SPARC Supercluster systems) provide critical software update recommendations.
6.7 Execute Data Guard Role Transitions

When you have a standby database(s), it is important to ensure that the operations and DBA teams are well prepared to use the standby database(s) at anytime when the primary database is down or underperforming, according to a predetermined threshold. By reacting and executing efficiently, which includes detection and making the decision to failover, overall downtime can be reduced from hours to minutes.

If you use Data Guard for high availability, disaster recovery, and data protection, Oracle recommends that you perform periodic switchover operations every three to six months or conduct full application and database failover tests.

See: My Oracle Support provides notes for Data Guard switchovers:

- *Oracle Data Guard Concepts and Administration*, Chapter 9, "Role Transitions"
- *Oracle Data Guard Broker*, Chapter 5, "Switchover and Failover Operations"

6.8 Establish Escalation Management Procedures

Establish escalation management procedures so repair is not hindered. Most repair solutions, when conducted properly are automatic and transparent with the MAA solution. The challenges occur when the primary database or system is not meeting availability or performance SLAs and failover procedures are not automatic as in the case with some Data Guard failover scenarios. Downtime can be prolonged if proper escalation policies are not followed and decisions are not made quickly.

If availability is the top priority, execute repair and failover operations first and then proceed with gathering logs and information for Root Cause Analysis (RCA) after the application service has been reestablished.
6.9 Configure Monitoring and Service Request Infrastructure for High Availability

To maintain your High Availability environment, you should configure the monitoring infrastructure that can detect and react to performance and high availability related thresholds before any downtime has occurred. Also, where available, Oracle can detect failures, dispatch field engineers, and replace failed hardware components such as disks, flash cards, fans, or power supplies without customer involvement.

6.9.1 Execute Database Health Checks Periodically

Oracle database health checks are designed to evaluate your hardware and software configuration and MAA compliance to best practices. All of the Oracle health check tools will evaluate Oracle Grid Infrastructure, Oracle Database, and provide an automated MAA scorecard or review that highlights when key architectural and configuration settings are not enabled for tolerance of failures or fast recovery. For Oracle’s engineered systems such as Exadata Database Machine, there may be hundreds of additional software, fault and configuration checks.

Oracle recommends periodically (for example, monthly for Exadata Database Machine) downloading the latest database health check, executing the health check, and addressing the key FAILURES, WARNINGS, and INFO messages. Use `exachk` for Exadata and SPARC supercluster engineered systems, `odachk` for Oracle Database Appliance, and `racchk` for all other Oracle database deployments including single instance environments.

Furthermore, it is recommended that you execute the health check prior to and after any planned maintenance activity.

See Also:

- Table 4–1, "Outage Types and Oracle High Availability Solutions for Unplanned Downtime"
- Table 5–7, "Oracle High Availability Solutions for System and Software Maintenance"
- For more information about MAA outage and repair, check the MAA web page on the Oracle Technology Network (OTN) at [http://www.oracle.com/goto/maa](http://www.oracle.com/goto/maa)
6.9.2 Configure Oracle Enterprise Manager Monitoring Infrastructure for High Availability

You should configure and use Enterprise Manager and the monitoring infrastructure that detects and reacts to performance and high availability related thresholds to avoid potential downtime. The monitoring infrastructure assists you with monitoring for High Availability and enables you to do the following:

- Monitor system, network, application, database and storage statistics
- Monitor performance and service statistics
- Create performance and high availability thresholds as early warning indicators of system or application problems
- Provide performance and availability advice
- Established alerts and tools and database performance
- Receive alerts for engineered systems hardware faults

See Also:

- "RACcheck - RAC Configuration Audit Tool" in My Oracle Support Note 1268927.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1268927.1
- "Oracle Exadata Database Machine exachk or HealthCheck" in My Oracle Support Note 1070954.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1070954.1
- "ODAchk- Oracle Database Appliance (ODA) Configuration Audit Tool" in My Oracle Support Note 1485630.1 at https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1485630.1

6.9.3 Configure Automatic Service Request Infrastructure

In addition to monitoring infrastructure with Enterprise Manager in the Oracle high availability environment where available, Oracle can detect failures, dispatch field engineers, and replace failing hardware without customer involvement. For example, Oracle Automatic Service Request (ASR) is a secure, scalable, customer-installable software solution available as a feature. The software resolves problems faster by using auto-case generation for Oracle’s Solaris server and storage systems when specific hardware faults occur.
6.10 Check the Latest MAA Best Practices

MAA solutions and best practices continue to be developed and published for Oracle Databases, Exadata Database Machine, Oracle Supercluster, Oracle Fusion Middleware, Oracle Applications, and Oracle Enterprise Manager Grid Control on the Oracle Technology Network (OTN) at

http://www.oracle.com/goto/maa

See Also: See "Oracle Automatic Service Request" in My Oracle Support Note 1185493.1 at

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1185493.1
High Availability Architectures and Solutions

The Maximum Availability Architecture (MAA) is Oracle’s best practices blueprint. It is based on proven and validated Oracle high availability technologies and recommendations. The goal of MAA is provide architectural recommendations based on your business requirements and then optimize and stabilize the architecture and recommended high availability features with configuration and operational best practices.

This chapter describes the various high availability architectures in an Oracle environment and helps you to choose the correct architecture for your organization. It includes the following sections:

- Oracle Database High Availability Architectures
- Choosing the Correct High Availability Architecture
- Integrating Oracle Fusion Middleware High Availability
- Integrating High Availability for All Applications

7.1 Oracle Database High Availability Architectures

The following sections provide an overview of Oracle Database high availability architectures and implement the MAA best practices:

- Oracle Database with Oracle Restart
- Oracle Database with Oracle RAC One Node
- Oracle Database with Oracle Real Application Clusters (Oracle RAC)
- Oracle Database with Data Guard
- Oracle Database with Oracle Restart and Data Guard
- Oracle Database with Oracle RAC One Node and Data Guard
- Oracle Database with Oracle RAC and Data Guard
- Oracle Database with Oracle GoldenGate

See Section 7.2 for a comparison of the different architectures and highlights of the benefits and considerations.

After you have chosen an architecture, then implement it using the operational and configuration best practices described in Chapter 6, "Operational Prerequisites to Maximizing Availability," the MAA white papers, and in Oracle Database High Availability Best Practices. These best practices are required to maximize the benefits of each architecture. See Section 1.5, "Roadmap to Implementing the Maximum
Oracle Database High Availability Architectures

7.1.1 Oracle Database with Oracle Restart

Oracle Database is a single-instance, standalone (noncluster) database and it is the foundation for all high availability architectures. There are numerous high availability features that you can use in the Oracle Database single-instance database architecture.

Oracle recommends that you use the following Oracle features to make a standalone database on a single computer that is highly available for certain failures and planned maintenance activities:

- **Fast-Start Fault Recovery** bounds and optimizes instance and database recovery times.
- **Oracle Restart** enhances the availability of Oracle databases, listeners, and Oracle ASM instances in a single-instance environment by monitoring and automatically restarting Oracle processes. (See "Configuring Automatic Restart of an Oracle Database" in Oracle Database Administrator's Guide for more details.)
- **Oracle Automatic Storage Management** (Oracle ASM) and Oracle Automatic Storage Management Cluster File System (Oracle ACFS) tolerate storage failures and optimize storage performance and usage.
- **Oracle Flashback Technology** optimizes logical failure repair. Oracle recommends that you use automatic undo management with sufficient space to attain your desired undo retention guarantee, enable Oracle Flashback Database, and allocate sufficient space and I/O bandwidth in the fast recovery area.
- **Fast Recovery Area** manages local recovery-related files.
- **Recovery Manager** (RMAN) optimizes local repair of data failures. Oracle recommends that you create and store the local backups in the fast recovery area.
- **Data Recovery Advisor** provides intelligent advice and repair of different data failures
- **Oracle Secure Backup** provides a centralized tape backup management solution
- **Oracle Security Features** prevent unauthorized access and changes.
- **Corruption Prevention, Detection, and Repair** detect and prevent some corruptions and lost writes.
- **Online Reorganization and Redefinition** allows for dynamic data changes.
- **Dynamic and Online Resource Provisioning** allows for dynamic system changes and online file movement.
- **Online Patching** allows for dynamic database patches for diagnostic and interim patches.
- **Online Application Maintenance and Upgrades** with Edition-based redefinition allows an application's database objects to be changed without interrupting the application's availability.
- **Oracle Enterprise Manager** support for patch application simplifies software maintenance.

*Figure 7–1* shows a basic, single-node Oracle Database that includes an Oracle ASM instance. This architecture incorporates several high availability features, including Flashback Database, Online Redefinition, Recovery Manager, and Oracle Secure Backup.
7.1.2 Oracle Database with Oracle RAC One Node

Traditionally, Oracle RAC is used in a multinode architecture, with many separate database instances running on separate servers. Oracle RAC One Node allows you to run one instance of an Oracle RAC database on a single node in a cluster. Thus, this feature allows you to consolidate many databases into a single cluster for easier management, while still providing high availability by quickly relocating instances for planned maintenance or failing over in the event of server failure.

If the node running your Oracle RAC One Node becomes overloaded, you can relocate the instance to another node in the cluster using the online database relocation utility (`srvctl relocate database`), with no downtime for application users.

You can allocate server resources to multiple instances using Oracle Database Resource Manager Instance Caging. Server scalability is unlimited, and if applications grow to require more resources than a single node can supply, you can perform an online upgrade to a traditional multinode Oracle RAC configuration.

The high availability benefits to using Oracle RAC One Node include the following:

---

1. Single-instance databases can use clustered Oracle ASM (Storage GRID) or nonclustered Oracle ASM.
Oracle Database High Availability Overview

- Offers better database availability and faster recovery than traditional cold failover solutions
- Provides better virtualization for databases than hypervisor-based solutions
- Enables online migration of database instances and online patching and upgrading of operating system and database software (incuring no downtime)
- Delivers a comprehensive, less complex, single-vendor solution, with no need to implement third-party products
- Is ready to scale and upgrade to multinode Oracle RAC
- Provides a standardized environment and a common toolset for both single-node and multinode Oracle database deployments
- Is less expensive than cold fail over solutions or a full Oracle RAC deployment
- Fully supports Data Guard. Any database in a Data Guard configuration, whether a primary or standby database, can be an Oracle One Node database
- Fast, automatic, and intelligent connection and service relocation and failover
- Like Oracle RAC mechanism in place to notify client processes about configuration and service level information that includes service status changes, such as UP or DOWN events
- Zero downtime when using the provisioning capability in Oracle Enterprise Manager Grid Control.
- Rolling upgrade for system, clusterware, operating system, CPUs, and some Oracle interim patches.

For virtualization, Oracle RAC One Node with Oracle VM increases the benefit of Oracle VM with the high availability and scalability of Oracle RAC. If your VM is sized too small, you can migrate the Oracle RAC One instance to another larger Oracle VM node in the cluster (using the online database relocation utility) or move the Oracle RAC One instance to another Oracle VM node, and then resize the Oracle VM. When you move the Oracle RAC One Node instance to the newly resized Oracle VM node, you can dynamically increase any limits programmed with Resource Manager Instance Caging.

**Note:** Since Oracle version 11.2 Oracle Clusterware (Cold Cluster Failover) has been replaced by Oracle RAC One Node or Oracle RAC. See Section 7.1.2, "Oracle Database with Oracle RAC One Node" and Section 7.1.3, "Oracle Database with Oracle Real Application Clusters (Oracle RAC)" for more information.

For more information, see the "Administering Oracle RAC One Node" section in the *Oracle Real Application Clusters Administration and Deployment Guide*.

### 7.1.3 Oracle Database with Oracle Real Application Clusters (Oracle RAC)

An architecture that combines Oracle Database with Oracle RAC is inherently a highly available system. Unlike a traditional monolithic database server that is expensive and is not flexible to changing capacity and resource demands, Oracle RAC combines the processing power of multiple interconnected computers to provide system redundancy, scalability, and high availability for planned and unplanned outages.
The clusters that are typical of Oracle RAC environments can provide continuous service for both planned and unplanned outages. Oracle RAC builds higher levels of availability on top of the standard Oracle Database features. All single-instance high availability features, such as the Flashback technologies and online reorganization, also apply to Oracle RAC. Applications scale in an Oracle RAC environment to meet increasing data processing demands without changing the application code. In addition, allowing maintenance operations to occur on a subset of components in the cluster while the application continues to run on the rest of the cluster can reduce planned downtime.

Oracle RAC exploits the redundancy that is provided by clustering to deliver availability with \( n - 1 \) node failures in an \( n \)-node cluster. Unlike the cold cluster model where one node is completely idle, all instances and nodes can be active to scale your application. Communication among the nodes is optimized by means of Redundant Interconnect Usage (without requiring the use of bonding or other technologies) to provide stability, reliability, and scalability.

Oracle Database with Oracle RAC architecture provides the following benefits over a traditional monolithic database server and the cold cluster failover model.

**Oracle RAC Advantages Over Traditional Cold Cluster Solutions**

- Scalability across database instances
- Flexibility to increase processing capacity using commodity hardware without downtime or changes to the application
- Ability to tolerate and quickly recover from computer and instance failures (measured in seconds)
- Application brownout can be zero or seconds compared to minutes and hours with cold cluster solutions
- Optimized communication in the cluster over redundant network interfaces, without using bonding or other technologies

Oracle Grid Infrastructure and Oracle RAC make use of Redundant Interconnect Usage that distributes network traffic and ensures optimal communication in the cluster. This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2). In previous releases, technologies like bonding or trunking were used to make use of redundant networks for the interconnect.

- Rolling upgrades for system and hardware changes
- Rolling patch upgrades for some interim patches, security patches, CPUs, and cluster software
- Fast, automatic, and intelligent connection and service relocation and failover
- Comprehensive manageability integrating database and cluster features with Grid Plug and Play and policy-based cluster and capacity management
- Load balancing advisory and run-time connection load balancing help redirect and balance work across the appropriate resources
- Oracle Quality of Service (QoS) Management for policy-based run-time management of resource allocation to database workloads to ensure service levels are met in order of business need under dynamic conditions. This is accomplished by assigning a service to a server pool where the database is running. Resources from the pool are used to make sure the required capacity is available.
Oracle Enterprise Management support for Oracle ASM and Oracle ACFS, Grid Plug and Play, Cluster Resource Management, Oracle Clusterware and Oracle RAC Provisioning and patching.

■ SCAN (Single Client Access Name) support as a single name to the clients connecting to Oracle RAC that does not change throughout the life of the cluster, even if you add or remove nodes from the cluster.

Figure 7–2 shows Oracle Database with Oracle RAC architecture. This figure shows Oracle Database with Oracle RAC architecture for a partitioned three-node database. An Oracle RAC database is connected to three instances on different nodes. Each instance is associated with a service: HR, Sales, and Call Center. The instances monitor each other by checking “heartbeats.” Oracle Net Services provide client access to the Application/web server tier at the top of the figure.

**Figure 7–2  Oracle Database with Oracle RAC Architecture**

---

**Note:** Since Oracle version 11.2 Oracle Clusterware (Cold Cluster Failover) has been replaced by Oracle RAC One Node or Oracle RAC. See Section 7.1.2, “Oracle Database with Oracle RAC One Node” and Section 7.1.3, “Oracle Database with Oracle Real Application Clusters (Oracle RAC)” for more information.

### 7.1.3.1 Oracle Database with Oracle RAC on Extended Clusters

Oracle Database with Oracle RAC architecture is designed primarily as a scalability and availability solution that resides in a single data center. It is possible, under certain circumstances, to build and deploy an Oracle RAC system where the nodes in the
cluster are separated by greater distances. This architecture is referred to as an extended cluster.

An Oracle RAC extended cluster is an architecture that provides extremely fast recovery from a site failure and allows for all nodes, at all sites, to actively process transactions as part of single database cluster. For example, for a business that has a corporate campus, the extended Oracle RAC configuration could consist of individual Oracle RAC nodes located in separate buildings. Oracle RAC on an extended cluster provides greater availability than a local Oracle RAC cluster, but an extended cluster may not completely fulfill the disaster recovery requirements of your organization.

When the two data centers are located relatively close to each other, extended clusters can provide great protection for some disasters, but not all. You should determine if both sites are likely to be affected by the same disaster. For example, if the extended cluster configuration is set up properly, it can protect against disasters such as a local power outage, an airplane crash, or a flooded server room. However, an extended cluster cannot protect against all data corruptions or specific data failures that impact the database, or against comprehensive disasters such as earthquakes, hurricanes, and regional floods that affect a greater geographical area. (For complete disaster recovery and data protection, use the architecture shown in Figure 7.1.7.)

The advantages to using Oracle RAC on extended clusters include:

- Ability to fully use all system resources without jeopardizing the overall failover times for instance and node failures
- Extremely rapid recovery if one site fails
- All of the Oracle RAC benefits listed in Section 7.1.3

Note: Although an extended cluster architecture can be effective and has been successfully implemented, you should implement it only in the environments (involving the distance, latency, and degree of protection) recommended in this discussion.

Figure 7–3 shows an Oracle RAC extended cluster for a configuration that has multiple active instances on six nodes at two different locations: three nodes at Site A and three at Site B. The public and private interconnects, and the Storage Area Network (SAN) are all on separate dedicated channels, with each one configured redundantly. For availability reasons, the Oracle database is a single database that is mirrored at both of the sites. Also, to prevent a full cluster outage if either site fails, the configuration includes a third voting disk on an inexpensive, low-end standard network file system (NFS) mounted device.
Data Guard is a high availability data protection and disaster-recovery solution that provides very fast automatic failover (referred to as fast-start failover) in database failures, node failures, corruption, and media failures. Furthermore, the standby databases can be used for read-only access and subsequently for reader farms, for reporting, and for testing and development.

Although traditional solutions (such as backup and recovery from tape, storage-based remote mirroring, and database log shipping) can deliver some level of high availability, data protection, and disaster recovery solutions for Oracle databases, Data
Guard provides the most comprehensive high availability and disaster recovery solution for Oracle databases.

**Data Guard Advantages Over Traditional Solutions**

Data Guard provides a number of advantages over traditional solutions, including the following:

- Fast, automatic or automated database failover for data corruptions, lost writes, and database and site failures, with recovery times of potentially seconds with Data Guard as opposed to hours with traditional solutions
- Zero data loss over wide area network using Oracle Active Data Guard Far Sync
- Offload processing for redo transport compression and redo transmission to up to 29 remote destinations using Oracle Active Data Guard Far Sync
- Automatic corruption repair automatically replaces a physical block corruption on the primary or physical standby by copying a good block from a physical standby or primary database
- Most comprehensive protection against data corruptions and lost writes on the primary database
- Reduced downtime for storage, Oracle ASM, Oracle RAC, system migrations and some platform migrations, and changes using Data Guard switchover
- Reduced downtime with Data Guard rolling upgrade capabilities
- Ability to off-load primary database activities—such as backups, queries, or reporting—without sacrificing the RTO and RPO ability to use the standby database as a read-only resource using the real-time query apply lag capability
- Ability to integrate non-database files using Oracle Database File System (DBFS) or Oracle Automatic Storage Management Cluster File System (Oracle ACFS) as part of the full site failover operations (see Section 3.15, "Oracle Replication Technologies for Non-Database Files")
- No need for instance restart, storage remastering, or application reconnections after site failures
- Transparency to applications
- Transparent and integrated support (application continuity and transaction guard) for application failover
- Effective network utilization

For data resident in Oracle databases, Data Guard, with its built-in zero-data-loss capability, is more efficient, less expensive, and better optimized for data protection and disaster recovery than traditional remote mirroring solutions. Data Guard provides a compelling set of technical and business reasons that justify its adoption as the disaster recovery and data protection technology of choice, over traditional remote mirroring solutions.

**Data Guard Advantages Compared to Remote Mirroring Solutions**

The following list summarizes the advantages of using Data Guard compared to using remote mirroring solutions:

- **Better network efficiency**
  
  With Data Guard, only the redo data needs to be sent to the remote site and the redo data can be compressed to provide even greater network efficiency. However, if a remote mirroring solution is used for data protection, typically you
must mirror the database files, the online redo log, the archived redo logs, and the control file. If the fast recovery area is on the source volume that is remotely mirrored, then you must also remotely mirror the flashback logs. Thus, compared to Data Guard, a remote mirroring solution must transmit each change many more times to the remote site.

- **Better performance**

Data Guard only transmits write I/Os to the redo log files of the primary database, whereas remote mirroring solutions must transmit these writes and every write I/O to data files, additional members of online log file groups, archived redo log files, and control files.

Data Guard is designed so that it does not affect the Oracle database writer (DBWR) process that writes to data files, because anything that slows down the DBWR process affects database performance. However, remote mirroring solutions affect DBWR process performance because they subject all DBWR process write I/O’s to network and disk I/O induced delays inherent to synchronous, zero-data-loss configurations.

Compared to mirroring, Data Guard provides better performance and is more efficient, Data Guard always verifies the state of the standby database and validates the data before applying redo data, and Data Guard enables you to use the standby database for updates while it protects the primary database.

- **Better recovery time (RTO)**

With Data Guard and integrated client failover, applications can fail over transparently and be up and running in seconds or minutes. Remote mirroring solutions will need to remaster the storage, recovered the database, and startup the database requiring one or more hours. Integrated client failover is not normally available and needs to be scripted. All these steps add to the RTO and can result in an hour or hours RTO time.

- **Better suited for WANs**

Oracle Active Data Guard Far Sync efficiently extends zero data loss protection to standby databases deployed hundreds or thousands of miles away, isolating the production database from the impact of network latency. Remote mirroring solutions based on storage systems often have a distance limitation due to the underlying communication technology (Fibre Channel or ESCON (Enterprise Systems Connection)) used by the storage systems. In a typical example, the maximum distance between the systems connected in a point-to-point fashion and running synchronously can be only 10 kilometers. By using specialized devices, this distance can be extended to 66 kilometers. However, when the data centers are located more than 66 kilometers apart, you must use a series of repeaters and converters from third-party vendors. These devices convert ESCON or Fibre Channel to the appropriate IP, ATM, or SONET networks.

- **Better resilience and data protection**

Data Guard ensures much better data protection and data resilience than remote mirroring solutions. This is because corruptions introduced on the production database probably can be mirrored by remote mirroring solutions to the standby site, but corruptions are eliminated by Data Guard.

For example, if a stray write occurs to a disk, or there is a corruption in the file system, or the host bus adaptor corrupts a block as it is written to disk, then a remote mirroring solution may propagate this corruption to the disaster-recovery site. Because Data Guard only propagates the redo data in the logs, and the log file consistency is checked before it is applied, all such external corruptions are
eliminated by Data Guard. Automatic block repair may be possible, thus eliminating any downtime in a Data Guard configuration.

- **Higher flexibility**—Data Guard is implemented on pure commodity hardware. It requires only a standard TCP/IP-based network link between the two computers. There is no fancy or expensive hardware required. It also allows the storage to be laid out in a different fashion from the primary computer. For example, you can put the files on different disks, volumes, file systems, and so on.

- **Better functionality**

  Data Guard provides full suite of data protection features that provide a much more comprehensive and effective solution optimized for data protection and disaster recovery than remote mirroring solutions. For example: Oracle Active Data Guard, Redo Apply for physical standby databases, and SQL Apply for logical standby databases, multiple protection modes, push-button automated switchover and failover capabilities, automatic gap detection and resolution, GUI-driven management and monitoring framework, cascaded redo log destinations.

- **Higher ROI**

  Businesses must obtain maximum value from their IT investments, and ensure that no IT infrastructure is sitting idle. Data Guard is designed to allow businesses get something useful out of their expensive investment in a disaster-recovery site. Typically, this is not possible with remote mirroring solutions.

The recommended high availability and disaster-recovery architectures that use Data Guard are described in the following sections:

- **Overview of Single Standby Database Architectures**
- **Overview of Multiple Standby Database Architectures**

### 7.1.4.1 Overview of Single Standby Database Architectures

A single standby database architecture consists of the following key traits and recommendations:

- **Primary database resides in Site A.**

- **Standby database resides in Site B.** You have three options with regard to data protection: 1. synchronous zero data loss if the standby is located close enough to support synchronous transmission without impacting primary database performance to an unacceptable level; 2. asynchronous (near zero data loss) to avoid any impact to primary database performance; 3 Oracle Active Data Guard 12c Far Sync, to enable zero data loss protection using a remote standby database without impacting primary database performance.

- Fast-start failover is recommended to provide automatic failover without user intervention and bounded recovery time. If the primary database uses the asynchronous redo transport, configure your maximum data loss tolerance or the Data Guard broker’s `FastStartFailoverLagLimit` property to meet your business requirements. The observer (thin client watchdog) resides in the application tier and monitors the availability of the primary database. See Oracle Data Guard Broker for a detailed description of the observer.

- Use a physical standby database if read-only access and support for DMLs for Global Temporarily tables and local sequences is sufficient.
- Evaluate logical standby databases if additional indexes are required for reporting purposes and if your application only uses data types supported by logical standby database and SQL Apply.

Figure 7-4 shows the relationships between the primary database, target standby database, and the observer before, during, and after a fast-start failover. The figure shows the same Data Guard configuration in three different frames, as described in the following list:

1. The leftmost frame shows the configuration before fast-start failover occurs. Data Guard is operating in a steady state, with the primary database transmitting redo data to the target standby database and the observer monitoring the state of the entire configuration.

2. The center frame shows the configuration during fast-start failover. Disaster strikes the primary database, and its network connections to both the observer and the target standby database are lost. Upon detecting the break in communication, the observer attempts to reestablish a connection with the primary database for the amount of time defined by the \texttt{FastStartFailoverThreshold} property before initiating a fast-start failover. If the observer cannot regain a connection to the primary database within the specified time, and the target standby database is ready for fast-start failover, then fast-start failover ensues.

3. The rightmost frame shows the configuration after fast-start failover has occurred. The fast-start failover has completed and the target standby database is running in the primary database role. After the former primary database has been repaired, the observer reestablishes its connection to that database and reinstates it as a new standby database. The new primary database starts transmitting redo data to the new standby database.

Figure 7-4 Primary and Standby Databases and the Observer During Fast-Start Failover

The following list describes examples of Data Guard configurations using single standby databases:

Before Fast-Start Failover

Fast-Start Failover Ensues

After Fast-Start Failover

Oracle Instance Primary Database

Oracle Instance Standby Database

Database

Oracle Instance Primary Database

Oracle Instance Standby Database

Database

Oracle Instance Primary Database

Oracle Instance Standby Database

Database

Oracle Instance Primary Database

Oracle Instance Standby Database

Database
A national energy company uses a standby database located in a separate facility 10 miles away from its primary data center. Outages or data loss that could affect customer service and safety are avoided by using Data Guard synchronous transport and automatic failover (fast-start failover).

An infrastructure services provider to the telecommunication industry uses a single standby database located over 400 miles away from the primary database configured for synchronous redo transport, enabling zero-data-loss failover for maximum data protection and high availability.

A telecommunications provider uses asynchronous redo transport to synchronize a primary database on the West Cost of the United States, with a standby database on the East Coast, over 3,000 miles away. This scenario enables the provider to use existing data centers that are geographically isolated, offering a unique level of high availability.

A global manufacturing company used Data Guard to replace storage-based remote mirroring and maintain a standby database at its recovery site 50 miles away from the primary site. Data Guard provides more comprehensive data protection and its more efficient network usage allows plenty of room to grow without the expense of upgrading its network.

7.1.4.2 Overview of Multiple Standby Database Architectures

This architecture is identical to the single-standby database architecture that was described in Section 7.1.4.1, except that there are multiple standby databases in the same Data Guard configuration. While the flexibility offered by Data Guard enables a single standby database to serve multiple-purposes (disaster recovery, high availability, reporting offload, performance testing, database rolling upgrades), it is not unusual for different use-cases to conflict with each other. The following list describes implementations where there is high value in deploying multiple standby databases, each able to serve a different purpose, even though there is a single primary database:

- There are mission critical environments where high availability is required for database failure as well as remote disaster recovery for site failure. For example, should a storage array serving a primary Oracle RAC cluster fail, a surviving middle tier can quickly reconnect and resume processing using a local Data Guard standby database that is deployed near enough to the primary to use Data Guard Maximum Availability and synchronous redo transport. A local standby gives database failover all the characteristics of an HA event - zero data loss, fast failover, no middle tier restart, and high performance due to its closeness to the original primary database. A second Data Guard standby database is deployed at the remote disaster recovery site to provide disaster protection should a single outage be large enough in scope to impact the primary and local standby at the same time. Failover to the remote standby would be reserved for a true disaster event; along with some data loss (asynchronous transport) and some increase in downtime while a new application tier at the remote site becomes operational.

- There are Disaster Recovery environments that require zero data loss failover to a remote site that may be located 1000’s of miles away from the primary site. The data may either be too important to lose (for example, a high-value financial transaction) or it may be that the operational impact of trying to determine which data is lost results in additional downtime (failover is the last resort - downtime is accepted while attempts are made to correct the outage and prevent data loss). Before there was Oracle Database 12c, a multi-standby Oracle Database 11g configuration with a local synchronous standby and remote asynchronous standby database could address this requirement (using a two-ste operation, zero
data loss failover to the local standby and planned switchover to the remote disaster recovery site). An Oracle Active Data Guard 12c configuration using Far Sync makes this even easier and less costly to implement. The primary database ships synchronously to a light-weight far sync instance (control file and archive logs, no data files, no media recovery), the far sync instance forwards to the remote standby. The standard failover commands execute a zero data loss failover to the remote standby just as though it had received changes directly from the primary database.

- Other customers have extreme availability and data protection requirements that must survive multi-site threats. For example, a Data Guard primary may keep standby databases synchronized at two or more remote locations (a minimum of 3 sites - one primary and two standbys) so that production continues to run even if two locations are impacted simultaneously.

- There are application environments characterized by very high read-only workloads: Internet catalog browsing, order/shipment lookup, travel reservations, course information, support information, bank statements, etc. It is easy to imagine many self-service applications that serve information to thousands of online users at the same time. In addition, such applications frequently have a heavy seasonal workload where read-only performance must scale by a factor of 10x or more. Configurations having multiple Oracle Active Data Guard standby databases can scale read performance for such applications - commonly referred to as a reader farm. Read/write transactions are handled by a primary database and Oracle Active Data Guard is used to keep the reader farm synchronized. The primary database usually has a corresponding Data Guard standby that is dedicated target for disaster recovery. An outage at the primary database is transparent to the reader farm (it immediately recognizes the new primary database and continues as if nothing had happened).

- With the advent of Oracle Active Data Guard, an increasing number of customers have been offloading read-only workload to their standby database. At first, this has yielded performance benefits by eliminating conflict between read/write and read-only workloads and utilizing previously idle disaster recovery systems. Over time, however, many customers find that as workload on both primary and Oracle Active Data Guard standby grows, they no longer have sufficient idle capacity to support required performance service levels when a failover requires all workload to run on a single database. It is not unusual for customers to reach this point and deploy a second standby database in the configuration to provide sufficient capacity regardless of the failure state.

- Data Guard Snapshot Standby is used frequently to convert a standby database into a test system. Data protection (RPO) is maintained the entire time a standby operates as a snapshot standby, but recovery time (RTO) can be extended if a failover occurs before the snapshot standby has been converted back to a synchronized physical standby. Customers with stringent RTO requirements will use multiple standby databases - one that is the DR target where production will quickly transition if a failover is required, and a second that is used for performance testing, or for creating dev/test snapshots and clones.

- Database Rolling Upgrades using Data Guard or Oracle Active Data Guard can use the same standby database used for disaster recovery. Alternatively, there are advantages to deploying a second standby database local to the production database to function as the upgrade target. A local upgrade target makes it easier to switchover the application tier once the upgrade is complete. A second standby database dedicated to supporting the rolling upgrade also isolates the remote copy used for DR from being impacted by the upgrade process until production has
moved to the new version. The second standby database can be de-commissioned after all systems have been upgraded.

### 7.1.5 Oracle Database with Oracle Restart and Data Guard

If your business does not require the scalability and additional high availability benefits provided by Oracle RAC, then Oracle Database with Oracle Restart and Data Guard is a good compromise architecture. Oracle Restart is integrated with Data Guard (Data Guard) and the Data Guard Broker (the broker). Configure your Oracle database with the Oracle Restart feature to automatically restart the database, the listener, and other Oracle components after a hardware or software failure or whenever your database host computer restarts.

### 7.1.6 Oracle Database with Oracle RAC One Node and Data Guard

Oracle RAC One Node provides an automated cold failover cluster solution for server and instance HA. Requirements for both HA and disaster recovery are addressed when Oracle RAC One Node is used in combination with Data Guard. Oracle RAC One Node provides additional HA during planned maintenance by keeping applications online while an Oracle Database instance is moved from one server to the next.

For more information see the MAA white paper "Rapid Oracle RAC One Node Standby Deployment" at

http://www.oracle.com/goto/maa

### 7.1.7 Oracle Database with Oracle RAC and Data Guard

You can achieve the highest level of availability when using Oracle RAC and Data Guard. Unlike Oracle RAC One Node, Oracle RAC provides active-active clustering. When a server fails in an Oracle RAC cluster, user connections are quickly transitioned to surviving nodes. In a drastic case where all nodes in the cluster become unavailable (for example, site failure, SAN failure), a Data Guard failover will quickly restore availability by transitioning a synchronized standby database to the primary production role. The combination of Oracle RAC and Data Guard also reduce planned downtime by providing many options for online maintenance, rolling maintenance across Oracle RAC nodes, or database rolling maintenance using the Data Guard standby database. This is the MAA architecture most recommended for business critical applications.

To protect against site failures, the MAA recommends that Oracle RAC and Data Guard reside on separate systems (clusters) and data centers. Figure 7–5 shows the recommended MAA configuration, with Oracle Database, Oracle RAC, and Data Guard.

Architectures with both Oracle RAC and Data Guard reap all of the Oracle RAC benefits described in section Section 7.1.3, "Oracle Database with Oracle Real Application Clusters (Oracle RAC)," and Data Guard benefits described in Section 7.1.4, "Oracle Database with Data Guard." The combination of Oracle RAC with multiple standby databases is also common and provides the additional benefits as described in Section 7.1.4.2, "Overview of Multiple Standby Database Architectures."

Configuring symmetric sites is recommended to ensure that each site can accommodate the performance and scalability requirements of the application after any role transition. Furthermore, operational practices across role transitions are simplified when the sites are symmetric.
7.1.8 Oracle Database with Oracle GoldenGate

Oracle GoldenGate provides a real-time, log-based change data capture and replication software platform. The software provides capture, routing, transformation, and delivery of transactional data across heterogeneous databases in real time. Oracle GoldenGate logical replication enables all databases, both source and target databases, to be open read/write. This fact combined with Oracle GoldenGate’s advanced replication features make it a key component of MAA for addressing a broad range of high availability challenges for zero downtime maintenance, cross platform migration, and continuous data availability using Active-Active bidirectional replication.

The generally recommended MAA best practice is to use Oracle GoldenGate for advanced replication requirements in conjunction with Oracle RAC and Data Guard. Oracle RAC is the most efficient method for achieving server HA and scalability. Data Guard is the most efficient and most optimized solution for disaster protection for Oracle Database. Oracle Golden Gate however, may be used with Oracle Database in place of either Oracle RAC or Data Guard when replication requirements out-weigh the use of Oracle solutions optimized for high availability and disaster recovery. Such replication requirements are characterized by the combination of the following attributes: 1) independent, geographically remote copies of a database that are synchronized in real-time, 2) distribution of read/write workload across all copies, and 3) cost considerations that make it prohibitive to invest in a separate disaster recovery system that is a physical copy of production.
7.2 Choosing the Correct High Availability Architecture

This section summarizes the advantages of the different high availability architectures and provides guidelines for you to choose the correct high availability architecture for your business.

Chapter 2 describes how the high availability requirements for the business plus its allotted budget determine the appropriate architecture. The key factors include:

- Recovery time objective (RTO) and recovery point objective (RPO) for unplanned outages and planned maintenance
- Management overhead (MO)
- Total cost of ownership (TCO) and return on investment (ROI)

For example, Table 7–1 and Table 7–2 provide some insight into the probability of different downtimes during unplanned and planned activities. The data is derived from actual user experiences and from Oracle service requests.

**Table 7–1 Unplanned Outage Frequency**

<table>
<thead>
<tr>
<th>Outage Cause</th>
<th>Outage Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media or disk failure</td>
<td>High</td>
</tr>
<tr>
<td>Application failure</td>
<td>High</td>
</tr>
<tr>
<td>Logical or user failure that manipulate logical data (DMLs and DDLs)</td>
<td>High</td>
</tr>
<tr>
<td>Data corruption and lost writes</td>
<td>Medium</td>
</tr>
<tr>
<td>Server failure</td>
<td>Medium</td>
</tr>
<tr>
<td>Database failure</td>
<td>Low</td>
</tr>
<tr>
<td>Site failure</td>
<td>Low</td>
</tr>
</tbody>
</table>

**Table 7–2 Planned Maintenance Frequency**

<table>
<thead>
<tr>
<th>Maintenance Activity</th>
<th>Maintenance Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application patches</td>
<td>High</td>
</tr>
<tr>
<td>Database and grid infrastructure patches</td>
<td>Medium</td>
</tr>
<tr>
<td>Hardware patches and upgrades</td>
<td>Low</td>
</tr>
<tr>
<td>Operating system patches and upgrades</td>
<td>Low</td>
</tr>
<tr>
<td>Database or application upgrades</td>
<td>Low</td>
</tr>
<tr>
<td>Platform migration</td>
<td>Low</td>
</tr>
<tr>
<td>Character set migration</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 7–3 identifies the key capabilities provided by the architectures that build on Oracle Database and attempts to label each architecture with its greatest strengths.
### Table 7–3  Summary of Key Capabilities Oracle High Availability Architectures

<table>
<thead>
<tr>
<th>Oracle High Availability Architecture</th>
<th>Key Characteristics and Additional Capabilities</th>
</tr>
</thead>
</table>
| **Oracle Database with Oracle Restart** *(Base Architecture)*  
The foundation for all high availability architectures | ■ Fast-Start Fault Recovery bounds and optimizes instance and database recovery times to minutes.  
■ Oracle Restart enhances the availability of Oracle databases, listeners, and Oracle ASM instances in a single-instance environment by monitoring and automatically restarting Oracle processes.  
■ **Oracle Flashback Technology** optimizes logical failure repair.  
■ **Recovery Manager** optimizes local repair of data failures using local backups.  
■ **Fast Recovery Area** manages local recover-related files automatically.  
■ **Oracle Secure Backup** provides a centralized tape backup management solution.  
■ **Oracle Security Features** prevent unauthorized access and changes.  
■ **Data Recovery Advisor** diagnoses persistent (on disk) data failures, presents appropriate repair options, and runs repair operations at your request. Support is for single-instance databases only.  
■ **Corruption Prevention, Detection, and Repair** detect and prevent some corruptions and lost writes.  
■ **Online Reorganization and Redefinition** allows for dynamic data changes.  
■ **Dynamic and Online Resource Provisioning** allows for dynamic system changes.  
■ **Online Patching** allows for dynamic database patching of typical diagnostic patches.  
■ **Online Application Maintenance and Upgrades** with Edition-based redefinition allows an application’s database objects to be changed without interrupting the application’s availability.  
■ **Oracle Enterprise Manager** support for patch application simplifies software maintenance |
Choosing the Correct High Availability Architecture

Oracle Database with Oracle RAC
One Node

- All of the benefits of Oracle Database
- Offers better database availability and faster recovery than traditional cold failover solutions
- Provides better virtualization for databases than hypervisor-based solutions
- Enables online migration of database instances and online patching and upgrading of operating system and database software (incurred no downtime)
- Delivers a comprehensive, less complex, single-vendor solution, with no need to implement third-party products
- Is ready to scale and upgrade to multinode Oracle RAC
- Provides a standardized environment and a common toolset for both single-node and multinode Oracle database deployments
- Is less expensive than cold fail over solutions or a full Oracle RAC deployment
- Fully supports Data Guard. Any database in a Data Guard configuration, whether a primary or standby database, can be an Oracle One Node database
- Fast, automatic, and intelligent connection and service relocation and failover
- Can notify client processes about configuration and service level information that includes service status changes, such as UP or DOWN events
- Zero downtime when using the provisioning capability in Oracle Enterprise Manager Grid Control.
- Rolling upgrade for system, clusterware, operating system, CPUs, and some Oracle interim patches.

Oracle Database with Oracle Real Application Clusters (Oracle RAC)
High availability, scalability, and foundation of server database grids

- All of the benefits of Oracle Database
- Scalability beyond a single system
- Automatic recovery of failed nodes and instances
- Fast application notification (FAN) integrated Oracle client connection pools which allow applications to easily mask failures to the end user.
- FAN with integrated Oracle client failover, including Java applications using UCP with Oracle RAC and Data Guard. Applications can easily mask failures to the end user.
- Run-time performance level management with Oracle Database Quality of Service Management (This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2))
- Zero downtime with Grid Control provisioning
- Rolling upgrade for system, clusterware, operating system, CPUs, PSUs, Bundle Patches, and some Oracle interim patches.

Oracle Database with Oracle RAC on Extended Clusters
Database Grid with site failure protection

- All of the benefits of Oracle RAC
- Protection from site failure

Table 7–3 (Cont.) Summary of Key Capabilities Oracle High Availability Architectures

<table>
<thead>
<tr>
<th>Oracle High Availability Architecture</th>
<th>Key Characteristics and Additional Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database with Oracle RAC One Node</td>
<td>All of the benefits of Oracle Database</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Database with Oracle Real Application Clusters (Oracle RAC) High availability, scalability, and foundation of server database grids</td>
<td>All of the benefits of Oracle Database</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle Database with Oracle RAC on Extended Clusters Database Grid with site failure protection</td>
<td>All of the benefits of Oracle RAC</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Oracle High Availability Architectures

<table>
<thead>
<tr>
<th>Oracle High Availability Architecture</th>
<th>Key Characteristics and Additional Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database with Data Guard</td>
<td>All of the benefits of Oracle Database</td>
</tr>
<tr>
<td>Simplest high availability, data</td>
<td>Automatic and fast failover for computer failure,</td>
</tr>
<tr>
<td>protection, and disaster-recovery</td>
<td>storage failure, data corruption, for configured</td>
</tr>
<tr>
<td>solution</td>
<td>ORA- errors or conditions and database failures</td>
</tr>
<tr>
<td></td>
<td>Protection from site failure</td>
</tr>
<tr>
<td></td>
<td>Rolling upgrade with standby-first patch apply</td>
</tr>
<tr>
<td></td>
<td>for Patch Set Updates, Bundle Patches, Critical</td>
</tr>
<tr>
<td></td>
<td>Patch Updates, operating system, system, cluster-</td>
</tr>
<tr>
<td></td>
<td>ware, and storage changes²</td>
</tr>
<tr>
<td></td>
<td>Database rolling upgrade with transient logical</td>
</tr>
<tr>
<td></td>
<td>standby or SQL apply standby for patchset and</td>
</tr>
<tr>
<td></td>
<td>major database upgrades</td>
</tr>
<tr>
<td></td>
<td>Ability to off-load backups to the standby</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td></td>
<td>Ability to off-load read and reporting workload</td>
</tr>
<tr>
<td></td>
<td>to the standby database</td>
</tr>
<tr>
<td></td>
<td>Only comprehensive lost write protection</td>
</tr>
<tr>
<td></td>
<td>Support for integrated client failover with</td>
</tr>
<tr>
<td></td>
<td>Oracle Net, Fast Application Notification, and</td>
</tr>
<tr>
<td></td>
<td>Global Data Services</td>
</tr>
</tbody>
</table>

| Oracle Database with Oracle Restart  | The sum of benefits of Oracle Clusterware with  |
| and Data Guard                      | Data Guard                                      |
| High availability solution with     |                                                 |
| added data and disaster recovery     |                                                 |
| protection.                          |                                                 |

| Oracle Database with Oracle RAC and | The sum of benefits of Oracle RAC with Data     |
| Data Guard                          | Guard                                          |
| Best high availability, data       |                                                 |
| protection, and disaster-recovery  |                                                 |
| solution with scalability built in |                                                 |

Table 7–3 (Cont.) Summary of Key Capabilities Oracle High Availability Architectures
### Choosing the Correct High Availability Architecture

Oracle Database High Availability Overview

Table 7–4 recommends architectures based on your business requirements for RTO, RPO, MO, scalability, and other factors.

#### Table 7–4  High Availability Architecture Recommendations

<table>
<thead>
<tr>
<th>Consider Using ...</th>
<th>Business or Application Impact ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Database with Oracle Restart</td>
<td>- Maximum RTO for instance or node failure is in minutes.</td>
</tr>
<tr>
<td></td>
<td>- MO is low.</td>
</tr>
<tr>
<td></td>
<td>- ROI is low.</td>
</tr>
<tr>
<td></td>
<td>- RPO is zero.</td>
</tr>
</tbody>
</table>

---

Rolling upgrades with Oracle Clusterware and Oracle RAC incur zero downtime.

Rolling upgrades with Data Guard incur minimal downtime.

The initial investment to build a robust solution is well worth the long-term flexibility and capabilities that Oracle GoldenGate delivers to meet specific business requirements.

---

1. Rolling upgrades with Oracle Clusterware and Oracle RAC incur zero downtime.
2. Rolling upgrades with Data Guard incur minimal downtime.
3. The initial investment to build a robust solution is well worth the long-term flexibility and capabilities that Oracle GoldenGate delivers to meet specific business requirements.
<table>
<thead>
<tr>
<th>Consider Using ...</th>
<th>Business or Application Impact ...</th>
</tr>
</thead>
</table>
| Oracle Database with Oracle RAC One Node | - Maximum RTO for instance or node failure is in seconds.  
- MO is low.  
- ROI is medium.  
- RPO is zero.  
- Online relocation.  
- All Oracle RAC nodes can be active by implementing multiple Oracle RAC One Node configurations for different databases.  
- Zero downtime when using the provisioning capability in Oracle Enterprise Manager Grid Control.  
- Rolling upgrade for system, clusterware, operating system, CPUs, and some Oracle interim patches. |
| Oracle Database with Oracle Real Application Clusters (Oracle RAC) | - RTO for instance or node failure is zero for the database.  
Brief application brownout in some failure scenarios.  
- MO is medium.  
- ROI is high.  
- RPO is zero.  
- Run-time performance level management with Oracle Database Quality of Service Management (This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2)).  
- Zero downtime when using the provisioning capability in Oracle Enterprise Manager Grid Control.  
- Rolling upgrade for system, clusterware, operating system, CPUs, and some Oracle interim patches.  
- Database scalability beyond one instance or node. |
| Oracle Database with Oracle RAC on Extended Clusters | - All of the business benefits of Oracle RAC.  
- MO is high.  
- ROI is medium.  
- RPO is zero.  
- Additional protection from data center failure with special considerations that are documented in Section 7.1.3.1  
- Highest level of availability for server or computer room failure  
- High availability benefits and workload balancing outweigh performance concerns.  
- Willing to make additional provisions for remote data protection to protect against database, data, and cluster failures and corruptions |
Choosing the Correct High Availability Architecture

Table 7–4  (Cont.) High Availability Architecture Recommendations

<table>
<thead>
<tr>
<th>Consider Using ...</th>
<th>Business or Application Impact ...</th>
</tr>
</thead>
</table>
| **Oracle Database with Data Guard** |  ■ Maximum RTO for instance or node failure is in seconds to minutes.  
  ■ Maximum RTO for data corruptions, database, or site failures is in seconds to minutes.  
  ■ Choice of RPO equal to zero (SYNC transport or ASYNC with Far Sync Instance) or near zero (ASYNC).  
  ■ MO is low.  
  ■ ROI is high.  
  ■ Rolling upgrade with standby-first patch apply for Patch Set Updates, Bundle Patches, Critical Patch Updates, operating system, system, clusterware, and storage changes.  
  ■ Database rolling upgrade with transient logical standby or SQL apply standby for patchset and major database upgrades  
  ■ Off-load read-only, reporting, testing and backup activities to the standby database, plus additional support for updates to Global Temp Tables and sequences  
  ■ Limited support for mixed platforms. For more information, see 'Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration' in My Oracle Support Note at [https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1](https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=413484.1)  

For Oracle Active Data Guard, this solution:  
  ■ Supports very high primary database throughput.  
  ■ Provides the simplicity of a physical replica.  
  ■ Provides maximum protection from physical corruptions.  
  ■ Provides read-only access to synchronized standby database and fast incremental backups to off-load production  
  ■ Provides database rolling upgrade with standby-first patch apply and transient logical standby  

For transient logical standby databases or logical standby databases, this solution:  
  ■ Allows for structural changes to the standby database, such as changes to local tables, adding schemas, indexes, and materialized views  
  ■ Off-loads production by providing read-only access to a synchronized standby database and allows read/write access to local tables that are not being modified by the primary database  

| **Oracle Database with Oracle Restart and Data Guard** |  ■ All of the business benefits of Oracle Clusterware (cold cluster failover) and Data Guard  
  ■ MO is low.  
  ■ ROI is medium.  
  ■ RPO is zero for cluster failover, choice of RPO equal to zero for database failover (Data Guard SYNC or ASYNC with Far Sync Instance), or near-zero (Data Guard ASYNC). |
Table 7–5 shows the recovery time (including detection and client failover time) of an integrated Oracle client, whenever relevant. You should adopt the MAA best practices to achieve the optimal recovery time and configuration. Oracle High Availability Best Practice recommendations can be found in *Oracle Database High Availability Best Practices* and in the white papers that can be downloaded from
Choosing the Correct High Availability Architecture

http://www.oracle.com/goto/maa

### Table 7–5. Attainable Recovery Times for Unplanned Outages

<table>
<thead>
<tr>
<th>Outage Type</th>
<th>Oracle Database with Oracle Restart</th>
<th>Oracle RAC One Node</th>
<th>Oracle RAC or Oracle RAC on Extended Clusters</th>
<th>Data Guard</th>
<th>Oracle RAC or Oracle RAC with Data Guard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site failure</td>
<td>Hours to days</td>
<td>Hours to days</td>
<td>Hours to days with Oracle RAC</td>
<td>Seconds to a minute¹</td>
<td>Seconds to a minute¹</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No downtime⁴ if the outage is limited to one</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>building for Oracle RAC on extended clusters</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hours to days if the outage affects both</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>buildings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer failure</td>
<td>Minutes to hours³</td>
<td>Minutes</td>
<td>No downtime⁴</td>
<td>Seconds to a minute</td>
<td>No downtime⁴</td>
</tr>
<tr>
<td>Storage failure</td>
<td>No downtime⁵</td>
<td>No downtime⁵</td>
<td>No downtime⁵</td>
<td>No downtime⁵</td>
<td>No downtime⁵</td>
</tr>
<tr>
<td>Human error</td>
<td>&lt; 30 minutes⁶</td>
<td>&lt; 30 minutes⁶</td>
<td>&lt; 30 minutes⁶</td>
<td>&lt; 30 minutes⁶</td>
<td>&lt; 30 minutes⁶</td>
</tr>
<tr>
<td>Data corruption</td>
<td>Potentially hours⁷</td>
<td>Potentially hours⁷</td>
<td>Potentially hours⁷</td>
<td>Zero downtime⁸</td>
<td>Zero downtime⁸</td>
</tr>
</tbody>
</table>

¹ Recovery time indicated applies to database and existing connection failover. Network connection changes and other site-specific failover activities may lengthen overall recovery time.
² The portion of any application connected to the failed system is temporarily affected. You can configure the failed application connections to fail over to the replica.
³ Recovery time consists largely of the time it takes to restore the failed system.
⁴ Database is still available, but there is a small application black out and additional brownout as the cluster reconfigures and portion of the application connected to the failed system is temporarily affected.
⁵ Storage failures are prevented by using Oracle ASM with mirroring and its automatic rebalance capability.
⁶ Recovery time for human errors depend primarily on detection time. If it takes seconds to detect a malicious DML or DLL transaction, it typically only requires seconds to flash back the appropriate transactions. Longer detection time usually leads to longer recovery time required to repair the appropriate transactions. An exception is undropping a table, which is literally instantaneous regardless of detection time.
⁷ Recovery time depends on block media recovery and the time it takes to restore a consistent block from the flashback logs or database backups, and to recover the block by applying all the redo from archive logs and online redo logs.
⁸ With automatic block repair, this should be the most common block corruption repair. There are some corruptions that cannot be addressed by automatic block repair, and for those we can rely on Data Guard failover that takes seconds to minutes.

Table 7–6 compares the attainable recovery times of each Oracle high availability architecture for many types downtime for planned system and software maintenance.
Choosing the Correct High Availability Architecture

High Availability Architectures and Solutions

Data and application maintenance activities can be performed online with no
downtime. The ability to perform these activities with no downtime is a feature of
Oracle Database and can be achieved with all high availability architectures.

<table>
<thead>
<tr>
<th>Maintenance Action</th>
<th>Oracle Database with Oracle Restart</th>
<th>Oracle RAC</th>
<th>Oracle RAC and Data Guard</th>
<th>Oracle GoldenGate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database server hardware maintenance or upgrade (see Section 5.4.1)</td>
<td>Downtime dependent on maintenance action time</td>
<td>No downtime</td>
<td>Seconds to 5 minutes</td>
<td>No downtime</td>
</tr>
<tr>
<td>Database server operating system upgrade or patch install (see Section 5.4.1)</td>
<td>Downtime dependent on maintenance action time</td>
<td>No downtime</td>
<td>Seconds to 5 minutes</td>
<td>No downtime</td>
</tr>
<tr>
<td>Oracle Database patch set or version upgrade</td>
<td>Minutes to hours</td>
<td>Minutes to hours</td>
<td>Seconds to 5 minutes (see Section 5.4.8)</td>
<td>Seconds to 5 minutes (see Section 5.4.8)</td>
</tr>
<tr>
<td>Oracle Database PSU, CPU, or bundle patch install (see Section 5.4.4)</td>
<td>Minutes to an hour</td>
<td>No downtime</td>
<td>Seconds to 5 minutes</td>
<td>No downtime</td>
</tr>
<tr>
<td>Oracle Database interim or diagnostic patch install (see Section 5.4.4)</td>
<td>Minutes to an hour</td>
<td>No downtime</td>
<td>Seconds to 5 minutes</td>
<td>No downtime</td>
</tr>
<tr>
<td>Oracle Database interim or diagnostic patch install using online patching (see Section 5.4.2)</td>
<td>No downtime</td>
<td>No downtime</td>
<td>No downtime</td>
<td>No downtime</td>
</tr>
<tr>
<td>Rolling Upgrade with Oracle Clusterware (see Section 5.4.5)</td>
<td>N/A</td>
<td>No downtime</td>
<td>N/A</td>
<td>No downtime</td>
</tr>
<tr>
<td>Oracle Exadata Storage Server upgrade or patch install (see Section 5.4.7)</td>
<td>No downtime</td>
<td>No downtime</td>
<td>No downtime</td>
<td>No downtime</td>
</tr>
</tbody>
</table>

Table 7–7 Data and Application Maintenance

<table>
<thead>
<tr>
<th>Maintenance Action</th>
<th>Downtime Attainable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Reconfiguration of the Database</td>
<td>No downtime with Oracle Database</td>
</tr>
<tr>
<td>Renaming and Relocating Online Datafiles</td>
<td>No downtime with Oracle Database</td>
</tr>
<tr>
<td>Automatic Tuning of Memory Management</td>
<td>No downtime with Oracle Database</td>
</tr>
<tr>
<td>Automated Distribution of Data Files, Control Files, and Log Files</td>
<td>No downtime with Oracle Database</td>
</tr>
</tbody>
</table>
Flexible and automated high availability solutions ensure that applications you deploy on Oracle WebLogic Server meet the required availability to achieve your business goals. The solutions introduced in this book are described in detail in the Oracle Fusion Middleware High Availability Guide.

This section contains the following topics:
- Oracle WebLogic Server High Availability Architectures
- Redundant Architectures
- High Availability Services in Oracle Fusion Middleware

See Also:
- Oracle Fusion Middleware Disaster Recovery Guide
- Part VII "Advanced Administration: Backup and Recovery" in Oracle Fusion Middleware Administrator’s Guide

### 7.3.1 Oracle WebLogic Server High Availability Architectures

Oracle WebLogic Server provides high availability and disaster recovery solutions for maximum protection against any kind of failure with flexible installation, deployment, and security options. These solutions are categorized into local high availability...
solutions that provide high availability in a single data center deployment, and disaster-recovery solutions, which are usually geographically distributed deployments that protect your applications from disasters such as floods or regional network outages.

At a high level, Oracle WebLogic Server local high availability architectures include several active-active and active-passive architectures. Although both types of solutions provide high availability, active-active solutions generally offer higher scalability and faster failover, although they tend to be more expensive. With either the active-active or the active-passive category, multiple solutions exist that differ in ease of installation, cost, scalability, and security.

See Also:
- *Oracle Fusion Middleware Disaster Recovery Guide*
- MAA Best Practices - Oracle Fusion Middleware

### 7.3.2 Redundant Architectures

Oracle WebLogic Server provides redundancy by offering support for multiple instances supporting the same workload. These redundant configurations provide increased availability either through a distributed workload, through a failover setup, or both.

From the entry point to an Oracle WebLogic Server system (content cache) to the back-end layer (data sources), all the tiers that are crossed by a request can be configured in a redundant manner with Oracle WebLogic Server. The configuration can be an active-active configuration using Oracle WebLogic Server Cluster or an active-passive configuration using Oracle WebLogic Server Cold Cluster Failover.

### 7.3.3 High Availability Services in Oracle Fusion Middleware

The *Oracle Fusion Middleware High Availability Guide* describes the following high availability services in Oracle Fusion Middleware in detail:

- **Process death detection and automatic restart**
  Oracle WebLogic Server Node Manager monitors the Managed Servers. If a Managed Server goes down, Node Manager tries to restart it for a configured number of times.

- **Clustering**
  Oracle Fusion Middleware uses WebLogic clustering capabilities, such as redundancy, failover, session state replication, cluster-wide JNDI services, Whole Server Migration, and cluster wide configuration.

- **State replication and routing**
  Oracle WebLogic Server can be configured for replicating the state of stateful applications.

- **Load balancing and failover**
  Oracle Fusion Middleware has a comprehensive feature set around load balancing and failover to leverage availability and scalability of Oracle RAC databases. All Oracle Fusion Middleware components have built-in protection against loss of service, data or transactions as a result of Oracle RAC instance unavailability due to planned or unplanned downtime.

- **Server migration**
Oracle Fusion Middleware components leverage WebLogic Server capabilities to provide failover and automatic restart on a different cluster member.

- **Rolling patching**
  Oracle WebLogic Server allows for rolling patching where a minor maintenance patch can be applied to the product binaries in a rolling fashion without having to shut down the entire cluster.

- **Configuration management**
  Most of the Oracle Fusion Middleware component configuration can be done at the cluster level. Oracle Fusion Middleware uses WebLogic Server’s cluster wide-configuration capabilities for server configuration, such as data sources, EJBs, and JMS, as well as component application artifacts, and ADF and WebCenter custom applications.

- **Backup and recovery**
  Oracle Fusion Middleware backup and recovery is a simple solution based on file system copy for Middle-tier components.

  **See Also:** *Oracle Fusion Middleware High Availability Guide*

### 7.4 Integrating High Availability for All Applications

A highly available and resilient application requires that every component of the application must tolerate failures and changes. A highly available application must analyze every component that affects the application, including the network topology, application server, application flow and design, systems, and the database configuration and architecture. This book focuses primarily on the database high availability solutions.

See the high availability solutions and recommendations for Oracle Fusion Middleware, Oracle Fusion Applications, Oracle Enterprise Manager, and Oracle Applications Unlimited on the MAA website at:

http://www.oracle.com/goto/maa
Oracle offers three engineered systems designed specifically to run Oracle Database in a high availability environment:

- **Oracle Exadata Database Machine** - the highest performance and most available platform for running the Oracle Database
- **Oracle SPARC SuperCluster** - the best multi-purpose engineering system for Oracle Database and applications
- **Oracle Database Appliance** - the simplest, high availability Oracle Database appliance

### 8.1 Oracle Exadata Database Machine

The Oracle Exadata Database Machine is an engineered system, complete with preoptimized and preconfigured software, servers, storage, configured to current best practices, that provides an optimal solution for all database workloads, ranging from scan-intensive data warehouse applications to highly concurrent OLTP applications. It combines Oracle Exadata Storage Server Software, Oracle Database software, and hardware components to deliver extreme performance in a highly available and highly secure environment. Along with Oracle's unique clustering and workload management capabilities, the Database Machine is also well-suited for consolidating multiple databases onto a single grid.

Exadata Database Machine is designed for high performance, scalability and availability for OLTP, Data Warehouse applications, database consolidation, and cloud computing. It is the only Engineered System focused on Oracle Database functionality and fully optimized for all database workloads using Exadata Smart Flash Technology, Exadata I/O Resource management, Exadata smart offloading capabilities and features. For the best combination of database performance, scalability and availability, use the Exadata MAA architecture.

The Oracle Exadata Database Machine hardware is fully redundant without any single points of failure. The Oracle software used on the Oracle Exadata Database Machine, used in conjunction with MAA best practices, yields a fault-tolerant system with the following benefits:

- Continuous database availability across node and instance failures through the use of Oracle RAC
- Data protection and continuous database accessibility across disk and cell failures through the use of Oracle ASM and the Oracle Exadata Storage Server Software
- Prevents and automatically repairs data corruption using the Oracle ASM automatic repair mechanism, the built-in corruption checks within the Exadata
storage, and the Oracle Database block corruption prevention software/parameters

- Provides redundant and fault tolerant networking, cabling and server interconnectivity
- Provides the ability to quickly reestablish database service if the Oracle Exadata Database Machine—or data center the machine resides in—is damaged, through the use of Data Guard and another Oracle Exadata Database Machine

For planned maintenance, Exadata Database Machine provides the following benefits:

- Supports Oracle ASM, Oracle Clusterware, and Oracle RAC rolling upgrade and software changes
- Supports Oracle Exadata Storage Server Software rolling upgrades for patches
- Allows application and system changes with Data Guard and Oracle GoldenGate
- Supports all of the online maintenance capabilities that are available in the Oracle Database

The Oracle Exadata Database Machine deployment takes between three to five days from hardware arrival, and the Oracle OneCommand utility is used to configure the software, creating an MAA compliant configuration. The OneCommand configuration includes:

- Installing the chosen operating system (Oracle Linux or Oracle Solaris) for the database servers
- Installing and configuring the Grid Infrastructure (ASM and clusterware), Database and Exadata software according to MAA best practices
- Configuring ASM disk groups accordingly to customer requirements. To ensure redundancy during a rolling upgrade of the Exadata Storage Server Software, MAA recommends ASM high redundancy disk groups.
- Validating the HA bonded InfiniBand network for database and storage operations and optionally configuring and configure redundant Ethernet networks for client and database backup to MAA recommendations.
- Installing and setting up the recommended monitoring framework including SNMP alerting, Oracle Configuration Manager, Automatic Service Requests, Enterprise Manager Grid Control Agents.
- Configuring Oracle Real Application Clusters and Oracle Database according to MAA and performance best practices resulting in a production ready database that is migration ready and requiring minimum database initialization changes. DBCA sample templates are available for OLTP or Data Warehouse applications. For consolidating multiple databases onto Exadata, please refer to the MAA white paper, "Best Practices For Database Consolidation On Oracle Exadata Database Machine" at
  
  http://www.oracle.com/goto/maa

With Exadata Database Machine, your Oracle Database High Availability architecture choices are simplified compared to Table 7–4, "High Availability Architecture Recommendations". Table 8–1 shows Oracle’s recommended high availability architecture choices:
Table 8–1  Recommended High Availability Architectures for Exadata Database Machine

<table>
<thead>
<tr>
<th>Consider Using</th>
<th>Business or Application Impact</th>
</tr>
</thead>
</table>
| Primary Exadata Database Machine | - All benefits of Real Application Cluster, ASM and Exadata Database Machine  
- Maximum RTO for instance or node failure is zero for the database  
- Maximum RTO for single storage failure is zero  
- MO is medium if customer follows exachk recommendations and uses Oracle’s free Platinum Services (refer to http://www.oracle.com/us/support/premier/engineered-systems-solutions/platinum-services/overview/index.htm)  
- ROI is high  
- RPO is zero  
- Run-time performance level management with Oracle Database Quality of Service Management (This functionality is available starting with Oracle Database 11g Release 2 (11.2.0.2)).  
- Zero downtime when using the provisioning capability in Oracle Enterprise Manager Grid Control.  
- Rolling upgrade for system, Exadata Bundle Patches, clusterware, operating system, CPUs, Exadata software changes, Switch upgrades, and some Oracle interim patches.  
- Database, Network, and Storage scalability. |
| Primary Exadata Database Machine and Standby Exadata Database Machine | - All benefits of Real Application Cluster, ASM, Data Guard and Exadata Database Machine  
- MO is medium if customer follows exachk recommendations and uses Oracle’s free Platinum Services (refer to http://www.oracle.com/us/support/premier/engineered-systems-solutions/platinum-services/overview/index.htm).  
- ROI is high  
- Choice of RPO equal to zero for database failover (Data Guard SYNC or FAR SYNC with no production performance impact), or near-zero (Data Guard ASYNC). |
| Primary Exadata Database Machine, Standby Exadata Database Machine and Oracle GoldenGate | - All benefits of Real Application Cluster, ASM, Data Guard, GoldenGate and Exadata Database Machine  
- MO is medium if customer follows exachk recommendations and uses Oracle’s free Platinum Services (refer to http://www.oracle.com/us/support/premier/engineered-systems-solutions/platinum-services/overview/index.htm). Much more flexibility with Oracle GoldenGate to leverage logical replica for scalability, availability and planned maintenance; however management investment is higher.  
- ROI is high  
- RPO is zero for cluster failover, choice of RPO equal to zero for database failover (Data Guard SYNC or FAR SYNC with no production performance impact), or near-zero (Data Guard ASYNC). |
The recommended Exadata MAA Architecture consists of three elements: 1) a production Exadata system (primary), 2) a standby Exadata system, and 3) an Exadata test or development system.

The second element can be an Active Standby Exadata system that is a replica of the primary that contains all the benefits of any Exadata Database Machine and the benefits described in Section 3.1, "Oracle Data Guard."

The third element can be a development/test Exadata system that is independent of the primary and standby Exadata systems, following the best practices described in Section 6.3, "Establish Test Practices and Environment."

It is recommended that you run Oracle's Exadata Health Check (exachk) monthly because it provides the most comprehensive configuration checks for Exadata software, network, and hardware components, and it reports any variance from MAA best practices. Other post-deployment and operational best practices specific to Exadata are described in Chapter 6, "Operational Prerequisites to Maximizing Availability," and the MAA white paper "MAA Best Practices for Oracle Exadata Database Machine." See My Oracle Support Note 1070954.1 for information about exachk.

The MAA configuration best practices will continue to be integrated and incorporated during the initial installation and deployment of the Exadata Database Machine and Exadata Cell.

See Also:

- OTN Exadata MAA website:
- MAA Best Practices for Oracle Exadata Database Machine (technical white paper)
- Oracle Exadata Database Machine Owner's Guide
- Oracle Exadata Storage Server Software User's Guide
- Oracle Exadata Database Machine on Oracle Technology Network at:

8.2 Oracle SPARC SuperCluster

The Oracle SPARC SuperCluster T4-4 is a multi-purpose engineered system for consolidating a wide range of mission critical applications and rapidly deploying cloud services. The SPARC SuperCluster T4-4 utilizes high performance technologies from Oracle Exadata Storage Servers and Oracle Exalogic Elastic Cloud combined with SPARC T4-4 servers, ZFS Storage Appliance, InfiniBand technology, Oracle Solaris 11, and the unified systems management of Oracle Enterprise Manager Ops Center 12c. With the addition of the SPARC SuperCluster, Oracle continues to set the standard for engineered systems: maximizing customer value with leading performance in a complete and tested package.

Oracle SPARC SuperCluster is ideally targeted to existing SPARC database customers or customers preferring a multi-purpose engineered system that hosts a combination
of various database releases (Oracle 10g and up compared to only Oracle 11g and up for Exadata) and application servers. Oracle SPARC Supercluster has additional shared storage with ZFS storage cluster that can be used for non-database files and has virtualization support through Oracle VM Server for SPARC and Oracle Solaris Zones. Exadata Database Machine is still the recommended database machine; however Sparc Supercluster provides other functionality for your application tier if you require a multi-purpose solution.

The recommended Oracle Sparc Supercluster MAA Architecture consists of three elements: 1) a production Sparc Supercluster system (primary), 2) a standby Sparc Supercluster system, and 3) a Sparc Supercluster test or development system. Additionally if non-database files need to be protected in the case of full stack failover or switchover, you can use a combination of Data Guard with ZFS storage remote mirroring. It is also recommended to run Oracle’s Exadata health check (exachk) monthly because it provides the most comprehensive configuration checks for Exadata software, network, and hardware components, and it reports any variance from MAA best practices. See My Oracle Support Note 1070954.1 for information about exachk.

8.3 Oracle Database Appliance

The Oracle Database Appliance is an engineered system consisting of hardware and software that saves customers time and money by simplifying deployment, maintenance, and support of high availability database solutions. Built with Oracle Real Applications Clusters (Oracle RAC) and Oracle Automatic Storage Management best practices, it offers customers a fully integrated system of software, servers, storage and networking in a single box delivering high availability database services for a wide range of home grown and packaged OLTP and Data Warehousing workloads. It comes as a 4RU (rack unit) server appliance that consists of two server nodes and 12 TB raw storage capacity running on Oracle Linux.

Building highly available systems can be difficult and complex and may require advanced integration skills that many organizations don’t have and be risky and error-prone with no vendor accountability. The Oracle Database Appliance is simple, reliable, and affordable.

- **Simple**

To deploy and use the Oracle Database Appliance, simply unpack it, plug in the power cord, plug in the network cables, and run the Oracle Appliance Manager installation to create a clustered, highly available database. The Oracle Database Appliance and its specially engineered software enables "one button" patching for all the elements of the software stack - firmware, operating system, clusterware, storage manager, and database software.

- **Reliable**

The Oracle Database Appliance is built on the Oracle software stack which is completely integrated along with the storage that include 600 GB SAS Hard Disk Drives between the two server nodes that can be triple-mirrored or double mirrored to provide highly available shared storage. This appliance also contains SAS Solid State Drives for redo logs, triple-mirrored to protect the Oracle database in case of instance failure. The appliance manager in conjunction with Oracle Automatic Storage Management (ASM) automatically configures, manages, and monitors the disk for performance and availability. The Oracle Appliance Manager provides alerts on performance and availability events as well as automatically configures replacement drives in case of a hard disk failure.

- **Affordable**
The Oracle Database Appliance offers customers a unique pay-as-you-grow software licensing capability allowing seamless scalability from 4 to maximum processor cores without any hardware upgrades.

The Oracle Database Appliance is an ideal database appliance for customers who value simplicity and who seek to avoid the complexity, costs, and risks in deploying a highly available database solution. The Oracle Database Application is also ideal for database customers who do not require Exadata Database Machine's additional performance and scalability with its Exadata software capabilities and additional availability (rolling patch upgrades vs "one button patching for all the elements of the software stack). Customers can now benefit from high availability (HA) database solutions without having special skills or HA expertise.

The recommended Oracle Database Appliance MAA Architecture consists of three elements: 1) a production Oracle Database Appliance system (primary), 2) a standby Oracle Database Appliance system, and 3) an Oracle Database Appliance test or development system. It is recommended to run Oracle Data Appliance health check (odachk) for its comprehensive configuration checks on operating system, Grid Infrastructure and database settings. See My Oracle Support Note 1485630.1 for information about odachk.

For more information about Oracle Database Appliance, refer to http://www.oracle.com/technetwork/server-storage/engineered-systems/database-appliance/index.html
Optimizing Return on Investment

Oracle Grid Computing, Oracle Active Data Guard real time reporting and utilization, Oracle Database Consolidation using Pluggable Database or Oracle Virtualization and Oracle Global Data Services can all optimize return on investment (ROI) of any of the high availability architectures and solutions.

Oracle Real Application Clusters (Oracle RAC) is the foundation of the Server Grid Computing while Oracle Automatic Storage Management (ASM) and Exadata are the foundation for Storage Grid Computing. Oracle Active Data Guard continues to be strategic for comprehensive data protection, availability and disaster recovery and can provide highly favorable return in investment by offloading reporting, and backups, and doing tests and planned maintenance activities. You can further leverage existing system resources effectively by using database consolidation techniques such as Oracle Pluggable Databases or Oracle Virtualization. Furthermore distributed databases can be used as a collective to maximize all the resources with Global Data Services.

This chapter covers the following topics:

- Grid Computing
- Higher Utilization Using Active Standby Databases
- Oracle Database Consolidation
- Oracle Global Data Services

9.1 Grid Computing

Grid computing is a computing architecture that effectively pools large numbers of servers and storage into a flexible, on-demand computing resource for all enterprise computing needs.

The Oracle Database captures the cost advantages of Grid enterprise computing without sacrificing performance, scalability, security, manageability, functionality, or system availability.

- A Database Server Grid is a collection of commodity servers connected to run one or more databases.
- A Database Storage Grid is a collection of low-cost modular storage arrays combined together and accessed by the servers in the Database Server Grid.

The same grid computing concept applies to primary as well as standby database environments. Figure 9–1 illustrates the Database Server Grid and Database Storage Grid in a grid enterprise computing environment.
The availability of low-cost and reliable blade servers, small multiprocessor servers, and open-source operating systems such as Linux, have made it possible to build a Database Server Grid that is highly available, scalable, flexible, and manageable.

Oracle Real Application Clusters is the technology that enables a Database Server Grid. You can drive down costs by deploying a single Oracle RAC database that spans multiple low-cost servers, each running an active Oracle database instance. Alternatively, you can use a single cluster to minimize the management and increase system utilization across multiple Oracle RAC Databases.

Oracle RAC provides the flexibility to dynamically provision resources and services in the grid as computing needs change, and to add or subtract systems from the grid as capacity demands change. In addition, Oracle RAC provides protection from system failures by automatically transitioning clients and redistributing the processing of the failed node to surviving nodes running the same Oracle RAC database. Note that the scalability and availability benefits of grid computing are not limited to lower cost servers. Any system architecture will benefit from grid computing.

The availability of low-cost Advanced Technology Attachment (ATA) disk-based storage arrays and low-cost storage networks has made it possible to use a Database Storage Grid with Oracle Database at a very low cost. One example solution is Oracle Exadata Storage Servers that offer excellent performance and availability characteristics. Each Exadata Cell can be viewed as a unit of I/O performance and capacity.

The Oracle Storage Grid is implemented using either Oracle Automatic Storage Management (Oracle ASM) and Oracle Exadata Storage Server Software or Pillar SAN Storage Systems or Oracle ASM and third-party storage. The Oracle Storage Grid with Exadata seamlessly supports MAA-related technology, improves performance,
provides I/O scalability, is easy to use and manage, and delivers mission-critical availability and reliability to your enterprise.

A database administrator can use the Oracle ASM interface to specify the disks in the Database Storage Grid that Oracle ASM can manage across all server and storage platforms. Oracle ASM partitions the disk space and evenly distributes the data storage throughout the entire storage array. Additionally, Oracle ASM automatically redistributes the data as disks or storage arrays are added or removed from the Database Storage Grid.

Additionally, use I/O Resource Management to manage and meet service-level requirements. The resource manager allows you manage the grid and prioritized applications within the database or in between databases.

9.2 Higher Utilization Using Active Standby Databases

Data Guard standby databases are an integral part of the Grid, providing data protection, availability and disaster recovery regardless of the cause or scope of an outage. Outages can range anywhere from data corruption that can affect an individual database, to natural disasters that impact a large geographic area.

Advanced Data Guard capabilities deliver maximum ROI by enabling standby databases to be used for productive purposes—such as for read-only queries and reporting—while running in the standby role. Rather than allowing standby databases to remain idle, you can employ them to support activities that would otherwise require you to purchase additional capacity for other systems. Thus, you can defer or eliminate the need to purchase additional capacity for the primary database. This effectively reduces the cost of providing world-class disaster protection for mission critical Oracle Databases.

The following sections describe the Data Guard scenarios that provide high business utilization and a maximum return in investment:

- **Oracle Active Data Guard Option for Physical Standby Databases**
- **Oracle Active Data Guard Reader Farms**
- **Data Guard (Standby) Hub**

9.2.1 Oracle Active Data Guard Option for Physical Standby Databases

Data Guard Redo Apply (physical standby database) is a popular solution for disaster recovery due to its relative simplicity, high performance, and superior level of data protection. The Oracle Active Data Guard option1 (available with Oracle Database 11g Release 1 (11.1) and later releases) enables a physical standby database to be opened for read-only access while Redo Apply is active. Offload capabilities of Oracle Active Data Guard 12c have been enhanced to include: read-only reporting and ad-hoc queries including DML to global temporary tables and unique global or session sequences, data extracts, fast incremental backups, redo transport compression, efficient servicing of multiple remote destinations and the ability to extend zero data loss protection to a remote standby database without impacting primary database performance. Furthermore standby database can help mitigate downtime and risk for planned maintenance activities by using Data Guard standby-first patch apply for patching and transient logical standby for upgrades.

---

1 Oracle Active Data Guard is referred to as real-time query in the Data Guard documentation.
9.2.2 Oracle Active Data Guard Reader Farms

You can use multiple physical standby databases (using the Oracle Active Data Guard option) to deploy an Oracle Active Data Guard reader farm. An example of such a configuration is provided Figure 9-2, complete with the use of Data Guard fast-start failover to automatically fail over should the primary database fail. Note that all standby databases in the reader farm automatically recognize the new primary database after a failover occurs.

A reader farm enables an application to scale read performance of the most demanding web applications beyond what the underlying system and storage architecture can support. This provides a relatively low-cost method of scaling out using a highly redundant Oracle Active Data Guard reader farm architecture where you simply satisfy your increased reporting requirements by adding additional Oracle Active Data Guard standby databases.

The concept is a single primary database that supports read/write transactions, and multiple standby databases that provide read-only access to data. Such an approach scales read performance linearly as additional standby databases are added. It is also an effective way to isolate faults or planned maintenance such as standby-first patching, because problems that affect one standby database are isolated from the other standby databases in the configuration. If there’s a standby database failure or standby system is offline for maintenance, Oracle 12c Global Data Services can be used to transparently failover to the existing standby systems or to the primary.

Creating a reader farm of physical standby databases provides the following benefits:

- Simplicity
- Fault isolation
- High performance with physical standby databases and Redo Apply
- Seamless support for all DDL and data types using Redo Apply
- All reader databases are kept up-to-date with changes made to the primary database
- Automatic, zero or minimal data loss failover capability
- Management as a unified configuration through Grid Control
- Scale-out using single writer database and n reader databases
- Rolling upgrade capabilities
- Integrated client failover to production database or other standby databases using Global Data Services

Figure 9-2 shows a good example of how you can use Data Guard, physical standby databases, and Oracle Active Data Guard option to provide the flexibility necessary to grow your business quickly, while still providing disaster recovery. In the configuration, the primary database transmits redo data to multiple standby databases, one of which is also enabled for fast-start failover for automatic, zero, or minimal data loss failover.
If a fast-start failover is triggered in the Data Guard configuration in Figure 9–2, then:

- Automatic failover occurs to the designated standby database
- All standby databases accept data from the new primary database
- You can perform a switchover at a convenient time in the future to return all databases to their original roles

### 9.2.3 Data Guard (Standby) Hub

With Database Server Grid and Database Storage Grid (described in Section 9.1.1 and Section 9.1.2), you can build standby database and testing hubs that use a pool of system resources. The system resources can be dynamically allocated and deallocated.
depending on various priorities. For example, if the primary database fails over to one of the standby databases in the Data Guard hub, the new primary database acquires more system and storage resources while the testing resources may be temporarily starved. With the Oracle Grid technologies, you can enable a high level of usage and low TCO without sacrificing business requirements.

A Data Guard hub can consist of:

- Several standby databases in an Oracle RAC environment residing in a cluster of servers, called a grid server
- Using the storage grid

The premise of the Data Guard hub is that it provides higher utilization with lower cost. The probability of failing over all databases at the same time is unlikely. Thus, when a failover occurs, you can prioritize the system resources to production activity and allocate new system resources in a grid for the standby database functions. At the time of role transition, more storage and system resources can be allocated toward that application.

For example, a Data Guard hub could include multiple databases and applications that are supported in a grid server and storage architecture. This configuration consists of a central resource supporting 10 applications and databases in the grid, rather than managing 10 separate system or storage units in a nongrid infrastructure.

Another possible configuration might be a testing hub consisting of snapshot standby databases. With the snapshot standby database hub, you can use the combined storage and server resources of a grid instead of building and managing individual servers for each application.

9.3 Oracle Database Consolidation

Database consolidation provides the ability to host multiple applications or databases on the same system platform or within the same database. For the most part, the HA architectures and solutions are still applicable. The benefits of consolidation is the reduce cost of ownership and management while leveraging all resources as effectively as possible; however the trade-off is a reduced level of isolation and independence compared to having separate database or system resources for each application and database.

9.3.1 Multitenant Architecture

Multitenant architecture is the capability that enables an Oracle database to contain a portable set of schemas, objects, and related structures that appears logically to an application as a separate database. This self-contained collection is called a pluggable database (PDB). A multitenant container database (CDB) contains PDBs.

Multitenant architecture is the most cost-effective form of database consolidation. By consolidating multiple physical databases on separate computers into a single database on an optimized engineered platform such as Exadata Database Machine, you gain the following benefits:

- Cost reduction for hardware
- Portability of an application’s database back end
- Ease of database and system administration
- Centralized management of database accounts and privileges
- Easier and faster upgrade paths
Multitenant architecture is especially useful when you have many databases deployed on different hardware in multiple Oracle Database installations. Each PDB might use only a fraction of the hardware resources dedicated to it, and each PDB might not require a full-time database administrator to manage it.

By combining these databases into a CDB, you can make better use of your hardware resources and database administrator resources. In addition, you can move PDBs from one CDB to another without requiring changes to the applications that depend on the PDB.

Multitenant architecture can leverage the proposed HA architectures in this book. For example, the targeted PDB can reside on Exadata Database Machine leveraging Oracle Real Application Clusters, Oracle Automatic Storage Management, and Exadata Storage Cells and have an additional Standby PDB residing on a separate Exadata Database Machine.

See Also:
- Oracle Database Administrator’s Guide for information about creating and administering pluggable databases.

9.3.2 Oracle Virtualization

Data centers today use virtualization techniques to make abstraction of the physical hardware, create large aggregated pools of logical resources consisting of CPUs, memory, disks, file storage, applications, networking, and offer those resources to users or customers in the form of agile, scalable, consolidated virtual machines. Even though the technology and use cases have evolved, the core meaning of virtualization remains the same: to enable a computing environment to run multiple independent systems at the same time with the main intent of saving people and hardware resources.

Oracle has three main virtualization technologies:

- Oracle VM for X86 and Oracle VM Manager are an enterprise-class server virtualization solution. Oracle VM Server for x86 is the most scalable x86 server virtualization solution in the market today, and it has been tested to handle mission critical enterprise workloads with support for up to 160 physical CPUs and 2 TB of memory. For virtual machines, Oracle VM 3 can support up to 128 virtual CPUs and 1TB memory per guest VM. Oracle VM supports industry standard x86 operating systems and servers from Oracle and other leading vendors, and it supports a broad range of network and storage devices, making it easy to integrate into your environment. Oracle VM Manager provides an easy-use-centralized management environment for configuring and operating your server, network, and storage infrastructure from a browser based interface (no Java client required), and it is accessible from just about anywhere.

- Oracle VM Server for SPARC provides highly efficient, enterprise-class virtualization capabilities for Oracle’s SPARC T-Series servers. Using the Oracle VM Server for SPARC software, you can create up to 128 virtual servers, called logical domains, on a single system. This kind of configuration enables you to take advantage of the massive thread scale offered by SPARC T-Series servers and the Oracle Solaris OS.

- Oracle Solaris Zones software partitioning technology, which provides a means of virtualizing operating system services to create an isolated environment for running applications. This isolation prevents processes that are running in one zone from monitoring or affecting processes running in other zones. Zones can be
used on any machine that is running the Oracle Solaris 10 or a later Oracle Solaris release. The upper limit for the number of zones on a system is 8192.

Oracle virtualization can be used in conjunction with HA features and HA architectures to reap the benefits of both target goals. Here are some of the HA benefits when integrating Oracle virtualization with HA architecture and features.

- Auto restart of VMs in the event of a failure making applications HA
- Oracle Real Application Clusters ensure business availability at the application layer and is integrated with Oracle VM to ensure business availability on the server as well as application data in a single or multiple geographic locations
- Generally any Oracle high availability feature, such as RMAN, flashback technologies, Data Guard, and Oracle GoldenGate, that works natively in non-virtualized environments will work seamlessly in a virtualized environment.
- Oracle VM accelerates the delivery of services to meet changing business need. This allows online growing of capacity

See Also:

- The Oracle VM website on OTN at http://www.oracle.com/virtualization

9.4 Oracle Global Data Services

A Global Data Services (GDS) configuration is a set of databases integrated by the GDS framework into a single virtual server that offers one or more global services, while ensuring high performance, availability and optimal utilization of resources. GDS manages these virtual resources with minimum administration overhead, and allows the GDS configuration to quickly scale to handle additional client requests.

The databases that constitute the GDS configuration can be globally distributed or located within the same data center. Clients can securely connect to the gds configuration by simply specifying a service name, without needing to know anything about the components and topology of the GDS configuration, enabling a highly flexible private cloud deployment for the enterprise.

GDS enables customers to scale their disparate computing resources and heterogeneous platforms in a highly flexible way, without requiring any application changes. Geographically dispersed data centers, whether regional or global, can now be effectively utilized within a uniform framework based on business, throughput, and localized demands, without affecting run-time applications. This additional computing scale will enable a truly elastic and agile enterprise and extend the benefits of cloud computing to all employees, business partners and stakeholders.

See Also: Oracle Database Global Data Services Concepts and Administration Guide
business impact analysis
An impact analysis that categorizes the business processes based on the severity of the impact of IT-related outages.

clusterwide failure
The whole cluster hosting the Oracle RAC database is unavailable or fails. This includes failures of nodes in the cluster, and any other components that result in the cluster being unavailable and the Oracle database and instances on the site being unavailable.

computer failure
An outage that occurs when the system running the database becomes unavailable because it has crashed or is no longer accessible.

cost of downtime
A complete business impact analysis provides the insight needed to quantify the cost of unplanned and planned downtime. Understanding this cost is essential because it helps prioritize your high availability investment and directly influences the high availability technologies that you choose to minimize the downtime risk.

data corruption
A corrupt block is a block that has been changed so that it differs from what Oracle Database expects to find. Block corruptions fall under two categories: physical and logical block corruptions.

See also physical corruption and logical corruption.

hang or slow down
Hang or slow down occurs when the database or the application cannot process transactions because of a resource or lock contention. Perceived hang can be caused by lack of system resources.

human error
An outage that occurs when unintentional or malicious actions are committed that cause data in the database to become logically corrupt or unusable. The service level impact of a human error outage can vary significantly depending on the amount and critical nature of the affected data.

logical corruption
The contents of the block are logically inconsistent. Examples of logical corruption include corruption of a row piece or index entry.
**logical unit numbers (LUNs)**

Three-bit identifiers used on a SCSI bus to distinguish between up to eight devices (logical units) with the same SCSI ID.

**lost write**

A lost write is another form of data corruption that can occur when an I/O subsystem acknowledges the completion of the block write, while in fact the write I/O did not occur in the persistent storage. No error is reported by the I/O subsystem back to Oracle Database.

**MAA environment**

An architecture that provides the most comprehensive set of solutions for both unplanned and because it inherits the capabilities and advantages of both Oracle Database 11g with Oracle RAC and Oracle Database 11g with Data Guard.

The MAA environment consists of a site containing an Oracle RAC primary database and a second site containing a standard cluster that hosts both logical and physical standby databases, or at least one physical or logical standby database.

**manageability goal**

More subjective than either the RPO or the RTO, the manageability goal results from an objective evaluation of the skill sets and management resources available in an organization, and the degree to which the organization can successfully manage all elements of a high availability architecture. Understanding manageability goals helps organizations differentiate between what is possible and what is practical to implement.

**network failure**

A network failure occurs when a network device stops or reduces network traffic and communication from your application to database, database to storage, or any system to system that is critical to your application service processing.

**network server processes**

The Data Guard network server processes, also referred to as LNSn processes, on the primary database perform a network send to the RFS process on the standby database. There is one network server process for each destination.

**Oracle Active Data Guard option**

A physical standby database can be open for read-only access while Redo Apply is active if a license for the Oracle Active Data Guard option has been purchased. This capability, known as Oracle Active Data Guard, also provides the ability to have block-change tracking on the standby database, thus allowing incremental backups to be performed on the standby.

*Note:* The Oracle Active Data Guard option may also be referred to as “real-time query” in other documentation.

**physical corruption**

The database does not recognize the block at all: the checksum is invalid, the block contains all zeros, or the header and footer of the block do not match. A physical corruption is also called a media corruption.

**recovery point objective (RPO)**

The maximum amount of data an IT-based business process may lose before causing harm to the organization. RPO indicates the data-loss tolerance of a business process.
or an organization in general. This data loss is often measured in terms of time, for example, five hours or two days worth of data loss.

**recovery time objective (RTO)**
The maximum amount of time that an IT-based business process can be down before the organization suffers significant material losses. RTO indicates the downtime tolerance of a business process or an organization in general.

**return on investment (ROI)**
Return on Investment (or Rate of return) is used to evaluate the efficiency of an investment in finance and economics.

**site failure**
An outage that occurs when an event causes all or a significant portion of an application to stop processing or slow to an unusable service level. A site failure may affect all processing at a data center, or a subset of applications supported by a data center.

**storage failure**
An outage that occurs when the storage holding some or all of the database contents becomes unavailable because it has shut down or is no longer accessible.

**total cost of ownership (TCO)**
A financial estimate designed to help consumers and enterprise managers assess direct and indirect costs. It is used in many industries and is a form of full cost accounting.

**transient logical standby database**
A transient logical standby database allows you to reuse your current physical standby database by temporarily converting it into a logical standby on which to perform a rolling database upgrade, incurring minimal downtime.
transient logical standby database
Index

Numerics
24x365, 1-1
64-bit systems
  migration from 32-bit, 5-20

A
access control
  security, 3-23
Active Data Guard
  See Oracle Active Data Guard
ADD COLUMN
  default values for columns, 5-28
analysis
  determining high availability requirements, 2-1
applications
  failover, 7-9
  online maintenance and upgrades, 5-27
applying interim database patches, 5-20
architectures
  choosing, 7-17, 7-19
  Data Guard, 7-8
  Data Guard standby hub, 7-13, 9-5
  failures in, 1-1
  MAA, 7-1
  manageability, 2-4
  multiple standby database, 7-13
  Oracle Application Server, 7-27, 7-28
  Oracle Clusterware and Data Guard, 9-5
  Oracle Database, 7-2
  Oracle RAC, 7-4, 7-5
  Oracle RAC and Data Guard, 7-15
  Oracle RAC on extended clusters, 7-6
  Oracle RAC One Node, 7-3
  Oracle RAC One Node and Data Guard, 7-15
  Oracle Restart and Data Guard, 7-15
  recommendations, 7-11, 7-22
  requirements, 1-1, 2-1, 2-5
  roadmap, 1-8
  same processor platforms, 5-19
  single standby database, 7-8, 7-11
ASR
  See Oracle Auto Service Request (ASR)
auditing
  security control, 3-23
authentication
  security controls, 3-23
automatic block repair, 3-27
automatic corruption repair, 7-9
automatic failover
  Data Guard, 7-11
automatic shared memory management
  dynamic memory allocation, 5-10
availability
  about, 1-1
  disruptions, 1-3
  roadmap, 1-7
  See also high availability

B
backing out a transaction, 3-25
backups
  Oracle Secure Backup, 3-14
  between objects, 5-29
block recovery
  using Flashback logs, 3-27
budget planning, 2-5
business impact analysis
  internal knowledge management system example, 2-3
  semiconductor manufacturer example, 2-3
business performance planning, 2-5

C
choosing the correct high availability architecture, 7-17
Cluster Ready Services (CRS)
  avoiding downtime during upgrades, 5-21
clusters
  extended, 7-6
  columns, invisible, 5-29
  compressed redo data
    Data Guard, 7-9
  computer failure, 1-4
corruptions
  automatic repair, 7-9
  prevention and detection, 3-20
  prevention, detection, and repair, 7-2
costs
quantifying, 2-3
CREATE TRIGGER statement
  clauses for, 5-28
crossedition triggers, 5-28

D
data corruptions, 1-5
detecting, 3-20
  prevention and detection parameters, 3-20
data distribution
  Oracle GoldenGate, 3-8
data encryption, 3-23
Data Guard
  about, 3-1
  benefits, 3-2, 7-9
  comparison to remote mirroring solutions, 7-9
  configured with Oracle Restart, 7-15
  configuring with Oracle GoldenGate, 5-25
  data protection features, 7-11
  hardware requirements, 7-11
  hub architecture, 9-6
  multiple standby database architecture, 7-13
  single standby database architecture, 7-11
  system and cluster upgrades, 5-19
Oracle Data Guard
  See Data Guard
data integration
  Oracle GoldenGate, 3-8
data protection, 7-10
  maximizing, 1-2
Data Recovery Advisor, 3-21, 7-2
data resilience, 7-10
Database Server Grid
  about, 9-2
Database Storage Grid
  about, 9-2
database upgrades
  using transportable tablespace, 5-24
databases
  applying Oracle interim patches, 5-20
  dynamic reconfiguration, 5-10
  security, 3-22
datafiles, moving online, 5-9
data-loss tolerance, 2-4
DBA_FLASHBACK_TRANSACTION_STATE
  view, 3-25
DBMS_FLASHBACK.TRANSACTION_BACKOUT()
  procedure, 3-25
DBWR process, 7-10
DDL with the WAIT option, 5-28
dependencies, 5-29
DISABLE clause
  FOLLOWS clause
    CREATE TRIGGER statement, 5-28
disaster recovery solutions
  Oracle Application Server, 7-28
disk group
  administering with Oracle ASM, 3-19
downtime
  causes, 1-3
  cost, 1-3, 2-3
  minimizing with Oracle GoldenGate, 5-25
  minimizing with Oracle GoldenGate and Data Guard, 5-25
  mitigating, 1-3
  reducing, 7-9
  solutions summary
    planned, 5-17
See also planned downtime
See also unplanned downtime
dynamic reconfiguration, 5-10
dynamic resource provisioning, 7-2

E
edition-based redefinition, 5-27
  crossedition triggers, 5-28
  editioning view, 5-27
  editions, 5-27
ENABLE clause
  CREATE TRIGGER statement, 5-28
encryption
  of data, 3-23
Enterprise System Connection (ESCON), 7-10
extended clusters
  architecture, 7-6
Extended Datatype Support (EDS)
  patch set and database upgrades, 5-22

F
failovers
  applications, 7-9
  fast, 7-11
  services, 7-5
  single standby database architecture, 7-11
  standby database architecture, 7-13
failure group
  administering with Oracle ASM, 3-19
  Oracle ASM, 3-19
failures
  computer, 1-4
  probability, 7-17
  site, 1-4
  storage, 1-4
Far Sync Instance, xi
fast application notification (FAN)
  for hardware upgrades, 5-18
  for operating system upgrades, 5-18
Fast Connection Failover
  for nonpooled connections, 3-17
Fast Mirror Resync
  Oracle ASM, 3-19
fast recovery area
  about, 3-20
  benefits, 3-20
  for standalone databases, 7-2
  in a Data Guard configuration, 7-10
fast-start failover, 7-8
  single standby database failover, 7-11
Fibre Channel, 7-10
file system
  NFS, 7-7
  SAN, 7-7
flashback logs, 7-10
  block recovery using, 3-27
Flashback technology
  See also Oracle Flashback technology
forward crossedition triggers, 5-28
frequency of outages, 7-17

G
  grid computing, 9-1
  growth planning, 2-5

H
  hangs or slow down, 1-6
  hardware upgrades
    avoiding downtime during, 5-18
    using FAN during, 5-18
  high availability
    24x365, 1-1
    about, 1-1
    analysis framework, 2-1
    applications, 7-28
    architectures, 1-2, 2-4, 7-1, 7-17
    business impact analysis, 2-3
    determining requirements, 2-1
    features of Oracle Database, 7-2
    importance, 1-2
    maximizing, 1-2
    Oracle Application Server, 7-28
    Oracle Database, 7-1
    planned downtime, 5-16
    planning, 2-5
    setting manageability goals, 2-4
    solutions, 1-1
    unplanned downtime, 4-1
    See also availability
  human errors, 1-6

I
  indexes
    invisible, 5-29
  indexes, multiple on same set of columns, 5-30
  instance failure, 7-5
  interblock corruption, 1-5
  interconnects
    Oracle RAC, 7-5
  intrablock corruption, 1-5
  invisible columns, 5-29
  invisible indexes, 5-29

L
  load balancing
  advisory, 7-5
  run-time connection, 7-5
  load balancing advisory, 3-17
  logical corruption, 1-5
  logical standby databases
    recommendations, 7-12
  lost writes, 1-6

M
  making data changes, 5-27
  manageability
    goals, 2-4
    overhead (MO), 2-4, 7-17
  Maximum Availability Architecture
    See Oracle Maximum Availability Architecture (MAA)
  media corruption
    physical corruption, 1-5
  memory
    automatic management of, 5-10
    MEMORY_MAX_TARGET initialization parameter, 5-10
    MEMORY_TARGET initialization parameter, 5-10
  metadata
    dependencies, 5-29
    migrating storage
      avoiding downtime, 5-8
      migrations
        32-bit to 64-bit systems, 5-20
        storage, 5-8
    mirroring
      Oracle ASM native, 3-19
      multiple standby databases
        failovers, 7-13

N
  network bonding, 7-5
  network efficiency, 7-9
  network file system (NFS), 7-7

O
  observer
    fast-start failover, 7-11
  offloading database activity, 7-9
  one-off patches, 5-20
  Online Application Maintenance and Upgrades, 7-2
  online database relocation utility, 7-3
  online maintenance
    application, 5-27
    online patching, 7-2
    online redefinition, 7-2
    of tables, 5-11
    online reorganization, 7-2
    online table redefinition, 5-30
    online, moving datafiles, 5-9
  OPatch utility
    patch upgrades for Oracle RAC, 5-21
    operating systems

Index-3
upgrades, 5-18
using FAN during upgrades, 5-18
Oracle Active Data Guard
benefits of standby databases, 3-4
for physical standby databases, 9-3
Oracle Application Server
high availability architectures, 7-28
security, 7-28
Oracle ASM Cluster File System (ACFS), 3-29
Oracle Auto Service Request (ASR), 6-8
Oracle Automatic Storage Management (Oracle ASM), 7-2
about, 3-18
benefits, 3-19
distribution of files, 5-11
failure group, 3-19
Fast Mirror Resync, 3-19
native mirroring, 3-19
storage migration, 5-22
upgrading, 5-22
Oracle Automatic Storage Management Cluster File System (Oracle ACFS), 3-19, 7-2
Oracle Call Interface (OCI), 3-17
Oracle Clusterware
avoiding downtime when upgrading, 5-21
cold cluster failover, 7-4, 7-6
Oracle Data Provider for .NET (ODP.NET), 3-17
Oracle Database
basic architecture, 7-2
with Data Guard, 7-8
with Oracle RAC, 7-4
with Oracle RAC on an extended cluster, 7-6
with Oracle Restarts and Data Guard, 7-15
Oracle Database File System (DBFS), 3-28, 7-9
Oracle Database Resource Manager Instance Caging, 7-3
Oracle Enterprise Manager, 7-2, 7-6
Oracle Exadata Database Machine, 8-1
Oracle Exadata Storage Server Software combined with Oracle Database Machine, 8-1
upgrading, 5-22
Oracle Flashback Data Archive, 3-27
Oracle Flashback Database, 3-26, 7-2
Oracle Flashback Drop, 3-25
Oracle Flashback Query, 3-24
Oracle Flashback Table, 3-25
Oracle Flashback technology, 3-23 to 3-27, 7-2
block recovery using Flashback logs, 3-27
Oracle Flashback Transaction, 3-25
Oracle Flashback Transaction Query, 3-25
Oracle Flashback Version Query, 3-24
Oracle GoldenGate
about, 3-8
configure to minimize downtime, 5-25
rolling upgrades, 5-28
upgrades, 5-25
Oracle interim (one-off) patches, 5-20
applying, 5-20
avoiding downtime during, 5-20
Oracle Maximum Availability Architecture (MAA)
about, 1-7
architectures, 7-1
roadmap, 1-8
Oracle Quality of Service (QoS) Management, 7-5
Oracle RAC One Node, 7-3, 7-4, 7-6
online database relocation utility, 7-3
Oracle Real Application Clusters One Node
See Oracle RAC One Node
Oracle Real Application Clusters (Oracle RAC)
about, 7-4
applying Oracle interim database patches, 5-20
automatic redundant interconnects, 7-5
benefits, 3-17, 7-5
extended clusters, 7-6
operating system and hardware upgrades, 5-18
Storage Area Network (SAN), 7-7
Oracle Restart
configured with Data Guard, 7-15
Oracle Secure Backup, 7-2
about, 3-14
benefits, 3-15
Oracle Security features, 7-2
Oracle Solaris ZFS Storage Appliance Replication, 3-29
Oracle UCP run-time connection load balancing, 3-17
Oracle VM
Domain Live Migration, 9-7
with Oracle RAC One Node, 7-4
outages frequency, 7-17
types of, 1-3
P
patching
online, 7-2
rolling, 7-5
physical corruption, 1-5
physical standby databases real-time query, 9-3
planned activities probability of failure during, 7-17
planned downtime online patching, 5-18
recovery times, 7-27, 7-28
planned outages
minimizing with Oracle GoldenGate, 5-25
minimizing with Oracle GoldenGate and Data Guard, 5-25
Plug and Play, 7-5
pluggable databases, 9-6
policy management security, 3-23
policy-based cluster management, 7-5
portable databases, 9-6
prioritizing
high availability investment, 2-3
probability
of different failures during unplanned and planned activities, 7-17
Program Global Area (PGA) automatic management, 5-10

R
real-time query, 9-3
recommended architectures, 7-11
reconfiguring databases dynamically, 5-10
Recovery Manager (RMAN), 7-2
about, 3-13
benefits, 3-13
recovery point objective (RPO) about, 2-4, 7-17
recovery time objective (RTO) about, 2-4, 7-17
recovery times
planned downtime, 7-27, 7-28
unplanned downtime, 7-26
redundant interconnect, 7-5
relocation online database, 7-3
remote mirroring
comparison to Data Guard, 7-9
replication
Oracle GoldenGate, 3-8
restore points
Oracle Flashback, 3-26
return on investment (ROI), 2-4, 7-17
and Data Guard, 7-11
reverse crossedition triggers, 5-28
roadmap to Maximum Availability Architecture (MAA), 1-7
rollback
transactions, 3-25
rolling patching, 7-5
rolling upgrades, 7-5
Oracle GoldenGate, 5-28
row level security
virtual private database, 3-22
run-time connection load balancing, 3-17, 7-5

S
scalability
with multiple standby databases, 7-14
with Oracle RAC, 7-5
SCAN, 7-6
security
about, 3-22
benefits, 3-22
data encryption, 3-23
features of Oracle Database, 7-2
Oracle Application Server, 7-28
Oracle ASM, 3-19
RMAN, 3-14
service-level agreements (SLAs), 2-2

failover, 7-5
simple database rolling upgrades, 5-23
site failure, 1-4
SLAs, 2-2
SONET networks, 7-10
SQL Apply, 5-22, 5-23
standby databases
active, 9-3
benefits, 3-2
Data Guard hub, 9-5
example hub configurations, 9-6
failover with a single standby database, 7-11
key traits, 7-11
multiple-standby architecture, 7-13
Oracle Active Data Guard, 9-3
single-standby architecture, 7-11
standby reader farms, 9-4
storage
failures, 1-4, 3-19
migration, 5-8
Oracle ASM protection, 3-18
Storage Area Network (SAN) extended clusters, 7-7
stray writes, 7-10
System Global Area (SGA)
amtomatic management, 5-10
system upgrades
avoiding downtime during, 5-18

T
tables
editionable, 5-27
tape backups
with Oracle Secure Backup, 3-14
thin client watchdog
observer for fast-start failover, 7-11
total cost of ownership (TCO), 2-4, 7-17
transactions
backing out with Flashback Transaction, 3-25
transportable tablespace
upgrading the database, 5-24

U
unplanned activities
probability of failure during, 7-17
unplanned downtime
causes, 1-4
recovery times, 7-26
solutions summary, 4-1
upgrade
simple rolling, 5-23
using Oracle GoldenGate, 5-25
upgrades
application, 5-27
cluster, 5-19
database, 5-22
hardware, 5-18
operating system, 5-18, 5-19
Oracle ASM, 5-22
Oracle Clusterware, 5-21
Oracle Exadata Storage Server Software, 5-22
Oracle Real Application Clusters (Oracle RAC), 5-18
patches, 5-20
rolling, 7-5
rolling with Oracle GoldenGate, 5-28
SQL Apply, 5-22
using crossedition triggers, 5-28
using transportable tablespace, 5-24

V
virtual IP (VIP) address
  managed by Oracle Clusterware, 3-17
virtual private database
  security, 3-22
virtualization, 7-4
  with Oracle VM Domain Live Migration, 9-7

W
WAIT option
  specifying DDL with, 5-28
web scalability
  using standby reader farms, 9-4
wide area networks (WANs), 7-10
workload
  offloading, 7-9
workload management, 7-5

Z
zero data loss, 7-9