Oracle® Database

High Availability Overview

11*g* Release 2 (11.2)

E10804-02

September 2009

ORACLE®

Oracle Database High Availability Overview, 11*g* Release 2 (11.2)

E10804-02

Primary Authors: Viv Schupmann, Lawrence To

Contributors: Lance Ashdown, Andrew Babb, Tammy Bednar, Peter Belknap, Janet Blowney, Larry Carpenter, Immanuel Chan, Tim Chien, Donna Cooksey, Tulika Das, Mark Dilman, Ray Dutcher, Richard Exley, Craig Foch, Ameet Kini, Frank Kobylanski, Bryn Llewellyn, Barb Lundhild, Rahim Mau, Patricia McElroy, Joe Meeks, Valarie Moore, Dan Norris, Michael Nowak, Darryl Presley, Ashish Ray, Jia Shi, Michael T. Smith, Vinay Srihari, Hubert Sun, Douglas Utzig, James Viscusi, Shari Yamaguchi

Contents

4 Oracle Database High Availability Solutions for Planned Downtime

List of Figures

List of Tables

Preface

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

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

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an

otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

Knowledge of Oracle Database, Oracle RAC, and Oracle 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 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 Database Storage 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. 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.otn.oracle.com/goto/maa`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview of High Availability

This chapter contains the following sections:

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 end 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 ready to serve them at all times 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**. However, exceptions can be made for minimal downtime to perform certain operations such as upgrading the 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 in a timely manner that meets 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 end 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 end 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

Type	Description	Examples
Site failure	A site failure may affect all processing at a data center, or a subset of applications supported by a data center.	■ Extended sitewide power failure ■ Sitewide network failure ■ Natural disaster makes a data center inoperable ■ Terrorist or malicious attack on operations or the site
Clusterwide failure	The whole cluster hosting an Oracle RAC database is unavailable or fails. This includes: ■ Failures of nodes in the cluster ■ Failure of any other components that result in the cluster being unavailable and the Oracle database and instances on the site being unavailable	■ The last surviving node on the Oracle RAC cluster fails and inability to restart the node ■ Both redundant cluster interconnects fail or clusterware failure ■ Database corruption so severe that continuity is not possible on the current data server ■ Disk storage failure
Computer failure	A computer failure outage occurs when the system running the database becomes unavailable because it has failed or is no longer accessible.	■ Database system hardware failure ■ Operating system failure ■ Oracle instance failure ■ Network interface failure
Storage failure	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 accessible.	■ Disk drive failure ■ Disk controller failure ■ Storage array failure
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 the following categories: physical and logical block corruptions: ■ 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).	■ Operating system or storage device driver failure ■ Faulty host bus adapter ■ Disk controller failure ■ Volume manager error causing a bad disk read or write ■ Software defects

Table 1–1 (Cont.) Causes of Unplanned Downtime

Type	Description	Examples
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
Hang or slowdown	Hang or slowdown occurs when the database or the application is unable to process transactions because of a resource or lock contention. A perceived hang can be caused by lack of system resources.	▪ Database or application deadlocks ▪ Runaway processes that consume system resources ▪ Log on 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.

Table 1–2 Causes of Planned Downtime

Type	Description	Examples
System and database changes	Planned system changes occur when performing routine and periodic maintenance operations and new deployments. Planned system changes include any scheduled changes to the operating environment that occur outside the organizational data structure in the database. The service-level impact of a planned system change varies significantly depending on the nature and scope of the planned outage, the testing and validation efforts made before implementing the change, and the technologies and features in place to minimize the impact.	▪ Adding or removing processors to or from an SMP server ▪ Adding or removing nodes to or from a cluster ▪ Adding or removing disks drives or storage arrays ▪ Changing configuration parameters ▪ Upgrading or patching system hardware and software ▪ Upgrading or patching Oracle software ▪ Upgrading or patching application software ▪ System platform migration ▪ Database relocation ▪ Moving from 32 bits to 64 bits ▪ Migrating to cluster architecture ▪ Migrating to new storage
Data changes	Planned data changes occur when there are changes to the logical structure or physical organization of Oracle Database objects. The primary objective of these changes is to improve performance or manageability.	▪ Table definition changes ▪ Adding table partitioning ▪ Creating and rebuilding indexes
Application changes	Planned application changes may include data changes and schema and programmatic changes. The primary objective of these changes is to improve performance, manageability, and functionality.	▪ Application upgrades

Oracle offers high availability solutions to help avoid both unplanned and planned downtime, and recover from failures. Chapter 3 and Chapter 4 discuss each of these high availability solutions in detail.

1.5 Roadmap to Implementing the Maximum Availability Architecture (MAA)

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

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

- Encompasses redundancy across all components
- Provides protection and tolerance from computer failures, storage failures, human errors, data corruption, lost writes, system hangs 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
- Achieves SLAs at the lowest possible total cost of ownership

To build, implement and maintain such an architecture, you need to:

1. Understand the key effects of the Oracle high availability features on businesses and applications, as described in Chapter 3 and Chapter 4.
2. 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"
3. Choose a high availability architecture, as described in Chapter 7, "High Availability Architectures and Solutions"
4. 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.

     You can download these MAA resources from the following Web site

     `http://www.otn.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.

An enterprise with a well-articulated set of high availability best practices that encompass high availability analysis frameworks, business drivers, and system capabilities, enjoys an improved operational resilience and enhanced business agility.

2

Determining Your High Availability Requirements

This chapter includes the following sections:

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 will achieve both business and technical objectives.

You can use the high availability analysis framework to:

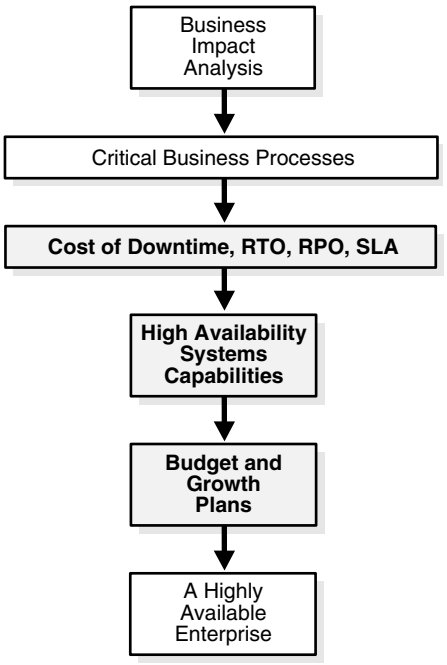
1. Complete a **business impact analysis**
2. Identify and categorize the critical business processes that have the high availability requirements
3. Formulate the **cost of downtime**
4. Establish utilization, **recovery time objective (RTO)**, and **recovery point objective (RPO)** goals for these various business processes
5. Understand goals for manageability, **total cost of ownership (TCO)**, and **return on investment (ROI)**

This framework enables the business to define service-level agreements (SLAs) in terms of high availability for critical aspects of its business. For example, it can categorize its business processes into several high availability tiers:

- Tier 1 processes have 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 need not 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 (RTO)

- Recovery Point Objective (RPO)
- Manageability Goal
- Total Cost of Ownership (TCO) and Return on Investment (ROI)

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 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 will affect 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. This leads us to the next element in the high availability requirements framework: *cost of downtime*.

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 various 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 and the reasons are obvious. 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 (RTO)

The business impact analysis will determine 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 very near to zero.

An organization is likely to have varying RTO requirements across its various business processes. Thus, for a high volume e-commerce Web site, 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 via the e-commerce Web site immediately (the RTO) may be more important than the RPO, because lost data can be reloaded later.

2.2.4 Recovery Point Objective (RPO)

The business impact analysis also determines your **recovery point objective (RPO)**. RPO is the maximum amount of data that an IT-based business process may 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 strictly require an RPO of zero, although a low RPO is essential for customer satisfaction. However, its back-end merchandising and inventory update system may have a higher RPO; as 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 (TCO) and Return on Investment (ROI)

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 end-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 details about high availability system capabilities and business performance, budget and growth plans.

See Also: "Choosing the Correct High Availability Architecture" on page 7-23

2.3.1 Business Performance, Budget, and Growth Plans

High availability solutions must also be based on business performance issues. 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, may affect 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 tradeoff, 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 may lead to budget overruns or an unmanageable and unscalable combination of solutions that is extremely 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 would be 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
- Forces the use of a specific vendor

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.

3

Oracle Database High Availability Solutions for Unplanned Downtime

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, seven days a week. However, the Oracle high availability solutions go beyond reducing downtime by providing solutions to increase system utilization on the primary and secondary systems and to help improve overall performance, scalability, and manageability.

Oracle provides the following features for high availability for unplanned downtime:

- Fast-Start Fault Recovery
- Oracle Restart
- Oracle Real Application Clusters and Oracle Clusterware
- Oracle Data Guard
- Oracle Streams
- Oracle Flashback Technology
- Oracle Automatic Storage Management
- Fast Recovery Area
- Recovery Manager
- Data Recovery Advisor
- Oracle Secure Backup
- Oracle Security Features
- LogMiner
- Oracle Exadata Storage Server Software (Exadata Cell)
- Oracle Database File System (DBFS)
- Client Failover
- Automatic Block Repair
- Corruption Prevention, Detection, and Repair

Also, Chapter 4, "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.

See Also:

- The "High Availability" chapter in *Oracle Database Concepts* for an overview of the high availability features
- The "Availability" section in the *Oracle Database New Features Guide* for a list of the high availability features introduced in Oracle Database 11*g* Release 2 (11.2)

3.1 Oracle High Availability Solutions and Recovery for Unplanned Downtime

Oracle Database provides high availability solutions to prevent and reduce downtime for all types of unplanned failures.

Table 3–1 describes the various Oracle high availability solutions for unplanned downtime. The table shows how the features discussed in Section 3.2 through Section 3.19 can be used to address various causes of unplanned downtime. Also, see Table 7–4 for a summary of the attainable recovery times for all types of unplanned downtime for each Oracle high availability architecture.

Table 3–1 Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Site Failures	Oracle Data Guard	▪ Fast-start failover and FAN with integrated Oracle clients
Site Failures	Oracle Streams	▪ Online replica database resumes processing
Site Failures	Recovery Manager	▪ Fully managed database recovery and integration with Oracle Secure Backup
Computer Failures	Oracle Real Application Clusters and Oracle Clusterware	▪ Automatic recovery of failed nodes and instances ▪ Fast application notification (FAN) with integrated Oracle client failover
Computer Failures	Fast-Start Fault Recovery	▪ Tunable and predictable cache recovery from computer failures
Computer Failures	Oracle Data Guard	▪ Fast-start failover and FAN with integrated Oracle clients
Computer Failures	Oracle Streams	▪ Online replica database resumes processing.
Storage Failures	Oracle Automatic Storage Management	▪ Mirroring and online automatic rebalancing places redundant copies of the data in separate failure groups.
Storage Failures	Oracle Data Guard	▪ Fast-start failover and FAN with integrated Oracle clients
Storage Failures	Recovery Manager with Fast Recovery Area and Oracle Secure Backup	▪ Fully managed database recovery and managed disk and tape backups
Storage Failures	Oracle Streams	▪ Online replica database resumes processing.

Table 3–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Data Corruption	Oracle Exadata Storage Server Software (Exadata Cell) and Oracle Automatic Storage Management	■ 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 is the most comprehensive solution, to prevent corruptions from being written to disk, and it is HARD compliant[1]
Data Corruption	Corruption Prevention, Detection, and Repair Database initialization settings such as `DB_ULTRA_SAFE`, `DB_BLOCK_CHECKING`, `DB_BLOCK_CHECKSUM`	■ Different levels of block corruption prevention and detection at the database level
Data Corruption	Data Recovery Advisor and Recovery Manager with Fast Recovery Area	■ Data Recovery Advisor automatically detects data corruptions and recommends the best recovery plan. ■ RMAN online block-media recovery time is faster because RMAN can use flashback logs to restore a more current copy of the data block for recovery.
Data Corruption	Oracle Data Guard	■ Repair primary data blocks in real time by fetching a good version from a physical standby database ■ Fast-start failover and FAN with integrated Oracle clients
Data Corruption	Oracle Streams	■ Processing resumes on the online replica database
Human Errors	Oracle Security Features	■ Restrict access as prevention
Human Errors	Oracle Flashback Technology	■ Fine-grained and database-wide rewind capability
Human Errors	LogMiner	■ Redo log analysis

Table 3–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Lost writes	Oracle Data Guard, Recovery Manager, and the DB_LOST_WRITE_PROTECT initialization parameter Also, setting DB_ULTRA_SAFE to DATA_ONLY or DATA_AND_INDEX automatically enables DB_LOST_WRITE_PROTECT.	▪ DB_LOST_WRITE_PROTECT initialization parameter provides lost write detection. ▪ 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 Oracle 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. **Note:** Lost writes can corrupt the entire database, which may require that you rebuild the affected database after resolving the hardware issue.
Lost writes	Oracle Data Guard Oracle Exadata Storage Server Software (Exadata Cell)	▪ Detection and prevention of stray or misdirected writes to another data file. Check with your HARD-compatible storage vendor to learn whether the vendor has implemented this additional protection. For the most comprehensive lost write protection, use Oracle Data Guard ▪ HARD does ***not*** detect a lost write in the following cases: *If any layer of software or hardware (host driver, volume manager, host bus adapter, storage array firmware) acknowledges the write but did not issue it *If the write was mistakenly written to a nondatabase file (for example, the write I/O was misdirected to the swap file) ▪ Exadata Cell is compliant with the HARD initiative, and implements all of the HARD checks. Unlike other implementations of HARD checking, HARD checks with Exadata Cell operate completely transparently. See Section 3.15 for details. ▪ For the most comprehensive lost write protection, use Oracle Data Guard and set either the DB_ULTRA_SAFE parameter (to DATA_ONLY or DATA_AND_INDEX) or set the DB_LOST_WRITE_PROTECT parameter (to TYPICAL or FULL) on both the primary and standby databases

Table 3–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Hangs or slow down	Oracle Database and Oracle Enterprise Manager	■ Oracle Database automatically monitors for database hangs and attempts to resolve them. ■ Oracle Enterprise Manager or a customized application heartbeat can be configured to detect application or response time slowdown and react to these SLA breaches. For example, you can configure the Enterprise Manager Beacon to monitor and detect application response times. Then, after a certain threshold expires, Enterprise Manager can call the Oracle Data Guard `DBMS_DG.INITIATE_FS_FAILOVER` PL/SQL procedure to initiate a failover. See the section about "Application Initiated Fast-Start Failover" in *Oracle Data Guard Broker*.

[1] Exadata Cell is compliant with the HARD initiative. In fact, Exadata Cell implements all of the HARD checks and because of its tight integration with Oracle Database, additional checks are implemented that are specific to Exadata Cell. Unlike other implementations of HARD checking, HARD checks with Exadata Cell operate completely transparently. You do not need to set parameters at the database or storage tier. The HARD checks transparently handle all cases, including Oracle ASM disk rebalance operations and disk failures.

3.2 Fast-Start Fault Recovery

Oracle Database provides fast and predictable recovery from system faults and database failures. The Fast-Start Fault Recovery technology included in Oracle Database automatically bounds instance recovery time at startup by using self-tuned checkpoint processing. This makes recovery time fast and predictable, and improves the ability to meet service-level objectives. The Oracle Fast-Start Fault Recovery feature can reduce recovery time on a heavily laden database from tens of minutes to a few seconds.

Fast-Start Fault Recovery features include:

- Predictable, bounded recovery from instance, database, and computer failures
- Database checkpointing that is self-tuning to maintain a desired recovery time objective

See Also: *Oracle Database Performance Tuning Guide*

3.3 Oracle Restart

Oracle Restart is a new feature in Oracle 11*g* Release 2 (11.2) that enhances the availability of a single-instance (nonclustered) Oracle database and its components. Oracle Restart is used in single-instance environments only. For Oracle Real Application Clusters (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware.

If you install Oracle Restart, it automatically restarts the database, the listener, and other Oracle components after a hardware or software failure or whenever the database's host computer restarts. It also ensures that the Oracle components are restarted in the proper order, in accordance with component dependencies.

Oracle Restart periodically monitors the health of components—such as SQL*Plus, the Listener Control utility (LSNRCTL), ASMCMD, and Oracle Data Guard—that are integrated with Oracle Restart. If the health check fails for a component, Oracle Restart shuts down and restarts the component.

Oracle Restart runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes.

See Also:

- *Oracle Database Administrator's Guide* for information about installing and configuring Oracle Restart
- The *Oracle Grid Infrastructure Installation Guide* for your platform

3.4 Oracle Real Application Clusters and Oracle Clusterware

Oracle RAC and Oracle Clusterware allow 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 Oracle Automatic Storage Management (Oracle ASM) to data management with Oracle Database and Oracle RAC

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.

3.4.1 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, as follows:
  - Upgrade Oracle Clusterware from Oracle Database 10*g* to Oracle Database 11*g*
  - Upgrade Oracle Clusterware from Oracle Database release 11.1 to release 11.2
  - Patch Oracle Clusterware from Oracle Database 11.1.0.6 to 11.1.0.7
  - Patch Oracle Clusterware from Oracle Database 10.2.0.2 Bundle 1 to Oracle Database 10.2.0.2 Bundle 2
- Automatically restarts failed Oracle processes.
- Automatically manages the virtual IP (VIP) address so when a node fails then 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 allows 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 Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*

3.4.2 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 3.4.1 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 (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.

3.5 Oracle Data Guard

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Oracle 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 disasters and data corruptions. Oracle Data Guard maintains standby databases as transactionally consistent copies of the primary (production) database. Then, if the primary database becomes unavailable because of a planned or an unplanned outage, Oracle Data Guard can switch any standby database to the primary role, minimizing the downtime associated with the outage. Oracle Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability.

With Oracle Data Guard, administrators can optionally improve primary database performance by off-loading resource-intensive backup and reporting operations to standby systems.

A Oracle Data Guard configuration consists of one primary database and one or more standby databases. Using a backup copy of the primary database, you can create up to 30 standby databases and incorporate them in a Oracle Data Guard configuration. After the database is created, Oracle Data Guard automatically maintains each standby database by transmitting redo data from the primary database and then applying the redo data to the standby database.

3.5.1 Types of Standby Databases

Similar to a primary database, a standby database can be either a single-instance Oracle database or an Oracle RAC database.

A standby database can be a physical standby database, a snapshot standby database, or a logical standby database, and a Oracle Data Guard configuration can include any combination of these types of standby databases: physical standby database, snapshot standby database, and logical standby database.

Physical Standby Database

A physical standby database provides a physically identical copy of the primary database, with data files that are identical to the primary database. The database schemas, including indexes, are the same. A physical standby database is kept synchronized with the primary database, through Redo Apply, which recovers the redo data received from the primary database and applies the redo data to the physical standby database.

You can use a physical standby database for business purposes other than disaster recovery. The physical standby database can be:

- Open a physical standby database for read-only access while redo data is being applied to the standby database. This mode is referred to as the **Active Data Guard option**[1], and allows users access to an up-to-date physical standby database.
- Use a physical standby database to offload the overhead of performing backups from the primary database. This is possible because a physical standby is an exact copy of the primary database.

Also, you can convert a physical standby database to:

- A logical standby temporarily, called a transient logical standby database, to perform a rolling upgrade.

  See Section 7.1.5.2, "Overview of Multiple Standby Database Architectures" on page 7-14, and the MAA white paper: "Database Rolling Upgrade Using Transient Logical Standby: Oracle Data Guard 11*g*" for more information at

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

- A snapshot standby database temporarily, to be used as a clone or a test database.

  See Section 7.1.5.2, "Overview of Multiple Standby Database Architectures" on page 7-14 for more information.

- Logical standby database

Snapshot Standby Database

A snapshot standby database is a physical standby database that is temporarily converted into an updatable standby database. A snapshot standby database receives and archives redo data from the primary database—protecting data on the primary database at all times—but the snapshot standby database does not apply redo data from the primary database while the standby is open for read/write access. Thus, the snapshot standby typically diverges from the primary database over time. Moreover, local updates to the snapshot standby database cause additional divergence.

Redo data from the primary database is not applied until you convert the snapshot standby database back into a physical standby database. With a single command, you can revert a snapshot standby to a physical standby database, at which time the changes made to the snapshot standby state are discarded, and Redo Apply

[1] Oracle Active Data Guard requires a license for the Active Data Guard option. The "real-time query" feature of Active Data Guard enables a physical standby database to be open read-only while Redo Apply is active.

automatically resynchronizes the physical standby database with the primary database using the redo data that was archived.

Logical Standby Database

A logical standby database contains the same logical information as the primary database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database through SQL Apply, which transforms the redo data received from the primary database into SQL statements and then executes the SQL statements on the standby database.

A key benefit of a logical standby database is that you can create significant auxiliary structures to optimize the reporting workload, including structures that could have a prohibitive effect on the transactional response time of the primary database. A logical standby database:

- Can have its data physically reorganized into a different storage type with different partitioning having many different indexes, and having on-demand refresh materialized views created and maintained. See *Oracle Database Concepts* for an overview of materialized views.
- Can be used to drive the creation of data cubes and other OLAP data views. See *Oracle OLAP Java API Developer's Guide* for more information.
- Can be used for other business purposes in addition to satisfying disaster recovery requirements, allowing users to access a logical standby database for queries and reporting purposes at any time.
- Can be used to upgrade Oracle Database software and patch sets with almost no downtime.

Thus, you can use a logical standby database concurrently for data protection, reporting, and database upgrades.

3.5.2 Benefits of Using Oracle Data Guard and Standby Databases

Oracle Data Guard provides the following overall benefits:

- Maintenance of real-time, transactionally consistent database copies to provide protection against unplanned downtime and disaster.
- Data protection against and fast repair of computer failures, human errors, data corruption, lost writes, and site failures.
- Automatic failover with flexible data protection levels to support all network configurations and business requirements.
- Faster redo application, redo transport, and role transitions with various enhancements.
- Reduction of planned downtime for system changes, some platform migrations, hardware and system upgrades, and Oracle patch set and database upgrades (see also Table 4–2).
- Multiple levels of data protection and performance to balance data availability against system performance requirements.
- Support for both physical standby databases (including the Active Data Guard option) and logical standby databases to provide more efficient use of system resources by diverting more querying and reporting functions from the primary database to standby databases (with the logical standby databases providing greater flexibility for any activity that requires access to a standby database that is

open for read/write access). See also the "Benefits of Physical Standby Databases" and "Benefits of Logical Standby Databases" on page 3-11.

- Support for the snapshot standby database for reporting or testing (cloning) purposes and automatic resynchronization with the primary database after reporting or testing has completed. See also "Benefits of Snapshot Standby Databases" on page 3-11.
- Support for automatic application notification so that application connections are seamless and fail over transparently.
- Automatic or automated resynchronization of a failed primary database following a failover.
- Management of all systems as a single configuration for simplified administration.
- Increased flexibility for Oracle Data Guard configurations where the primary and standby systems may have different CPU architectures, operating systems (for example, Windows and Linux), operating system binaries (32-bit and 64-bit), and Oracle database binaries (32-bit and 64-bit); this is subject to restrictions that are defined in support note 413484.1 at `http://metalink.oracle.com/`.

Benefits of Physical Standby Databases

- Guarantees a physical, block-for-block copy of the primary database
- Can be open for read-only queries while Redo Apply is active for real-time reporting (requires the Active Data Guard option that is described in Section 5.4.1)
- At role transition, offers assurance that the standby database is an exact replica of the old primary database
- Can be used to offload backups from primary database
- Provides very high performance, completely transparent to workload profile
- Has no data type restrictions
- Can be useful for minimizing downtime for many planned maintenance events

Benefits of Snapshot Standby Databases

- Inherits all the attributes of a physical standby database
- Can be open for read/write I/O and can process transactions independent of the primary database
- Protects the primary database the entire time it is open for read-write I/O
- Allows you to issue a single command to convert a Snapshot Standby back to a synchronized physical standby database
- Provides an ideal test system, especially when combined with Oracle Real Application Testing

Benefits of Logical Standby Databases

- Provides a logical, transaction-for-transaction copy of the primary database
- Allows creation of additional objects, modification of objects
- Provides the ability to skip apply on certain objects
- Supports real-time reporting
- Is open for read/write I/O (the data in tables that is maintained by SQL Apply cannot be changed)

- Minimizes downtime for software upgrades (see Oracle Data Guard Concepts and Administration for information about using SQL Apply to perform a rolling upgrade of Oracle Database software.)

3.6 Oracle Streams

Oracle Streams is a very flexible and powerful database feature that is used to implement fine-grained replication, multimaster replication, many-to-one replication, data transformation, hub-and-spoke replication, and message queuing.

Comparing Oracle Streams and Oracle Data Guard

Oracle Streams is designed for information sharing and it is the only technology that enables a globally distributed, update anywhere architecture. Oracle Streams enables highly customized replication strategies to satisfy the many varied uses of data replicated to a destination database. These same capabilities also make Oracle Streams a useful technology for addressing high availability and disaster recovery requirements and for minimizing planned downtime during upgrades to new database releases and patch sets.

Oracle Data Guard is designed for simple, one-way replication of an entire database expressly for maintaining a synchronized copy that can assume the primary role in a failure. Redo Apply (physical standby database) best exemplifies this notion of *simplicity*, as a disaster recovery solution that is both data type and application agnostic, and able to scale to very high levels of performance.

Although Oracle Data Guard also provides capabilities that enable a standby database to off-load from the primary database the overhead of performing backups, queries, and reports, these capabilities are ancillary to the primary mission of Oracle Data Guard, and are provided to increase your return on investment in high availability and disaster recovery.

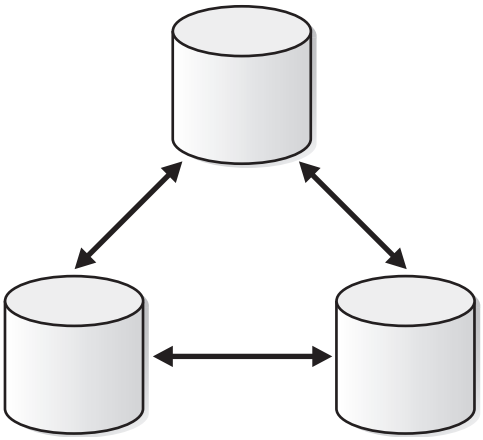
To get additional value from a Oracle Data Guard configuration, you can use SQL Apply (logical standby database) to minimize planned downtime during upgrades to new database releases and patch sets.

Oracle Streams Messaging and Information Flow

Oracle Streams enables information sharing. In Oracle Streams, each unit of shared information is called a message, and you can share these messages in a stream. The stream can propagate information within a database or from one database to another.

For example, Figure 3–1 shows a Oracle Streams multimaster configuration where all sites are directly connected to all other sites participating in the replication environment. The multimaster configuration enables data to be replicated between all locations at a rate that is nearly matches the real-time rate of replication.

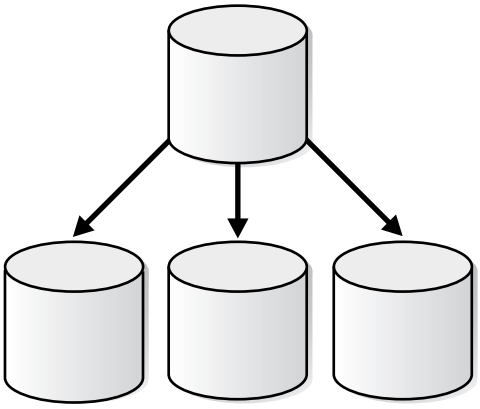
Figure 3–1 Oracle Streams Multimaster Configuration



Another example is the Oracle Streams 1-N, or *hub-and-spoke configuration*, in which changes made at the primary (hub) location are propagated to the remote (spoke) locations in a near real-time manner.

Although it is possible to configure a hub-and-spoke configuration for bidirectional replication, you may prefer to restrict updates to a single location—the hub—as shown in Figure 3–2. In query-intensive environments, you can still balance the load between multiple locations, with fast local access, whereas updates are restricted to the hub. By off-loading reporting to the spoke locations, you improve performance at the hub, or primary OLTP location. This type of configuration is easier to implement than multimaster replication because it is not necessary to establish connectivity between all locations in the replication environment and it is not necessary to implement a conflict resolution strategy.

Figure 3–2 Information Dissemination with Oracle Streams (1-N Configuration)

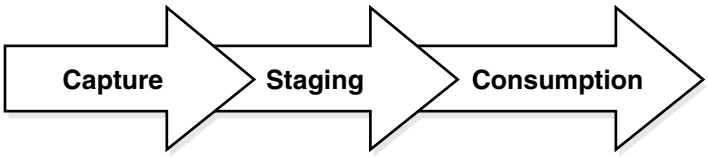


The stream routes specific information to specific destinations. The result is a feature that provides greater functionality and flexibility than traditional solutions for capturing and managing messages and sharing the messages with other databases and applications. Oracle Streams provides the capabilities needed to build and operate distributed enterprises and applications, data warehouses, and high availability solutions. You can use all of the capabilities of Oracle Streams at the same time. If your business requirements change, then you can implement a new capability of Oracle Streams without sacrificing existing capabilities.

Every Oracle Streams configuration has three phases: capture, stage (propagate), and consume (apply). Using Oracle Streams, you control what information is put into a stream, how the stream flows or is routed from database to database, what happens to messages in the stream as they flow into each database, and how the stream terminates. By configuring specific capabilities of Oracle Streams, you can address specific requirements. Based on your specifications, Oracle Streams can capture, stage, and manage messages in the database automatically, including, but not limited to, data manipulation language (DML) changes and data definition language (DDL) changes. You can also put user-defined messages into a stream, and Oracle Streams can propagate the information to other databases or applications automatically. When messages reach a destination, Oracle Streams can consume them based on your specifications.

Figure 3–3 shows the Oracle Streams information flow.

Figure 3–3 Oracle Streams Information Flow

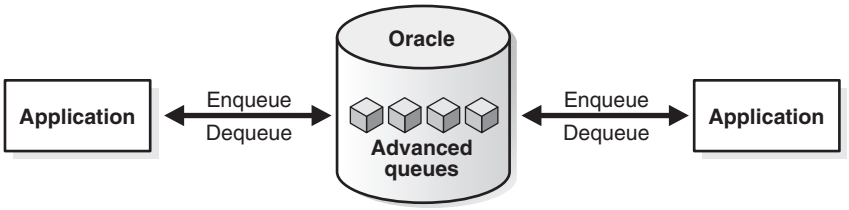


With Oracle Streams, you can create a local or remote copy of a production database. If a human error or catastrophe occurs, you can use the copy to resume processing. You can use Oracle Streams to configure flexible high availability environments.

You can use the features of Oracle Streams to achieve little or no database downtime during database upgrade and maintenance operations. Maintenance operations include migrating a database to a different platform, migrating a database to a different character set, modifying database schema objects to support upgrades to user-created applications, and applying an Oracle software patch.

Figure 3–4 shows an application that explicitly enqueues and dequeues messages through Oracle Streams Advanced Queuing as a method of sharing information with business partners or customers with different messaging systems. After it is enqueued, messages can be transformed and propagated as desired, before being dequeued to a business partner's application that is a nondatabase-oriented messaging system.

Figure 3–4 Oracle Streams Message Queuing



Benefits of Using Oracle Streams

Oracle Streams provides the following benefits:

- Data protection by maintaining a full or partial remote copy of the database
- Little or no downtime during database upgrade or maintenance operations such as migrating a database to a different platform or character set, modifying database objects to support upgrades to applications, and applying an Oracle software patch

- Data replication by capturing DML and DDL changes made to database objects and replicating these changes to one or more other databases

  A bidirectional replication environment, in which exactly two databases share the replicated database objects and data, is possible.

- Event management and notification by enqueuing messages or capturing events, propagating the messages and events through queues, and dequeuing and applying or acting upon the message or event (as shown in Figure 3–4)
- Heterogeneous platform support across databases in the configuration
- Character sets that can differ between replicas
- Fine-grained control of data sharing

See Also:

- *Oracle Streams Concepts and Administration* for Oracle Streams concepts
- *Oracle Streams Replication Administrator's Guide* for configuring hub-and-spoke replication environments
- *Oracle Database 2 Day + Data Replication and Integration Guide* for information about replicating data continuously between databases

3.7 Oracle Flashback Technology

Flashback technology provides a set of features to switch between views of the data as it existed at different points in time. Using flashback features, you can query past versions of schema objects and historical data. You can also perform change analysis and self-service repair to recover from logical corruption while the database is online.

Flashback technology provides a SQL interface to quickly analyze and repair human errors. Flashback technology provides fine-grained analysis and repair for localized damage such as deleting the wrong customer order. Flashback technology also enables correction of more widespread damage, yet does it quickly to avoid long downtime. Flashback technology is unique to Oracle Database and supports recovery at all levels including row, transaction, table, tablespace, and database.

Most of the flashback features use undo data, whereas other features (such as Flashback Database and Block Media Recovery) use flashback logs:

- Undo tablespace—A dedicated tablespace that stores only undo information when the database is run in automatic undo management mode.
- Flashback Data Archive—An archive that 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 much longer duration than the retention period offered by an undo tablespace.
- Flashback logs—Oracle-generated logs used to perform Flashback Database or block media recovery operations. The database can only write flashback logs to the fast recovery area. Flashback logs are written sequentially and are not archived. They cannot be backed up to disk.

The following sections describes the Flashback features:

3.7.1 Oracle Flashback Query

Oracle Flashback Query provides the ability to view the 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 have been deleted or updated can be immediately repaired even after they have been committed.
- Comparing current data with the corresponding data at some time in the past. For example, 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. Using a 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 Advanced Application Developer's Guide*

3.7.2 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 in 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 (DBA) 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 Advanced Application Developer's Guide*

3.7.3 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 has been 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 Advanced Application Developer's Guide*
- *Oracle Database PL/SQL Packages and Types Reference*

3.7.4 Oracle Flashback Transaction Query

Oracle Flashback Transaction Query provides a mechanism to view all 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 Advanced Application Developer's Guide*

3.7.5 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 has been modified 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*

3.7.6 Oracle Flashback Drop

Oracle Flashback Drop

Dropping objects by accident is a problem for database users and database administrators alike. 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*

3.7.7 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 timestamp—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 Recovery Manager (RMAN) operations. Furthermore, a database can be flashed back through a previous database recovery and open resetlogs by using guaranteed restore points. Guaranteed restore points allow major database changes—such as database batch jobs, upgrade, or patch—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:

- Provides 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)
- Allows the snapshot standby to be resynchronized with the production database
- Serves as a quick mechanism to restore a test or cloned database to its original state

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

3.7.8 Oracle Flashback Database

Oracle Flashback Database provides a more efficient alternative to database point-in-time recovery. With Oracle Flashback Database, you can revert current data files to their contents at a past time. The result is much like restoring data from data file backups and executing point-in-time database recovery. However, Flashback Database skips the data file restoration and most of the application of redo data.

Enabling Oracle Flashback Database provides the following benefits:

- Eliminates the time to restore a backup when fixing human error that has a database-wide impact.
- Because human errors can be quickly undone, it allows standby databases to use real-time apply to synchronize with the primary database.
- Allows quick standby database reinstantiation after a database failover.

See Also:

- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Database SQL Language Reference*

3.7.9 Block Media Recovery Using Flashback Logs

After attempting automatic block repair, 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 `V$DATABASE_BLOCK_CORRUPTION` table. You can subsequently issue the RMAN `RECOVER BLOCK` command to fix the associated block. In addition, the RMAN `RECOVER BLOCK` command restores blocks from a physical standby database, if it is available.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for block media repair
- *Oracle Database Backup and Recovery Reference* for the RMAN `RECOVER BLOCK` command
- Section 3.18, "Automatic Block Repair"

3.7.10 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.

See Also: *Oracle Database Advanced Application Developer's Guide*

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 as it can mirror at the database file level rather than the entire disk level.

More importantly, Oracle ASM simplifies the processes of setting up mirroring, adding disks, and removing disks. Instead of managing hundreds and 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.

Oracle ASM provides the following benefits:

- Provides the ability to mirror and stripe across drives and storage arrays
- Automatically remirrors from a failed drive to remaining drives
- Automatically rebalances stored data when disks are added or removed while the database remains online
- Supports Oracle database files and non-database files using Automatic Storage Management File Systems (ASMFS).
- Allows for operational simplicity in managing database storage
- Manages the Oracle Cluster Registry (OCR) and voting disks
- Provides preferred read capability on disks that are local to the instance, which gives better performance for an extended cluster

- Supports very large databases
- Supports Oracle ASM rolling upgrades
- Supports finer granularity in tuning and security
- Provides fast repair after a temporary disk failure through Oracle ASM Fast Mirror Resync and automatic repair of block corruptions if good copy exists in one of the mirrors

See Also: *Oracle Database Storage Administrator's Guide*

3.9 Fast Recovery Area

The **fast recovery area**, previously referred to as a flash 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 importantly, 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
- Ability to back up and restore the entire fast recovery area
- Ability to tolerate failures to the fast recovery area

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

3.10 Recovery Manager

RMAN is an Oracle 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:

- 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 resetlogs
- 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 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 restore of required files only
- Retention policy to ensure that relevant backups are retained
- Ability to resume backup and restore 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

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

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, and snapshot standby databases
- Physical standby database and Oracle RAC databases

  Data Recovery Advisor only takes the presence of a physical standby database into account when recommending repair strategies.

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 has been diagnosed correctly, selecting the right 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.

- 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.

See Also: "Diagnosing and Repairing Failures with the Data Recovery Advisor" in *Oracle Database Backup and Recovery User's Guide*

3.12 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, supporting releases since Oracle9*i*. 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 tape backup for Oracle Database by backing up only the currently used blocks and increasing backup performance by 10% to 25%

- 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 allowing 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
- 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

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

3.13 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 truly needed 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 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.

See Also:

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

3.14 LogMiner

Oracle log files contain useful information about the activities and history of Oracle Database. Log files contain all data necessary to perform database recovery, and also record all changes made to the data and metadata in the database.

LogMiner is a fully relational tool that allows redo log files to be read, analyzed, and interpreted using SQL. Using LogMiner, you can analyze log files to:

- Track or audit changes to data
- Provide supplemental information for tuning and capacity planning
- Retrieve critical information for debugging complex applications
- Recover deleted data
- Provide additional browser-based simplification to help troubleshoot and resolve logical failures

LogMiner features include:

- Pinpointing when a logical corruption to the database—such as errors made at the application level—may have occurred
- Determining the necessary actions to perform fine-grained recovery at the transaction level
- Providing performance tuning and capacity planning through trend analysis
- Auditing

See Also: *Oracle Database Utilities*

3.15 Oracle Exadata Storage Server Software (Exadata Cell)

Oracle Storage Grid using Exadata. **Oracle Exadata Storage Server Software** is a storage product that is highly optimized for use with Oracle Database. Oracle Exadata Storage Server Software, also referred to as Exadata Cell, is used to store and access Oracle Database. It runs the Exadata Cell software. It can be used in addition to traditional storage arrays and products. Exadata Cell provides database-aware storage services, such as the ability to offload database processing from the database server, while remaining transparent to SQL processing and database applications.

The Oracle Storage Grid is implemented using either Oracle ASM and Oracle Exadata Storage Server Software or Oracle ASM and third-party storage. The Oracle Storage Grid with Exadata provides seamless support for MAA-related technology, improves performance, provides unlimited I/O scalability, is straightforward to use and manage, and delivers mission-critical availability and reliability to your enterprise. Also, Exadata Cell is the most comprehensive solution, to prevent corruptions from being written to disk, and it is HARD compliant. Also, Exadata Cell with the `DB_ULTRA_SAFE=DATA_AND_INDEX` initialization parameter provides the most comprehensive solution to prevent corruptions from being written to disk, and Exadata Cell is HARD complaint. For HARD compliance, you will require a minimum setting of `DB_BLOCK_CHECKSUM=TYPICAL`.

Note: Exadata Cell is compliant with the Hardware Assisted Resilient Data (HARD) initiative. In fact, Exadata Cell implements all of the HARD checks and because of its tight integration with Oracle Database, additional checks are implemented that are specific to Exadata Cell. Unlike other implementations of HARD checking, HARD checks with Exadata Cell operate completely transparently. You do not need to set parameters at the database or storage tier. The HARD checks transparently handle all cases, including Oracle ASM disk rebalance operations and disk failures.

See Also:

- *Oracle Database High Availability Best Practices* to learn about the best practice recommendations for Oracle Storage Grid
- The HP Oracle Exadata Storage Server Web site at `http://www.oracle.com/technology/products/bi/db/exadata/index.html`

3.16 Oracle Database File System (DBFS)

Oracle Database File System (DBFS) creates a standard file system interface on top of files and directories that are stored in database tables. Oracle DBFS is similar to the Network File System (NFS) protocol in that it provides a shared network file system that looks like a local file system.

Similar to NFS, there is a server component and a client component. The server is Oracle Database and files are stored as SecureFile LOBs in a database table. Because the files are stored in the database you inherit the high availability and disaster-recovery protection Oracle Database offers, providing a full stack disaster-recovery solution. The implementation of the file system in the database is called the DBFS Content Store and allows each database user to create one or more file systems that can be mounted by clients. Each file system has its own dedicated tables that hold the file system content. A set of PL/SQL procedures implement file system access (such as `CREATE`, `OPEN`, `READ`, `WRITE`, and `LIST DIRECTORY`) to the database.

Oracle DBFS provides the following benefits:

- The ability to store both non-structured and structured data in the same database.

  This allows you to perform backups and synchronous point-in-time recovery of both types of data.

  Oracle DBFS provides the ability to store unstructured content in the database by presenting an NFS-like file system to the client. The file system itself is stored in a tablespace in Oracle Database. The database storage aspect is transparent to the client because it appears as a traditional NFS mounted file system with the same functionality, but DBFS provides the ability to store any type of file directly in the database—such as logs or generated reports—that you would normally store in a file system.

- Clustered file system capability with a lightweight process.

  You can mount Oracle DBFS on multiple client machines (database servers, mid-tiers) and therefore the file system can also be available for use as a clustered file system. A lightweight process is started on each client machine to make the file system accessible. This process uses the FUSE (Filesystem in Userspace) API to implement the file system access.

- Fast and transparent client failover of both file system and database operations (full stack disaster recovery).

  The process on the client systems is OCI based. Thus, clients can take advantage of FAN and Fast Connection Failover capabilities using the same service-based connection methods.

  See Also: *Oracle Database SecureFiles and Large Objects Developer's Guide* for more information about Oracle DBFS

3.17 Client Failover

A highly available architecture must achieve fast database and client failover:

- Database failover to a designated, synchronized standby database must occur quickly, and reliably in the event of loss of the primary database.
- Client failover must enable middle-tier applications (or any client program that connects directly to a database) to quickly and seamlessly fail over to an available database service when the primary database service is unavailable.

Client failover encompasses failure notification, previous connection cleanup, automatic reconnection, and possible query replay. Until Oracle Database 10g Release 2 (10.2), automatic, fast, and transparent client failover (at the session level) has been difficult to achieve for all client types and for all failures.

Client failover provides the capability to integrate automatic database failover with failover procedures at the middle tier to automatically redirect clients and applications to the new primary database at the standby location within seconds of failover, providing an end-to-end solution for achieving business continuity.

See Also: The MAA white paper "Client Failover in Data Guard Configurations for Highly Available Oracle Databases Updated" at `http://www.otn.oracle.com/goto/maa`

3.18 Automatic Block Repair

Automatic block repair allows corrupt data blocks to be automatically repaired as soon as the corruption is detected. This feature reduces the amount of time that data is inaccessible due to block corruption. This reduces block recovery time by using up-to-date good blocks in real-time, as opposed to retrieving blocks from disk or tape backups, or from Flashback logs.

Table 3–2 Automatic Detection and Repair of Corrupt Data Blocks

IF ...	THEN ...
A corrupt data block is discovered on a primary database	A physical standby database operating in real-time query mode can be used to repair corrupt data blocks in a primary database. If possible, any corrupt data block encountered when a primary database is accessed is automatically replaced with an uncorrupted copy of that block from a physical standby database operating in real-time query mode. An `ORA-1578` error is returned when automatic repair is not possible.

Table 3–2 (Cont.) Automatic Detection and Repair of Corrupt Data Blocks

IF ...	THEN ...
A corrupt data block is discovered on a physical standby database	The server attempts to automatically repair the corruption by obtaining a copy of the block from the primary database if the following database initialization parameters are configured on the standby database: ■ Configure the LOG_ARCHIVE_CONFIG parameter with a DG_CONFIG list ■ Configure a LOG_ARCHIVE_DEST_*n* parameter for the primary database

You can also manually repair a corrupted data block by using the RMAN RECOVER BLOCK command. This command searches several locations for an uncorrupted copy of the data block. By default, one of the locations is any available physical standby database that is operating in real-time query mode. You can use the EXCLUDE STANDBY option of the RMAN RECOVER BLOCK command to exclude physical standby databases as a source for replacement blocks.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for information about block media recovery
- *Oracle Database Backup and Recovery Reference* for the RMAN RECOVER BLOCK command
- *Oracle Data Guard Concepts and Administration* for a description of automatic block repair using real-time query standby database

3.19 Corruption Prevention, Detection, and Repair

Starting in Oracle Database 11*g* Release 2 (11.2), the primary database automatically attempts to repair the corrupted block in real time by fetching a good version of the same block from a physical standby database.

Also, Exadata Cell is the most comprehensive solution to prevent corruptions from being written to disk, and it is Hardware Assisted Resilient Data (HARD) compliant. HARD uses block checking where the storage subsystem validates the Oracle block contents, and it can detect corruption early and prevent corrupted data from being written to disk. See Section 3.15 for more information about Exadata Cell and HARD.

Before Oracle Database 11*g*, block corruptions detected by RMAN were recorded in V$DATABASE_BLOCK_CORRUPTION. Beginning with Oracle Database 11*g*, several database components and utilities in addition to RMAN can detect a corrupt block and record it in that view. Oracle Database automatically updates this view when block corruptions are detected or repaired (for example, using block media recovery or data file recovery). Block corruptions are now discovered sooner.

You must use the DB_ULTRA_SAFE initialization parameter to automatically configure the appropriate data protection block checking level in the database. The performance impact may vary depending on the application and available system resources, but the effect can vary from 1% to 10%.

The DB_ULTRA_SAFE initialization parameter:

- Controls the setting of other related initialization parameters, including DB_BLOCK_CHECKING, DB_BLOCK_CHECKSUM, and DB_LOST_WRITE_PROTECT

- Controls other data protection behavior in Oracle Database, such as requiring Oracle ASM to perform sequential mirror writes

By making it possible to detect data corruptions in a timely manner, these features provide critical high availability benefits for Oracle Database.

> **See Also:** *Oracle Database Reference* for more information about these views and initialization parameters

4

Oracle Database High Availability Solutions for Planned Downtime

Planned downtime can be just as disruptive to operations as unplanned downtime. This holds 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, seven days a week.

In the past, planned downtime was necessary to perform periodic maintenance or to upgrade to new deployments including:

- 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.

Table 4–1 shows the high availability solutions to eliminate or reduce planned downtime.

Table 4–1 High Availability Solutions to Reduce Planned Downtime

Solution	Reduces Planned Downtime Due to ...
Online patching, rolling upgrades, and migrations (planned maintenance topics in Section 4.1)	System, clusterware, operating system, and database upgrades
Dynamic Resource Provisioning	System and database changes
Online Reorganization and Redefinition	Data changes
Transportable Technologies	Database migration to a new platform
Online Application Maintenance and Upgrades	Application changes

Section 4.1 summarizes Oracle's high availability solutions that prevent, tolerate, and reduce downtime for all types of planned maintenance.

4.1 Oracle High Availability Solutions and Recovery Times for Planned Downtime

Oracle provides high availability solutions to prevent, tolerate, and reduce downtime for all types of planned maintenance. Table 4–2 describes the various Oracle high availability solutions for planned downtime, along with the outage time that can be attained with each solution. In all cases, Oracle recommends that you extensively test before performing any rolling upgrade.

See Also: Table 7–5 on page 7-30 for a summary of the attainable recovery times for all types of planned downtime for each Oracle high availability architecture

Table 4–2 Oracle High Availability Solutions for Planned Downtime

Maintenance Type	Oracle Recommended Solution	Solution Description	Outage Time
Operating system and hardware upgrades	Oracle Real Application Clusters and Oracle Clusterware	Section 4.1.1	No downtime
Oracle interim patches	Oracle Real Application Clusters (Oracle RAC)	Section 4.1.3	No downtime[1]
Online patches	Online Patching	Section 4.1.4	No downtime
Oracle Clusterware upgrades and patches	Oracle Clusterware	Section 4.1.5	No downtime
Oracle ASM upgrades	Oracle Automatic Storage Management	Section 4.1.6	No downtime
Storage migration[2]	Oracle Automatic Storage Management	Section 4.1.7	No downtime
Migrating to Exadata Storage	Oracle MAA best practices discussed in the "Best Practices for Migrating to Oracle Exadata Storage Server" white paper	Section 4.1.8	Outage time depends on solution chosen
Upgrading Exadata Storage	The Exadata Patch Manager	Section 4.1.9	No downtime
Migrating a single-instance database to Oracle RAC	Oracle Clusterware	Section 4.1.1	No downtime
Migrating to Oracle ASM or migrating a single-instance database to Oracle RAC	Oracle Data Guard	Section 4.1.2	Seconds to minutes
Patch set and database upgrades	Oracle Data Guard using SQL Apply	Section 4.1.10	Seconds to minutes
Platform migration across Windows and Linux platforms and other select platforms[3]	Oracle Data Guard	Section 4.1.10	Seconds to minutes
Platform Migration across the same endian format platforms	Transportable database	Section 4.1.11	Minutes to hours
Platform migration across different endian format platforms	Transportable tablespaces	Section 4.1.12	Minutes to hours
Patch set and database upgrades, platform migration, rolling upgrades, and when different character sets are required	Oracle Streams	Section 4.1.10, Section 4.1.11, Section 4.1.12, and Section 4.5	Seconds to minutes
Application upgrades	Online Application Maintenance and Upgrades	Section 4.5	No downtime

[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.

[2] An example is migration from traditional storage to low-cost storage.

[3] See My Oracle Support (formerly Oracle*Metalink*) Note 413484.1 at `http://metalink.oracle.com/`.

See Also:

- *Oracle Data Guard Concepts and Administration* for more information about using Oracle Data Guard with SQL Apply to upgrade an Oracle database
- *Oracle Database Concepts* and the *Oracle Database Administrator's Guide* for more information about transportable tablespaces
- The MAA white papers about rolling upgrade best practices at

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

4.1.1 Operating System Upgrades and Hardware Upgrades

Using Oracle RAC is the recommended solution for avoiding downtime during system and hardware upgrades.

If you cannot perform the upgrade using Oracle RAC, then the recommended solution is to use Oracle Data Guard and physical standby databases as described in Section 4.1.2. Alternatively, you can use cold cluster failover with Oracle Clusterware as described in Section 4.1.5.

Oracle RAC Solution Description

To perform upgrades using Oracle RAC:

1. 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).
2. Shut down destination instance or instances with the `IMMEDIATE` option.
3. Shut down and disable Oracle Clusterware.

   Disabling Oracle Clusterware prevents it from starting automatically.
4. Perform maintenance.
5. Enable and start Oracle Clusterware.

   This step implicitly starts the database instances.
6. Start the application service.

   This step implicitly redirects connections to the destination instance when using FAN.
7. Repeat all steps on the next node.

Additional Considerations

Verify the following:

- Ensure that the planned maintenance can be done in a rolling fashion from an operating system perspective.
- Ensure that the database and clusterware versions are certified with the new system and hardware changes.

  See Also: Your operating system-specific Oracle Real Application Clusters installation guide

4.1.2 System and Cluster Upgrades and Migrations Using Oracle Data Guard

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

In general, you first upgrade the physical standby database and then perform an Oracle Data Guard switchover to 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 impact 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 an Oracle 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 an Oracle Data Guard switchover to return the standby database to the primary database role.

Additional Considerations

- For fastest switchover, configure the standby database to use real-time apply and, if possible, ensure that the databases are synchronized before the switchover operation.
- Use Oracle 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.
- The 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 additional post-upgrade steps that are described in the My Oracle Support Note 414043.1 at `http://metalink.oracle.com/`. Also, see the *Oracle Database Upgrade Guide* for more information about upgrades.

4.1.3 Oracle Interim Database Patches

Use Oracle RAC to avoid downtime when applying Oracle interim database patches. You can apply approximately 90% of the new patches using Oracle RAC.

If you cannot apply patches using Oracle RAC, then use Oracle Data Guard and physical standby databases. See Section 4.1.2 for more information.

Solution Description

Oracle interim (one-off) 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.

Oracle provides the capability to do rolling patch upgrades with Oracle RAC with little or no database downtime using the OPatch command-line utility.

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 outages. The Oracle OPatch utility enables you to apply the patch successively to the different instances in an Oracle RAC installation.

Additional Considerations

Performing a rolling upgrade is possible only for patches that are certified for rolling upgrades. Typically, patches that can be installed in a rolling upgrade include:

- Patches that do not affect the contents of the database, such as the data dictionary
- Patches not related to Oracle RAC internode communication
- Patches related to client-side tools such as SQL*Plus, Oracle Database utilities, development libraries, and Oracle Net
- Patches that do not change shared database resources, such as data file headers, control files, and common header definitions of kernel modules

Do not use Oracle RAC to perform rolling upgrades of patch sets.

> **See Also:** Your operating system-specific Oracle Real Application Clusters installation guide

4.1.4 Online Patching

Online patching is the recommended solution for avoiding downtime when an online patch is available for debug patches and interim patches.

Solution Description

Online patches are a special type of interim patch that you can apply while the instance remains online.

Oracle provides the capability to perform online patching with any Oracle database using the OPatch command-line utility.

Additional Considerations

- Oracle provides combo patches, which are online patches that can also be applied when the instance is offline.

  Thus, you can apply the online patch initially to avoid unplanned downtime. However, because online patches have a memory overhead, you should roll back the online patch and apply the offline patch during schedule downtime.
- Oracle provides certified online patches for diagnostic patches or various bug fixes.

- Oracle provides online patches when the patch does not change shared memory structures in the System Global Area (SGA), or other critical internal code structures.
- 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:

- *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX* for information about online patching and the OPatch utility
- *Oracle Database Upgrade Guide* for an overview of rolling upgrades and rolling patches

4.1.5 Upgrading Oracle Clusterware

Performing rolling upgrades of Oracle Clusterware is the recommended solution for avoiding downtime when upgrading Oracle Clusterware.

Solution Description

You can perform all upgrades to Oracle Clusterware in a rolling fashion.

See Also: Your operating system-specific Oracle Clusterware installation guide

4.1.6 Upgrading Oracle Automatic Storage Management (Oracle ASM)

Performing rolling upgrades is the recommended solution for upgrading Oracle ASM.

Solution Description

You can perform all upgrades starting with Oracle Database 11*g* (and later releases) in a rolling fashion.

See Also: *Oracle Database Storage Administrator's Guide*

4.1.7 Storage Migration

Using Oracle ASM is the recommended solution for performing storage migrations.

Solution Description

Oracle ASM enables you to add all disks in one storage array and subsequently drop all disks from another array. Oracle ASM automatically rebalances and migrates data to the new storage while the database remains operational.

Additional Considerations

Before removing the source storage array, ensure that the rebalancing is complete.

See Also: The chapter about performing Oracle ASM Data Migration in the *Oracle Database Backup and Recovery User's Guide*

4.1.8 Migrating Oracle Exadata Storage Server Software

The guidelines in the MAA white paper "Best Practices for Migrating to Oracle Exadata Storage Server" define best practices for pre-migration and post migration from legacy storage to Oracle Exadata Storage Server. The best practices help you determine the most appropriate migration strategy given the application service levels and attributes.

The MAA white paper is available at http://www.oracle.com/technology/products/bi/db/exadata/pdf/migration-to-exadata-whitepaper.pdf.

Solution Description

This section summarizes the steps described in the "Best Practices for Migrating to Oracle Exadata Storage Server" MAA white paper.

To migrate Oracle Exadata Cell deployments:

Pre migration:

1. Perform capacity planning
2. Employ Exadata configuration automation
3. Ensure the Infiniband network is used for Oracle Clusterware and Oracle RAC communication
4. Ensure proper Oracle ASM software and disk group attributes
5. Ensure proper database software and compatibility
6. Enable the `DB_BLOCK_CHECKSUM` initialization parameter

Migration:

1. Configure the Oracle ASM allocation unit size to 4 MB
2. Configure optimal database extent sizes
3. Choose to migrate to Oracle Exadata Storage Server either logically or physically:
   - Logical migrations: database extent size changes
   - Physical migrations: no database extent size change
4. Create new tablespaces on Oracle Exadata Storage Server

Post-Migration:

1. Check disk groups for rebalance
2. Assess index requirements

> **See Also:** The HP Oracle Exadata Storage Server Web site at http://www.oracle.com/technology/products/bi/db/exadata/index.html

4.1.9 Upgrading Oracle Exadata Storage Server Software

See the Oracle Exadata Storage Server Software documentation to learn about the solutions and tools used to perform upgrades.

See Also:

- My Oracle Support (formerly Oracle*Metalink*) Note 791275.1 at `http://metalink.oracle.com/` that includes:

  Oracle Exadata Storage Server Software Documentation Addendum

  Oracle Exadata Storage Server Software Planning and Deployment Guide

  Oracle Exadata Storage Server Software Patch Application Example

- The HP Oracle Exadata Storage Server Web site at `http://www.oracle.com/technology/products/bi/db/exadata/index.html`

4.1.10 Patch Set and Database Upgrades

Oracle Data Guard using SQL Apply is the recommended solution for performing patch set and database upgrades with minimal downtime. Section 4.1.10.1 describes this solution. 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 Database Upgrade Assistant (DBUA)[1], transportable tablespace, or Oracle Streams:

- 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 transportable tablespaces to perform a database upgrade in less than one hour.
- Transportable tablespaces is the solution if you cannot use SQL Apply but the maintenance window requires downtime to be less than an 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 4.1.10.2 describes the transportable tablespace solution.
- Oracle Streams is the solution that provides the most flexibility when performing database upgrades and additional data type support. Section 4.1.10.3 describes this solution.

See Also: *Oracle Database High Availability Best Practices* for more information and for help choosing the database upgrade method appropriate for your configuration

4.1.10.1 Solution for Database Upgrades Using Data Guard and SQL Apply

To upgrade an Oracle database using SQL Apply:

1. Upgrade logical standby database to the new release and evaluate the change.
2. Ensure that SQL Apply has applied all redo data to the logical standby database.
3. Disconnect applications.

[1] 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.

4. Perform an Oracle Data Guard switchover.
5. Reconnect applications to the new primary database.
6. Shut down the original primary database (now the logical standby database).
7. Execute database software upgrade steps on the new standby database.
8. Restart the standby database and allow recovery to synchronize.
9. Optionally, perform an Oracle Data Guard switchover to return to the original database.

Additional Considerations

- SQL Apply rolling upgrades are only supported for Oracle Database release 10.1.0.3 and later. For complete information, see the chapter about using SQL Apply to upgrade Oracle Database in *Oracle Data Guard Concepts and Administration*.
- SQL Apply has some data type restrictions (see *Oracle Data Guard Concepts and Administration* for a list of the restrictions). If there are data type restrictions, consider implementing Extended Datatype Support (EDS).

  EDS enables SQL Apply to replicate changes to tables that contain some data types not natively supported from one database to another. Beginning with Oracle Database 10*g* Release 2 (10.2.0.4) Patch Set 3, SQL Apply supports the ability for triggers to fire on the logical standby database, which provides the basis of EDS. For an overview of EDS, see the MAA white paper "Extended Datatype Support" available at `http://www.otn.oracle.com/goto/maa`.

  For examples using EDS to support data types that are not natively supported by SQL Apply, see support note 559353.1 at `http://metalink.oracle.com/`.
- Beginning with Oracle Database 11*g* release 11.1, you can use a physical standby database to execute a rolling database upgrade using the `KEEP IDENTITY` clause and a transient logical standby database.
- Oracle Data Guard is the best approach if performing an Oracle RAC rolling upgrade is not possible and there are no data type restrictions.

  See Also:

  - *Oracle Data Guard Concepts and Administration*
  - The following MAA white papers available at `http://www.otn.oracle.com/goto/maa`:

    "Database Rolling Upgrade Using Data Guard SQL Apply"

    "Database Rolling Upgrade Using Transient Logical Standby"

4.1.10.2 Solution for Database Upgrades Using Transportable Tablespaces

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 transportable tablespaces to upgrade your database.

To use the transportable tablespaces 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. Stop Redo Apply and shut down the standby database.

c. Transport the user tablespaces.

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

See Also: The MAA white paper "Database Upgrade Using Transportable Tablespaces" available at `http://www.otn.oracle.com/goto/maa`

Additional Considerations

- The transportable tablespace feature is an option for performing a database upgrade in less than one 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 Tablespaces" available on the MAA Web site at

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

- Using transportable tablespaces 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 tablespaces, 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.

4.1.10.3 Solution Description for Database Upgrades Using Oracle Streams

Oracle Streams is similar in function to Oracle Data Guard SQL Apply. Like SQL Apply, Oracle Streams can use Extended Datatype Support (EDS) to replicate changes to tables that contain some data types not natively supported from one database to another.

To perform a database upgrade using Oracle Streams:

1. Before you begin the upgrade process, see *Oracle Streams Concepts and Administration* for information about how to perform a database upgrade on a database that has user-defined types.

2. Create a duplicate database. (The ideal replica will begin as a physical standby database that is up-to-date.)

3. Activate and upgrade the database to the later version.

4. Enable Oracle Streams replication.

5. During the upgrade of the replica, the source database continues ahead. After the replica is caught up, perform a switchover.

See Also:

- *Oracle Streams Concepts and Administration* for complete information about performing an online database upgrade with Oracle Streams
- *Oracle Database Backup and Recovery User's Guide* to learn about duplicating a database

4.1.11 Platform Migration Across the Same Endian Format Platforms

Consider the following approaches when you perform platform migrations across the same endian format platforms:

- Oracle Data Guard (physical standby database) is the recommended solution for performing platform migration across Linux and Windows platforms. Section 4.1.2 on page 4-4 describes this solution.
- If cross-platform physical standby database is not available for the platform combination to be migrated, then use the transportable database feature. Section 4.1.11.1 on page 4-11 describes this solution.
- If the transportable database feature cannot perform the migration quickly enough, then use Oracle Streams. Section 4.1.11.2 on page 4-12 describes this solution.

4.1.11.1 Solution Description for Platform Migration Using Transportable Database

Use transportable database for platform migration only when cross-platform physical standby database or logical standby database is not supported for the platform combination in question[2].

For example, to move from Windows x86-64 to Linux x86-64, it is best to use a cross-platform standby database instead of transportable database. There is less downtime (only the time it takes to switch over) and it is possible to run the standby database on the new platform temporarily to ensure that everything is working as planned.

To perform a platform migration using transportable database (with destination system conversion):

The high-level steps are as follows:

1. Place the source database in read-only mode.
2. Run the RMAN `CONVERT DATABASE` command.
3. Move files to the destination system.
4. Run RMAN generated script to convert data files with undo data to destination platform format.
5. Run RMAN generated script to complete the migration.

When using transportable database, the downtime required for a platform migration is determined by the time needed to:

- Place the source database in read-only mode

[2] Beginning with Oracle Database 11*g*, the primary and standby systems in an Oracle Data Guard configuration can have different CPU architectures, operating systems (for example, Windows and Linux), operating system binaries (32-bit and 64-bit), and Oracle Database binaries (32-bit and 64-bit). For the latest capabilities and restrictions, see support note 413484.1 at `http://metalink.oracle.com/`.

- Convert data files that contain UNDO to the new platform format (data files without UNDO do not require conversion)
- Transfer all data files from the source system to the destination system

  You can significantly minimize this time by using a storage infrastructure that can make the data files available to the destination system without the need to physically move the files.
- Invalidate and recompile all PL/SQL using SQL scripts utlirp.sql and utlrp.sql

See Also: The "Platform Migration using Transportable Database" white paper available at http://www.otn.oracle.com/goto/maa

4.1.11.2 Solution Description for Platform Migration Using Oracle Streams

Oracle Streams enables replication of updates between multiple databases, independent of Oracle platform or database release. Therefore, Oracle Streams may provide the fastest approach for database upgrades and platform migration.

Oracle Streams provides database support for a wide variety of datatypes, but does not provide native support for data movement of some advanced datatypes, such as for SDO_GEOMETRY and object types. However, you can work around datatype restrictions as follows:

- By using Extended Datatype Support (EDS), you can take advantage of the flexibility of Streams to accommodate several more advanced datatypes.

  A PL/SQL package, EXTENDED_DATATYPE_SUPPORT (EDS), is available to generate the appropriate database objects to accomplish this workaround. The EXTENDED_DATATYPE_SUPPORT package is available for download as an attachment to this article. The downloaded file (available from My Oracle Support Note 556742.1:1) contains a Readme file and SQL files to load in the database.

  The EDS package generates workaround scripts to enable Oracle Streams support on tables with the following data types:

  – Object column with simple object types
  – Object column with nested object types
  – Varray
  – Spatial type SDO_GEOMETRY
  – XMLType

  After installing the EDS package, you can query the EDS_SUPPORTED view to identify the list of tables with datatypes unsupported natively by Oracle Streams that can be supported with EDS.
- By creating *shadow tables* on the source database.

  You can create a trigger on tables with unsupported data types to capture and propagate changes to tables with supported data types. Those changes are replicated by Oracle Streams to the destination database. You can customize the apply mechanism to apply the changes to the original tables in the destination database.

Oracle Streams implementations are very flexible and can be customized, and thus may require additional effort for configuration, testing, and administration.

To perform a platform migration with Oracle Streams:

1. Set up the Oracle Streams environment on the source database.
2. Instantiate the replica database (destination database) using the new destination version or on the destination platform.
3. Set up the Oracle Streams environment on the destination database.
4. Enable Oracle Streams to propagate all changes made on the source database to the destination database to completely synchronize the destination database with the source.
5. Connect users to destination database and shutdown source database.
6. Remove the Oracle Streams configuration.

See Also:

- The MAA white paper "Extended Datatype Support: SQL Apply and Streams" at `http://www.otn.oracle.com/goto/maa`
- My Oracle Support Note 556742.1:1 at `http://metalink.oracle.com`
- *Oracle Streams Concepts and Administration*

4.1.12 Platform Migration Across Different Endian Format Platforms

Consider the following approaches when performing platform migrations on different endian format platforms:

- Transportable tablespace is the recommended solution for performing platform migration across different endian format platforms and reduces downtime significantly. See the "Solution Description for Transportable Tablespace" section on page 4-13 for more details.
- Oracle Data Pump is the simplest of all the approaches. See *Oracle Database Utilities* for complete information about using Oracle Data Pump.
- For planned downtime of potentially seconds, consider using Oracle Streams as described in Section 4.1.11.2, "Solution Description for Platform Migration Using Oracle Streams" on page 4-12.

Solution Description for Transportable Tablespace

Migrating a database to a new platform using a different endian format with transportable tablespaces requires the following high level steps.

To migrate a database to a new platform using Transportable Tablespace:

1. Create a new, empty database on the destination platform.
2. Import objects required for transport operations from the source database into the destination database.
3. Export transportable metadata for all user tablespaces from the source database.
4. Transfer data files for user tablespaces to the destination system.
5. Use RMAN to convert the data files to the destination system's endian format.
6. Import transportable metadata for all user tablespaces into the destination database.

7. Import the remaining database objects and metadata (that were not moved by the transport operation) from the source database into the destination database.

If the destination database is being moved to a new location (for example, to a new data center) during the migration, then create a physical standby database from the original primary database co-located with the destination database. After an Oracle Data Guard switchover, transport the tablespaces from the source to the destination without incurring the file transfer time as part of the downtime.

Additional Considerations

Transportable tablespace has limitations and restrictions in regard to character sets, opaque types, and system tablespace objects. Unlike previous solutions, the steps are not automated.

Perform a platform migration using transportable tablespaces if all of the following are true:

- The source and destination platforms have different endian formats.
- The time required to perform a full Data Pump Export and Import does not fit in the maintenance window.

See Also:

- The MAA white paper "Oracle Database 10*g* Release 2 Best Practices: Platform Migration using Transportable Tablespaces" available at `http://www.otn.oracle.com/goto/maa`
- *Oracle Database Backup and Recovery User's Guide* for information about data file conversion

4.2 Dynamic Resource Provisioning

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

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

4.2.1 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 grow and shrink 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[3]

- 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
- 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.

4.2.2 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.

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

Table 4–3 MEMORY_MAX_TARGET and MEMORY_TARGET

IF ...	AND ...	THEN ...
You omit `MEMORY_MAX_TARGET`	You omit `MEMORY_TARGET`	The initialization parameters are left at their default values (0) and Oracle Database does not automatically tune memory
You omit `MEMORY_MAX_TARGET`	Include a value for `MEMORY_TARGET`	The database automatically sets `MEMORY_MAX_TARGET` to the value of `MEMORY_TARGET`
You omit `MEMORY_TARGET`	Include a value for `MEMORY_MAX_TARGET`	The `MEMORY_TARGET` parameter defaults to zero

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 the *Oracle Database Administrator's Guide* for more information about all supported platforms.

[3] See the Exadata white paper "Best Practices for Migrating to HP Oracle Exadata Storage Server" at http://www.oracle.com/technology/products/bi/db/exadata/index.html.

4.2.3 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 Database Storage Administrator's Guide*

4.3 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.

This online architecture provides the following benefits:

- Online table 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.
- Online index operations:
  - Create indexes online and analyze them simultaneously. You can also use online repair of the physical guess component of logical rowids (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 rowids 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.
- 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 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 DML (no exclusive DML locks are required)
- No recompilation of dependent objects when online redefinition does not logically affect objects (for example, when columns are added to tables, or when procedures are added to packages)
- Easier table DDL operations online (there is an option to wait for active DML operations instead of aborting)
- Support for redefinition of tables that have materialized views or materialized view logs

The ability to modify table physical attributes and transform both data and table structure has been available since Oracle8*i*. Table 4–4 provides a comprehensive table of data reorganization capabilities.

Table 4–4 New Data Reorganization Capabilities by Release

Action	Oracle 9*i*	Oracle Database 10*g* Release 1	Oracle Database 10*g* Release 2	Oracle Database 11*g*
Online Reorganization using the package `DBMS_REDEFINITION`	Modify table storage parameters Move the table to a different tablespace Add support for parallel queries Add or drop partitioning support Re-create the table to avoid fragmentation Change from a table to an Index-Organized Table, or vice-versa Add or drop a column Transform a column using a function	Clones grants, constraints, and triggers Convert a `LONG` to a `LOB` Reorganize using a unique key Specify columns to order table by	Reorganize a single partition Advanced queue and clustered tables Table containing an ADT Retain and clone statistics Clone check and not null constraints Copies dependent objects for nested tables	Table with materialized view logs or materialized views No recompilation of dependent objects when redefinition does not logically affect objects

Reclaiming Unused Space	Not applicable	Use the `SHRINK SPACE` clause on the following statements:
		`ALTER TABLE`
		`ALTER INDEX`
		`ALTER MATERIALIZED VIEW`
		`ALTER MATERIALIZED VIEW LOG`
			Not applicable	Not applicable

Index Create Online	`CREATE INDEX emp.ename.idx ON emp(ename) ONLINE;` ■ Parallel operations supported ■ Partitions supported ■ All index types except cluster	Not applicable	Not applicable	DML lock-free online index creation, allowing transparent creation with no dependency on workload
Index Coalesce Online	`ALTER INDEX emp.ename_idx COALESCE;` ■ Parallel operations supported ■ Partitions supported ■ All index types	Not applicable	Not applicable	Not applicable
Index-Organized Table Move Online	`ALTER TABLE emp MOVE ONLINE;` ■ Parallel operations not supported ■ Partitions supported ■ Index-Organized Table only	Not applicable	Not applicable	Not applicable

See Also: *Oracle Database Administrator's Guide*

4.4 Transportable Technologies

For database migration to a new platform, use the transportable technology features. Transportable technologies provides transportable database and transportable tablespace:

- **Transportable database** moves an entire database (user data and the Oracle dictionary) to a new platform with the same endian format. Transportable database permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the destination database.
- **Transportable tablespaces** moves a subset of one database into another, even among platforms that differ in endian format:
  - You can use the cross-platform capability of transportable tablespaces to migrate all user data in a database to a new platform with a different endian format. Leveraging transportable tablespaces in this manner permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the destination database.
  - You can use transportable tablespaces to reduce downtime for database upgrades in circumstances where the database has simple schemas and when the data files do not have to be copied during the transport process (for example, when the data files are used in place.

    See Also:

    - Section 4.1.10.2, "Solution for Database Upgrades Using Transportable Tablespaces" and Section 4.1.11.1, "Solution Description for Platform Migration Using Transportable Database"
    - *Oracle Database Administrator's Guide* for details about how to move or copy tablespaces to another database, including details about transporting tablespaces across platforms

4.5 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 Streams 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
- Materialized View Logging Control

- Dependent PL/SQL Recompilation After Online Table Redefinition

4.5.1 Edition-Based Redefinition

Edition-based redefinition allows you to upgrade the database component of an application while the application is in use, thereby minimizing or eliminating down time. 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 pre-upgrade and the post-upgrade editions can be used concurrently so that sessions that were started before the post-upgrade edition was published can continue to use the pre-upgrade edition until they are terminated naturally while new sessions use the post-upgrade edition. In less favorable cases, all pre-upgrade sessions must be terminated before new sessions can be allowed to use the post-upgrade edition. In such cases, the application suffers a small amount of downtime.

The following sections describe the Editions, Editioning Views, and Crossedition Triggers features of edition-based redefinition.

See Also: *Oracle Database Advanced Application Developer's Guide*

4.5.1.1 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 allows 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 have been made in private, they are published in a single atomic operation.

4.5.1.2 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 post-upgrade edition to expose and to populate them. (Editions do not allow versions of the underlying table.)

Triggers may be created on an editioning view and its columns may be used in SQL hints. The defining `SELECT` statement for an editioning view has exactly one table in its `FROM` list and `NO WHERE` clause. 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.

4.5.1.3 Crossedition Triggers

Crossedition triggers are used as part of edition-based redefinition to keep the data in the pre-upgrade and post-upgrade editions in step with each other. The pre-upgrade application remains in use concurrently while changes are applied, redefining the pre-upgrade edition to a post-upgrade edition.

If users must be able to change data in the tables while you are changing the table structure, you also use *forward* crossedition 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* crossedition triggers. Crossedition 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.

4.5.2 Oracle Streams for Rolling Upgrades

Consider using Oracle Streams for fast rolling upgrades. However, while Oracle Streams upgrades can achieve little or no database down time, your ability to configure this solution will require some operational investment. See Section 3.6, "Oracle Streams" and Oracle Streams Concepts and Administration for more information.

4.5.3 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 could have possibly succeeded subseconds later. Specifying DDL 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.

See Also: *Oracle Database Administrator's Guide*

4.5.4 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.

See Also: *Oracle Database Advanced Application Developer's Guide*

4.5.5 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 `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 subseconds and works independently of the existing data volume, but it also consumes no space.

See Also: *Oracle Database Administrator's Guide*

4.5.6 Finer-Grained Dependencies

Prior to Oracle Database 11*g*, metadata would record 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 cases such as these, the dependent objects were sometimes invalidated when there was no logical requirement to do so. 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 11*g* 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 10*g*, 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, application availability is increased. The benefit occurs both in the development environment and when a live application is parsed or upgraded. The benefit occurs when an Oracle Database patch set is applied because changes to schema objects must be compatible and, therefore does not cause consequential invalidations.

See Also: *Oracle Database Advanced Application Developer's Guide*

4.5.7 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 have to be modified without being able to bring the complete application 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.

See Also: *Oracle Database Administrator's Guide*

4.5.8 Materialized View Logging Control

Oracle Database includes session-level control for materialized view logs. You can disable the capture of changes for materialized views (materialized view logs) for an individual session while logging continues for changes made by other sessions. This feature reduces application patching downtime.

See Also: *Oracle Database Data Warehousing Guide*

4.5.9 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.

This feature reduces the time and effort to manually recompile dependent PL/SQL after an online table redefinition. This also includes views, synonyms, and other table dependent objects (with the exception of triggers) that are not logically affected by the redefinition.

See Also: *Oracle Database Administrator's Guide* for information about redefining tables online

5

Optimizing Return on Investment (ROI)

Oracle Grid Computing, disaster-recovery solutions and advanced standby database usage, and virtualization all optimize Return on Investment (ROI) capabilities.

You can scale out your existing system infrastructure while achieving both high availability and disaster protection. Oracle Data Guard standby databases are an integral part of the Grid, providing data protection and availability 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.

This chapter covers the following topics:

5.1 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.

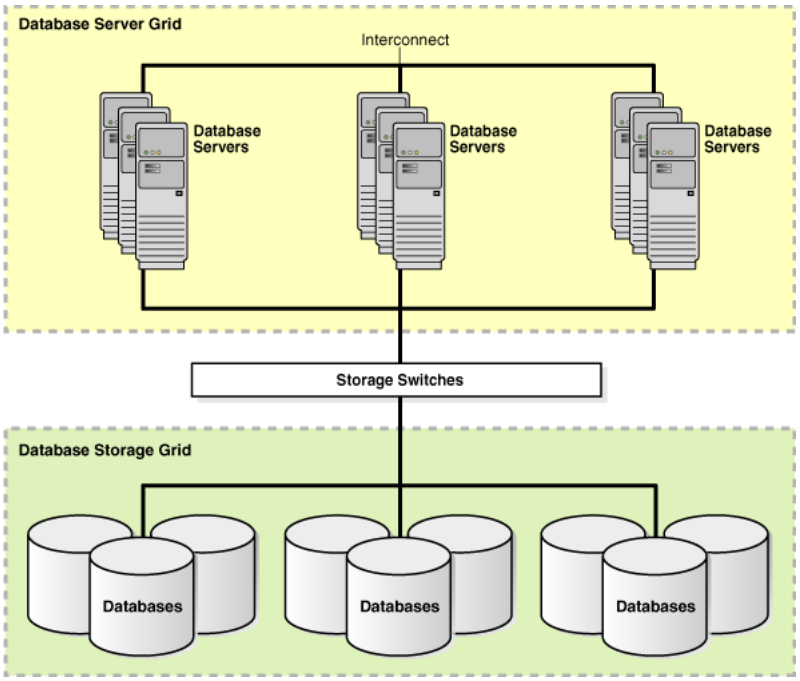
Oracle Database captures the cost advantages of Grid enterprise computing without sacrificing performance, scalability, security, manageability, functionality, or system availability.

- A **Database Server Grid** is a collection of commodity servers connected together 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 computers in the Database Server Grid.

The same Grid computing concepts can be used to create a standby database hub that provides data protection, minimizes planned downtime, and provides ideal test systems for quality assurance testing and all for multiple primary databases. Grid capabilities enable system resources to be dynamically provisioned and de-provisioned depending on current priorities. For example, if a primary database fails over to one of the standby databases in the Data Guard hub, it can be allocated more system and storage resources while resources allocated to test activities are reduced. The Grid enables a high level of utilization and low TCO without compromising your business requirements.

Figure 5–1 illustrates the Database Server Grid and Database Storage Grid in a Grid enterprise computing environment.

Figure 5–1 Grid Computing Environment



5.2 Database Server Grid

The availability of low-cost and reliable blade servers, small multiprocessor servers, and inexpensive 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 RAC 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 consolidate the management in increased 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 in 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.

5.3 Database Storage Grid

The availability of low-cost ATA disk-based storage arrays and low-cost storage networks has made it possible to use a Database Storage Grid with Oracle Database at 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 (ASM) and Oracle Exadata Storage Server Software or ASM and third-party storage. The Oracle Storage Grid with Exadata seamlessly supports MAA-related technology, improves performance, provides unlimited I/O scalability, is easy to use and manage, and delivers mission-critical availability and reliability to your enterprise. See the *Oracle Database High Availability Best Practices* for Oracle Storage Grid recommendations and using Exadata.

A database administrator can use the Oracle ASM interface to specify the disks in the Database Storage Grid that Oracle ASM should 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 storage as storage arrays are added or removed from the Database Storage Grid.

Additionally, use the I/O Resource Management (IORM) to manage and meet service-level requirements. IORM allows you manage the grid and prioritized applications within the database or in between databases.

5.4 Disaster Recovery Using Active Standby Databases

You can use standby databases for dynamic IT and application requirements in addition to providing disaster recovery. The **Active Data Guard option** in Oracle Data Guard enables you to use physical standby databases for other useful work during normal operations, in addition to providing a disaster-recovery solution.

The following sections describe the Oracle Data Guard features that help you use physical standby databases for additional business purposes:

- Active Data Guard Option for Physical Standby Databases
- Web Scale Using Standby Reader Farms

5.4.1 Active Data Guard Option for Physical Standby Databases

Oracle 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 **Active Data Guard option** (available with Oracle Database 11*g* Release 1 (11.1) and later releases) enables a physical standby database to be opened for read-only access while Redo Apply is active. This makes it possible to run queries and reports against an up-to-date physical standby database without compromising data protection or extending recovery time in the event a failover is required. You can offload the read-only workload from the primary database to the active standby database, improving performance and postponing the day when you need to purchase additional capacity.

To enable the Active Data Guard option, open the database in read-only mode and then issue the `ALTER DATABASE RECOVER MANAGED STANDBY` statement. Note that the `COMPATIBLE` parameter must be set to 11.0.0 or later on both the primary and physical standby databases. Using this feature is completely transparent to any application requiring read-only access to the Oracle Database.

The Active Data Guard option provides an ultimate high availability solution because it:

- Supports Oracle RAC on the primary and standby databases

  The Active Data Guard option works on both single-instance and Oracle RAC physical standby databases. Although Redo Apply can be running on only one Oracle RAC instance, any of the instances can run in read-only mode, including the apply instance.

- Returns transactionally consistent results that are very close to being up to date with the primary database

  Depending on any delay settings or apply rates, the standby database can be current with the primary database or lagging seconds behind. The queries will always be transactionally consistent and will represent a consistent view of the last committed transaction at that time.

- Allows fast switchovers or failovers because the redo generated by the primary database while the standby database was open read-only has already been applied to the standby database, making it immediately available to assume the primary database role

- Enables you to use fast-start failover for automatic failover in the case the primary database fails

---

Note: Transactions that attempt to modify a physical standby database running with Active Data Guard enabled will fail with an error.

See Also: *Oracle Data Guard Concepts and Administration* for complete information about using Active Data Guard

5.4.2 Web Scale Using Standby Reader Farms

You can use multiple physical standby databases (using the **Active Data Guard option**) and logical standby databases to deploy a *reader farm*. An example of such a configuration is provided Figure 5–2, complete with the use of Oracle 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 you to boost 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 Grid architecture where I/O is the driving factor.

The concept is straightforward—a single primary database that supports read/write transactions, and multiple standby databases that provide read-only access to Web users. Such an approach scales read performance linearly as additional standby databases are added. It is also an effective way to isolate faults, because problems that

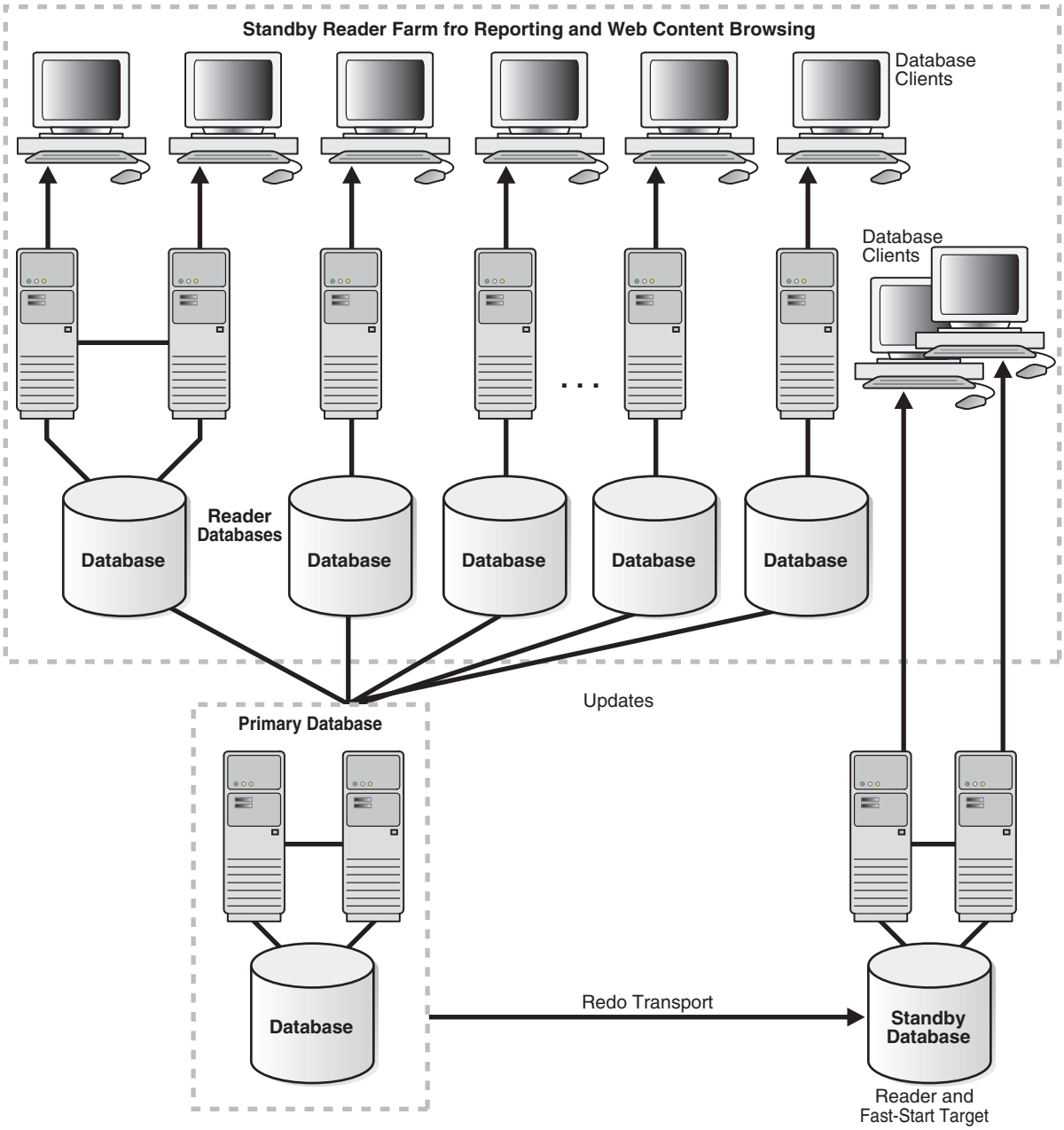
affect one standby database are isolated from the other standby databases in the configuration.

Creating a reader farm of physical standby databases provides the following benefits:

- 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

Figure 5–2 shows a good example of how you can use Oracle Data Guard, physical standby databases and **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 database, one of which is also enabled for fast-start failover for automatic, zero or minimal data loss failover.

Figure 5–2 Standby Database Reader Farms



If a fast-start failover is triggered in the Oracle Data Guard configuration in Figure 5–2, then:

- Automatic failover occurs to the designated standby database
- All standby databases accept data from new primary database
- You can perform a switchover at a convenient time in the future to return all databases to their original roles

5.5 Oracle VM and Domain Live Migration

Grid Computing, disaster-recovery solutions with advanced standby database usage, and virtualization all encourage better ROI. Virtualization's main benefits include consolidation and using all resources efficiently.

Oracle VM enables you to deploy operating systems and application software within a supported virtualization environment. Because each virtual machine has its own virtual CPU, network interfaces, storage and operating system, Oracle VM disassociates workloads from the physical constraints of the underlying hardware. Oracle VM presents the opportunity to significantly reduce service outages associated with planned server outages or workload imbalance.

You can improve availability by using the domain live migration feature of Oracle VM to migrate a domain from one physical server to another, identical computer that is running virtual machines. For example, you can:

- Migrate the domain from a failing server to a viable server on which operations can continue during repairs
- Rebalance the workload to prevent one server from becoming overloaded

During the migration, the domain continues to provide services and end users remain unaware of any change.

See Also:

- *Oracle VM Server User's Guide*
- The Oracle VM Web site on OTN at http://www.oracle.com/technologies/virtualization/index.html

6

Optimizing Manageability

Complex environments demand coordinated configuration changes, system upgrades and new application roll-outs. The topics in this section describe how to automate and simplify operations in high availability architectures, allowing you to step toward self-managing Oracle databases.

This section contains these topics:

6.1 Intelligent Infrastructure

Oracle Database has a sophisticated self-management infrastructure that allows the database to learn about itself and use this information to adapt to workload variations or to automatically remedy any potential problem. The self-management infrastructure includes the following:

- **Automatic Workload Repository**

  The Automatic Workload Repository (AWR) is a built-in repository that contains performance statistics used by Oracle Database for problem detection and self-tuning purposes. At regular intervals, Oracle Database makes a snapshot of vital statistics and workload information and stores them in the AWR. The data contained in the snapshots is then analyzed by the Automatic Database Diagnostic Monitor (ADDM).

  See the *Oracle Database Performance Tuning Guide* for information about the AWR and ADDM.

- **Active Session History**

  Transient performance problems are short-lived and do not appear in the ADDM analysis. To address these problems, you can use Active Session History (ASH) to start sampling active sessions when the database starts. In particular, ASH samples can be collected:

  – Before the database is mounted, such as on an Oracle ASM instance.

  – When the database is mounted but not open, such as on an Oracle Data Guard physical standby instance.

  – When the database is mounted but open read-only, such as on an Oracle Active Data Guard physical standby instance (also known as the real-time query feature).

Note: Active session history sampling is available for Oracle Active Data Guard instances and Oracle Automatic Storage Management (ASM) instances. However, not all Intelligent Infrastructure features are available on Oracle Data Guard configurations for this release.

On a physical standby instance, the ASH data on disk represents activity on the primary database and the ASH data in memory represents activity on the standby database (in `V$ACTIVE_SESSION_HISTORY`). The ASH report prompts you to specify whether to generate the report using data sampled from the primary or standby database.

See Also:

- The topic about generating an ASH report on a specific database instance in *Oracle Database Performance Tuning Guide*
- *Oracle Data Guard Concepts and Administration* for information about physical standby databases and Oracle Active Data Guard and real-time queries
- The MAA white paper "Oracle Active Data Guard for Oracle Data Guard 11g Release 1" at `http://otn.oracle.com/goto/maa`

- **Automatic Maintenance Tasks**

  By analyzing the information stored in the AWR, the database can identify the need to perform routine maintenance tasks. The automated maintenance tasks infrastructure (known as "AutoTask") enables Oracle Database to automatically schedule such operations. AutoTask schedules automatic maintenance tasks to run in a set of Oracle Scheduler windows known as maintenance windows. Maintenance windows are those windows that are members of the Oracle Scheduler window group `MAINTENANCE_WINDOW_GROUP`. See the *Oracle Database Administrator's Guide* and the *Oracle Database 2 Day DBA* for more information.

- **Fault diagnosability infrastructure**

  Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems that are targeted are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption. This includes:

  – The automatic diagnostic repository (ADR), which is a file-based repository for database diagnostic data such as traces, the alert log, health monitor reports, and more. It has a unified directory structure across multiple instances and multiple products.

  – The incident packaging services that a database administrator can use to automatically and easily gather all diagnostic data (traces, health check reports, SQL test cases, and more) pertaining to a critical error and package the data into a zip file suitable for transmission to Oracle Support.

  See the *Oracle Database Administrator's Guide* for more information about these components.

- **Server generated alerts**

  For problems that cannot be resolved automatically and require administrators to be notified (such as running out of space) Oracle Database provides server-generated alerts. Oracle Database can monitor itself and send out alerts to

notify you of any problem and provide recommendations on how the reported problem can be resolved. This ensures quick problem resolution and helps prevent potential failures.

- **Advisor framework**

  Oracle Database includes a number of advisors for different subsystems in the database to automatically determine how the operation of the corresponding subcomponents could be further optimized. The SQL Tuning Advisor and the SQL Access Advisor, for example, provide recommendations for running SQL statements faster. Memory advisors help size the various memory components without resorting to trial-and-error techniques. The Segment Advisor handles space-related issues, such as recommending wasted-space reclamation and analyzing growth trends, while the Undo Advisor guides you in sizing the undo tablespace correctly. See the *Oracle Database 2 Day DBA* for more information about using advisors.

6.2 Change Assurance

Oracle Database provides automatic capture and replay of workloads before and after changes so that you can analyze the effect of a database or a SQL change:

- Database Replay

  The Database Replay feature enables you to perform real-world testing by capturing the actual database workload on the production system and replaying it on the test system. It also provides analysis and reporting to highlight potential problems (for example, errors encountered and divergence in performance) and recommend ways to remedy the problems.

- SQL Performance Analyzer

  SQL performance regression is always a concern during system changes such as database upgrades, initialization parameter changes, and adding or dropping indexes. The SQL Performance Analyzer feature alleviates this concern by providing a way to assess the impact of a change on the performance of SQL statements by comparing and contrasting their response times before and after the change. SQL Performance Analyzer enables you to capture the SQL workload from the source system, such as the production database, and to replay it on the test system where the change has been applied.

See Also: *Oracle Database Real Application Testing User's Guide*

6.3 Oracle Enterprise Manager Grid Control

By reducing the amount of human intervention required to execute routine and repetitive tasks, services become more stable, reliable, and available. This is particularly important when administrators need to manage very large numbers of systems as efficiently as possible.

Oracle Enterprise Manager Grid Control is an HTML-based interface that provides the administrator with complete monitoring across the entire Oracle technology stack—business applications, application servers, databases, and the E-Business Suite—and non-Oracle components. If a component of fast application notification (FAN) becomes unavailable or experiences performance problems, then Grid Control displays the automatically generated alert so that the administrator can take the appropriate recovery action.

The components of Grid Control include:

- Oracle Management Service (OMS)

  The OMS is now a set of J2EE applications that renders the interface for Grid Control, works with all Management Agents to process monitoring information, and uses the Management Repository as its persistent data store.

- Oracle Management Agents

  These are processes deployed on each monitored host to monitor all targets on the host, communicate that information to OMS, and maintain the host and its targets.

- Oracle Management Repository

  This is a schema in Oracle Database that contains all available information about administrators, targets, and applications managed by Grid Control.

Communication between Grid Control, the OMS, and Oracle Management Agents is done through HTTP. Also, you can enable Secure Sockets layer (SSL) to allow secure communications between tiers in firewall-protected environments. The Management Agent uploads collected monitoring data to the OMS, which in turn loads the data into the Management Repository. Changes in a target state (such as an availability state change) result in an alert being generated to Grid Control.

Using Grid Control, an administrator can:

- Monitor architecture components and be alerted when a failure occurs
- View overall system status, such as the number of nodes in the database cluster and their current status
- View alerts aggregated across all instances
- Set thresholds for alert generation for each database on a clusterwide basis
- Monitor performance metrics across all instances
- Perform database clusterwide operations such as backup and recovery
- Interconnect monitoring of cluster databases

See Also:

- *Oracle Enterprise Manager Grid Control Installation and Basic Configuration* and *Oracle Enterprise Manager Concepts* for more information about Oracle Enterprise Manager Grid Control
- The MAA white papers for configuring Enterprise Manager for high availability at

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

7

High Availability Architectures and Solutions

The Maximum Availability Architecture (MAA) is Oracle's best practices blueprint. It is based on proven Oracle high availability technologies and recommendations. The goal of the MAA is to remove the complexity in designing the optimal high availability architecture by providing configuration recommendations and tuning tips to optimize your architecture and Oracle features.

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 Application Server 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
- Oracle Database with Oracle Clusterware (Cold Cluster Failover)
- Oracle Database with Oracle Real Application Clusters (Oracle RAC)
- Oracle Database with Oracle RAC on Extended Clusters
- Oracle Database with Oracle Data Guard
- Oracle Database with Oracle Clusterware and Oracle Data Guard
- Oracle Database with Oracle RAC and Oracle Data Guard
- Oracle Database with Oracle Streams

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 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 Availability Architecture (MAA)" for more information about the best practices documentation.

7.1.1 Oracle Database

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.

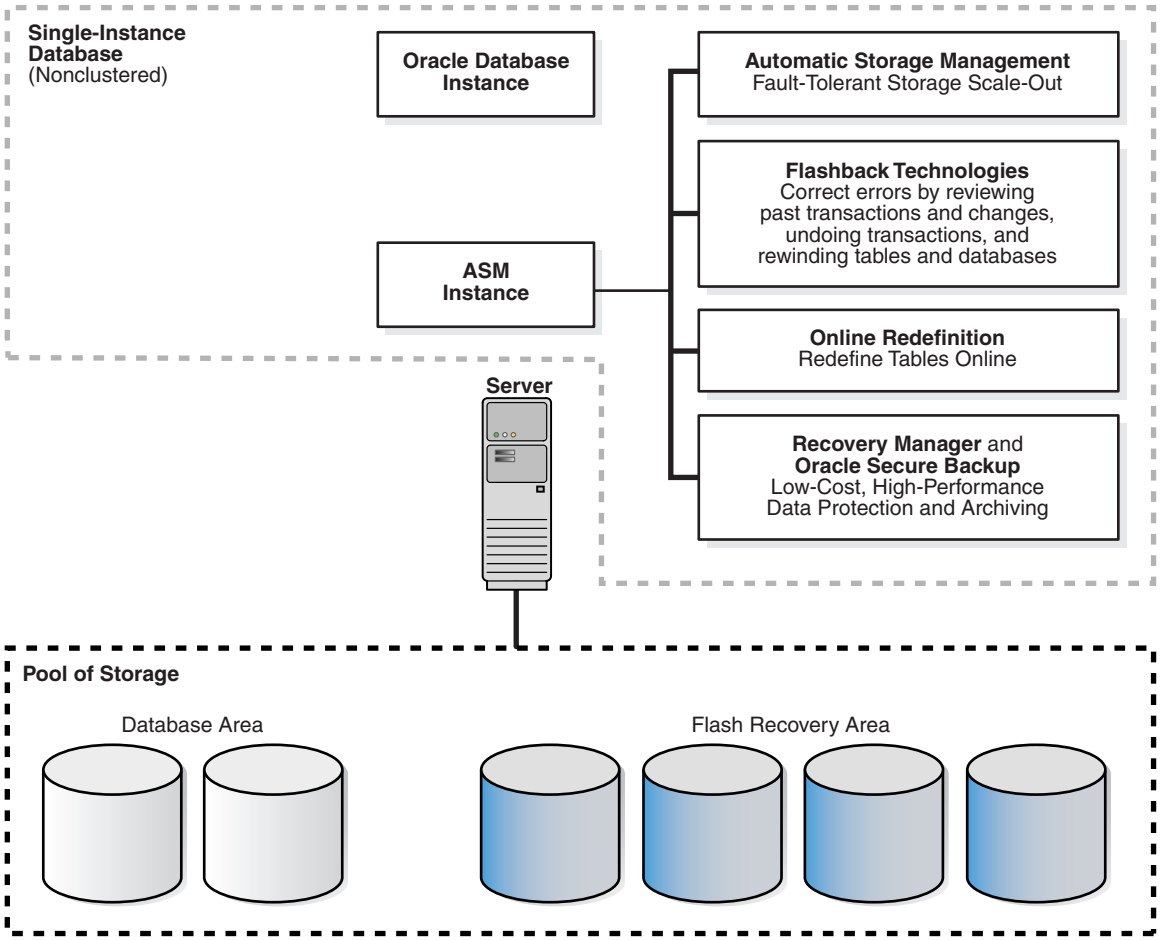
Oracle recommends that you use the following Oracle features to make a standalone database on a single computer available for certain failures and planned maintenance activities:

- 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.
- Oracle Automatic Storage Management (Oracle ASM) and Automatic Storage Management Cluster File System (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 Resource Provisioning allows for dynamic system changes.
- Online Patching allows for dynamic database patches 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

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

[1] Single-instance databases can use clustered Oracle ASM (Storage GRID) or nonclustered Oracle ASM.

Figure 7–1 Single-Node, Nonclustered Oracle Database with an Oracle ASM Instance



7.1.2 Oracle Database with Oracle Clusterware (Cold Cluster Failover)

Section 3.4.1 describes how 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. The servers on which you want to run Oracle Clusterware must be running the same operating system.

Many high availability architectures today use clusters alone to provide some rudimentary node redundancy and automatic node failover. However, when you use Oracle Clusterware, there is no need or advantage to using third-party clusterware.

Oracle Clusterware provides a number of benefits over third-party clusterware. Oracle Clusterware:

- Enables you to use an entire software solution from Oracle, avoiding the cost and complexity of maintaining additional cluster software.

  By reducing the combinations of software that you must coordinate and support, you can increase the manageability and availability of your system software.

- Provides seamless integration with, and migration to, Oracle Real Application Clusters (Oracle RAC) and Oracle Data Guard.

  Section 7.1.7 on page 7-19 describes how you can achieve the highest level of availability with Oracle RAC and Oracle Data Guard.

- Includes all of the features required for cluster management, including node membership, group services, global resource management, and high availability functions such as managing third-party applications, event management, and Oracle notification services that enable Oracle clients to reconnect to the new primary database after a failure.
- Uses a private network and a voting disk to detect and resolve *split-brain*[2] scenarios.

With Oracle Clusterware, you can provide a *cold cluster failover* to protect an Oracle instance from a system or server failure. The basic function of a cold cluster failover is to monitor a database instance running on a server, and if a failure is detected, to restart the instance on a spare server in the cluster. Network addresses are failed over to the backup node. Clients on the network experience a period of lockout while the failover occurs and are then served by the other database instance after the instance has started. Also, you can use the Oracle Clusterware ability to relocate applications and application resources (using the `CRS_RELOCATE` command) as a way to move the workload to another node so that you can perform planned system maintenance on the production server.

The cold cluster failover solution with Oracle Clusterware provides these additional advantages over a basic database architecture:

- Automatic recovery of node and instance failures in minutes
- Automatic notification and reconnection of Oracle integrated clients[3]
- Ability to customize the failure detection mechanism

  For example, you can use your favorite application query in the database check action. Providing application-specific failure detection means Oracle Clusterware can fail over not only during the obvious cases such as when the instance is down, but also in the cases when, for example, an application query is not meeting a particular service level.
- High availability functionality to manage third-party applications
- Rolling release upgrades of Oracle Clusterware

The operation of an Oracle Clusterware cold cluster failover is depicted in Figure 7–2 and Figure 7–3. These figures show how you can use the Oracle Clusterware framework to make both Oracle Database and your custom applications highly available.

Figure 7–2 shows a configuration that uses Oracle Clusterware to extend the basic Oracle Database architecture and provide cold cluster failover. In the figure, the configuration is operating in normal mode in which Node 1 is the active instance connected to Oracle Database that is servicing applications and users. Node 2 is connected to Node 1 and to Oracle Database, but it is currently standby mode.

[2] Network splits, commonly referred to as split brains, occur when nodes on one side of the cluster cannot see the nodes on the other side of the cluster.

[3] Oracle Clusterware sends the service events and FAN-integrated clients automatically react to those events.

Figure 7–2 Oracle Database with Oracle Clusterware (Before Cold Cluster Failover)

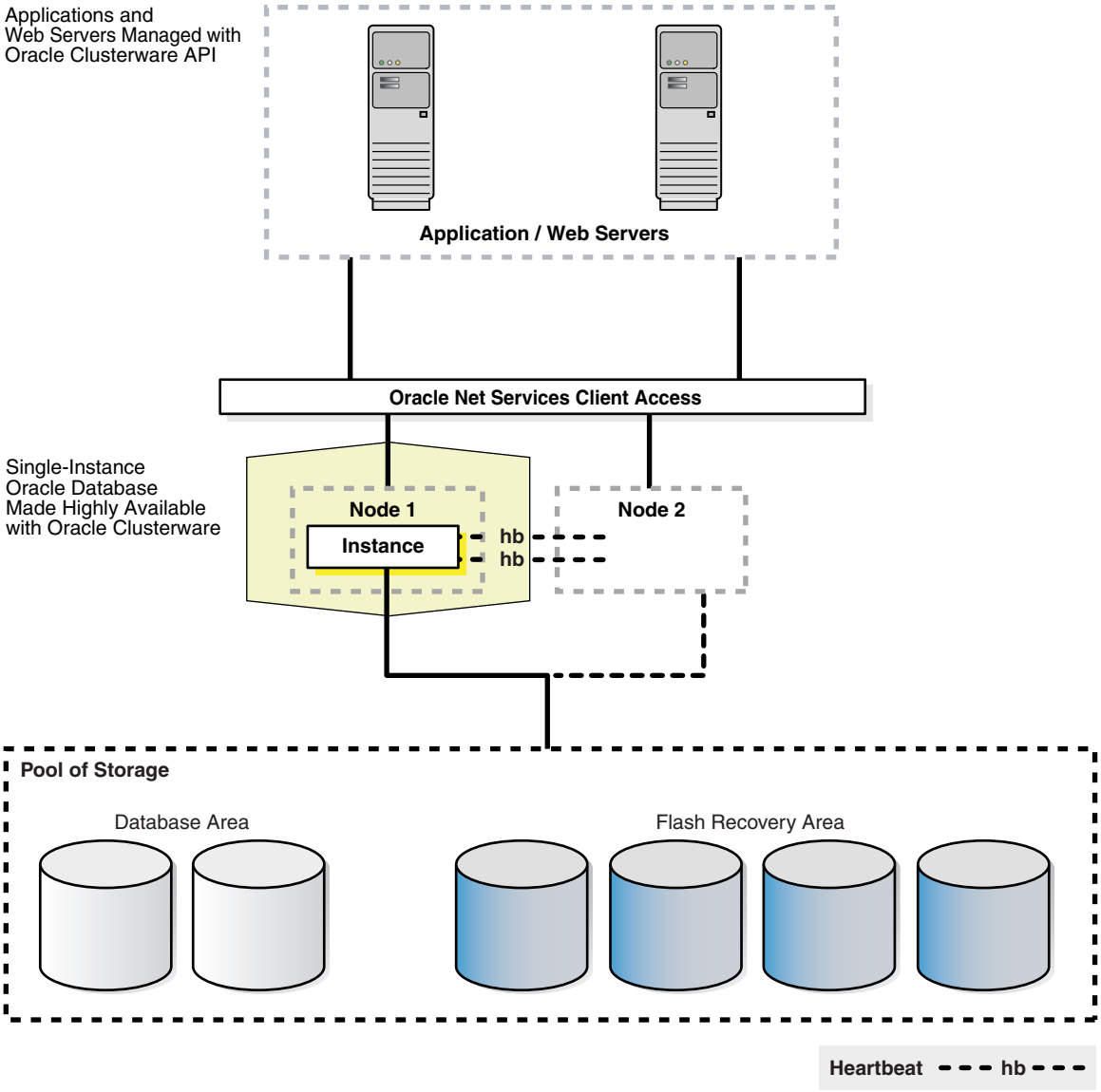
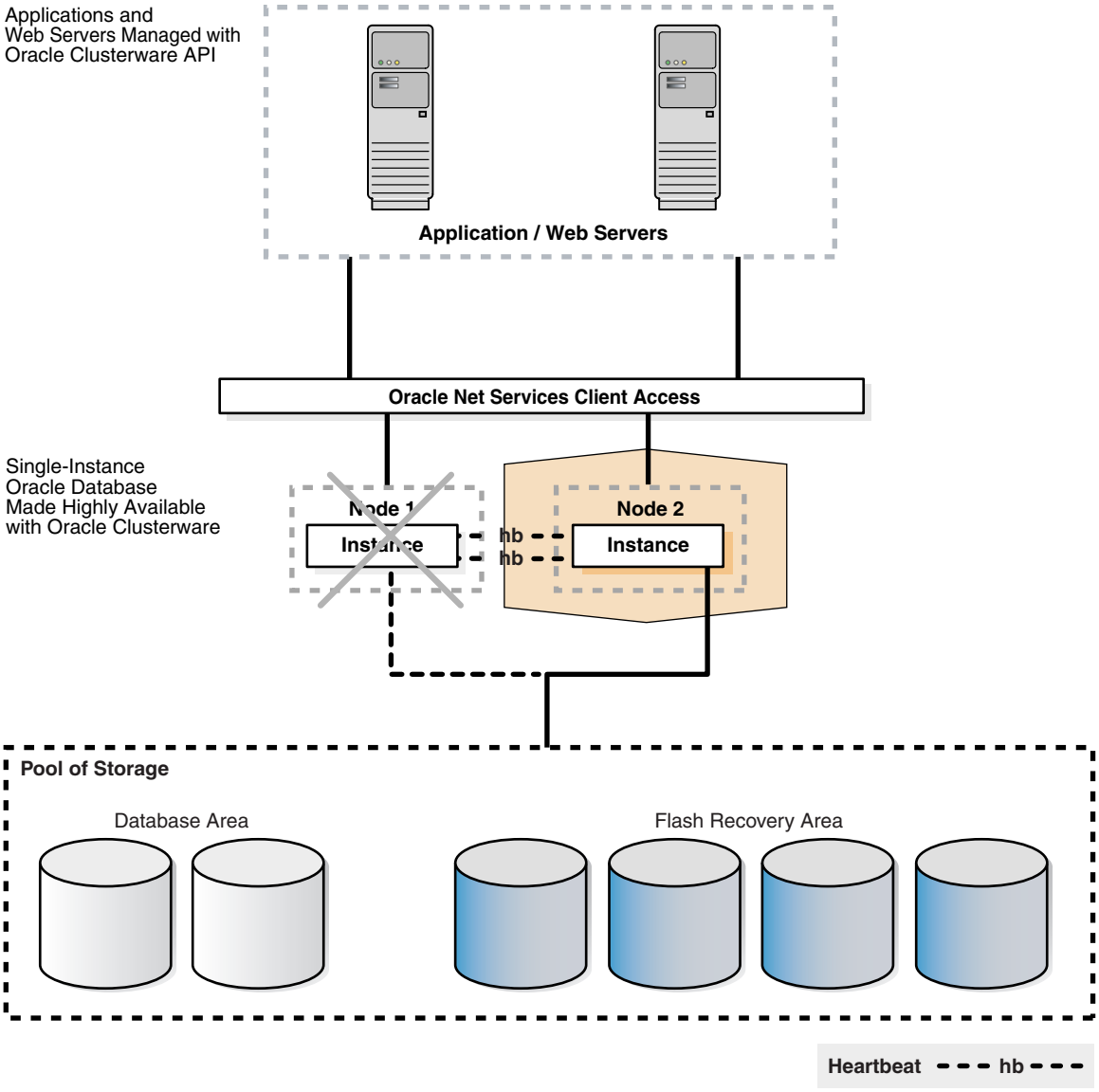


Figure 7–3 shows the Oracle Clusterware configuration after a cold cluster failover has occurred. In the figure, Node 2 is now the active instance connected to the Oracle database and servicing applications and users. Node 1 is connected to Node 2 and to the Oracle database, but Node 1 is currently idle, in standby mode.

To provide this transparent failover capability, Oracle Clusterware requires a virtual IP (VIP) address for each node in the cluster. With Oracle Clusterware, you also define an *application* VIP so that users can access the application independently of the node in the cluster where the application is running. You can define multiple application VIPs, with generally one application VIP defined for each application running. The application VIP is tied to the application by making it dependent on the application resource defined by Cluster Ready Services (CRS).

Figure 7–3 Oracle Database with Oracle Clusterware (After Cold Cluster Failover)



Note: Neither Oracle Enterprise Manager nor Oracle Universal Installer (OUI) provides configuration support for Oracle Clusterware. To configure an Oracle Clusterware environment, follow the step-by-step instructions in your platform-specific Oracle Clusterware installation 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.

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.

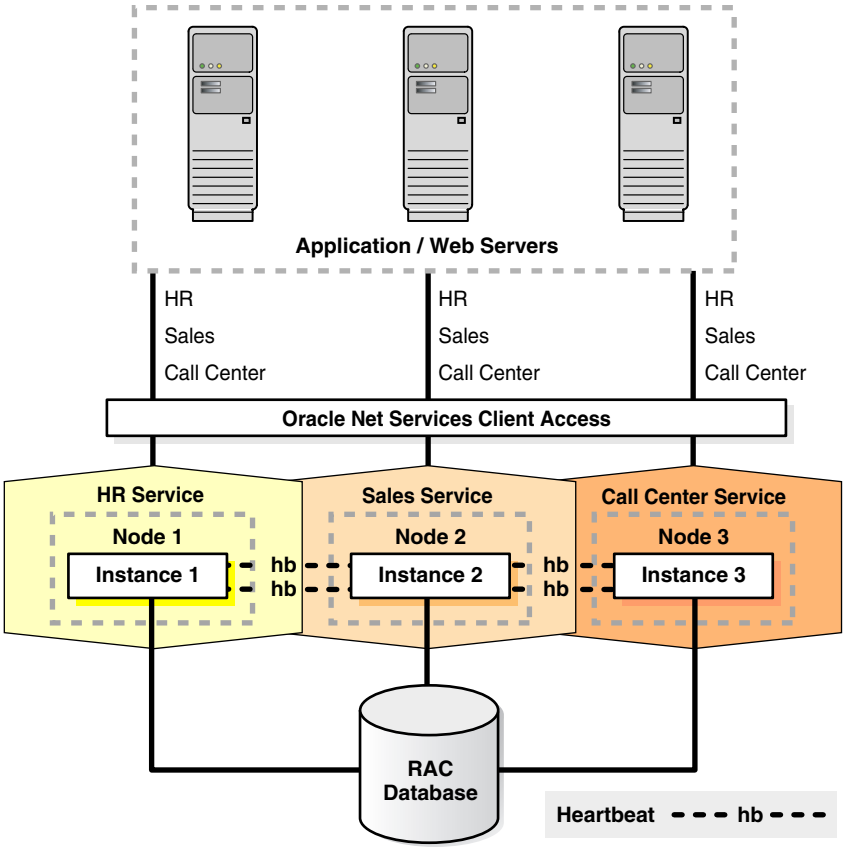
Oracle RAC exploits the redundancy that is provided by clustering to deliver availability with *n* - 1 node failures in an *n*-node cluster. Unlike the cold cluster model where one node is completely idle, all instances and nodes can be active to scale your application.

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

- 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)
- 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
- Load balancing advisory and Runtime Connection Load Balancing
- Comprehensive manageability integrating database and cluster features with Grid Plug and Play and policy-based cluster and capacity management
- Workload management, load balancing advisory, and runtime connection load balancing help redirect and balance work across the appropriate resources
- Quality of Service Management reduces the number of outages due to performance problems by identifying and advising on problems that can breach performance objectives
- Oracle Enterprise Management support for Oracle ASM and Cluster File System, Grid Plug and Play, Cluster Resource Management, Oracle Clusterware and Oracle RAC Provisioning and patching

Figure 7–4 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 7–4 Oracle Database with Oracle RAC Architecture



7.1.4 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.)

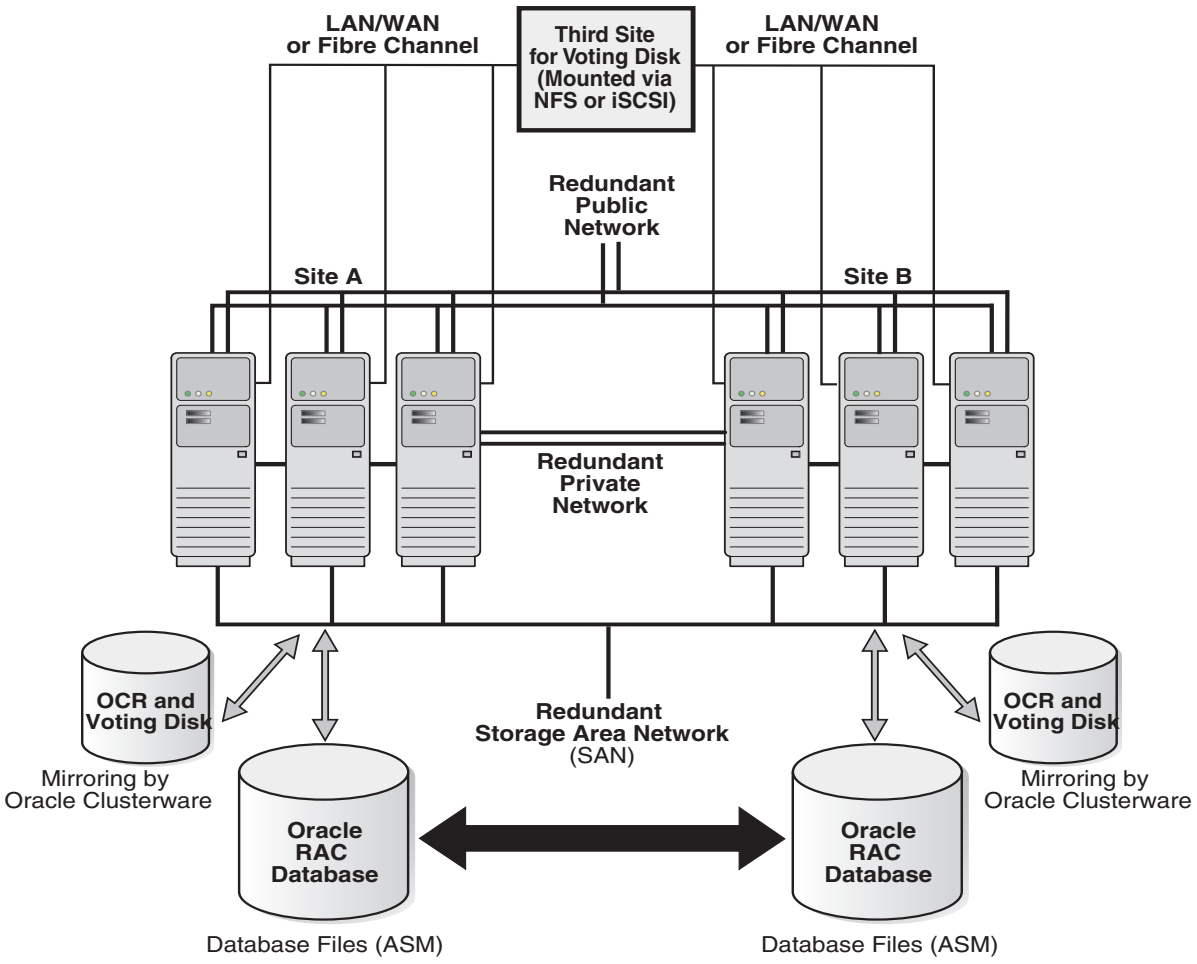
The advantages to using Oracle RAC on extended clusters include:

- 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 7–5 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.

Figure 7–5 Oracle RAC Extended Cluster



See Also:

- *Oracle Database High Availability Best Practices* for information about configuring Oracle Database 11*g* with Oracle RAC on extended clusters
- The white paper about extended (stretch) clusters on the Oracle Real Application Clusters Web site at http://www.oracle.com/technology/products/database/clustering/
- The white paper about using standard NFS to support a third voting disk on an extended cluster configuration that is available on the Oracle RAC Web site at http://www.oracle.com/technology/products/database/clustering/index.html

7.1.5 Oracle Database with Oracle Data Guard

Oracle Data Guard is a high availability 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, Oracle Data Guard provides the most comprehensive high availability and disaster recovery solution for Oracle databases.

Oracle Data Guard Advantages Over Traditional Solutions

Oracle 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
- Automatic corruption repair automatically replaces a corrupted block 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 Oracle 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) as part of the full site failover operations
- No need for instance restart, storage remastering, or application reconnections after site failures
- Transparency to applications
- Transparent and integrated support for application failover

- Effective network utilization

For data resident in Oracle databases, Oracle 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. Oracle 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.

Oracle Data Guard Advantages Compared to Remote Mirroring Solutions

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

- **Better network efficiency**—With Oracle 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 Oracle Data Guard, a remote mirroring solution must transmit each change many more times to the remote site.
- **Better performance**—Oracle 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.

  Oracle 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, Oracle Data Guard provides better performance and is more efficient, Oracle Data Guard always verifies the state of the standby database and validates the data before applying redo data, and Oracle Data Guard enables you to use the standby database for updates while it protects the primary database.
- **Better suited for WANs**—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**—Oracle 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 Oracle 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 Oracle 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 Oracle Data Guard. Automatic block repair may be possible, thus eliminating any downtime in an Oracle Data Guard configuration.

- **Higher flexibility**—Oracle 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**—Oracle 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: 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. Oracle 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 Oracle Data Guard are described in the following sections:

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

7.1.5.1 Overview of Single Standby Database Architectures

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

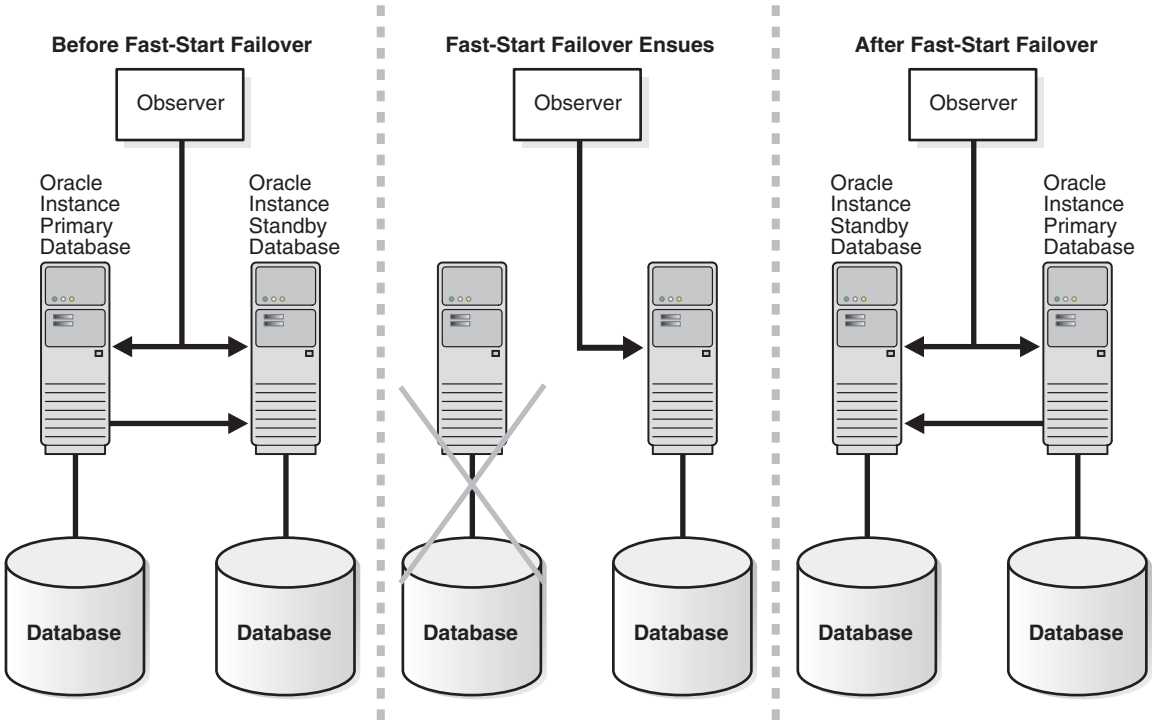
- Primary database resides in Site A.
- Standby database resides in Site B. If zero data loss is required with minimum performance impact on the primary database, then the best practice is to locate the secondary site within 200 miles of the primary database. Note, however, that the synchronous redo transport does not impose any physical distance limitation.
- 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 Oracle 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 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 7–6 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 Oracle 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. Oracle 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 `FastStartFailoverThreshold` property before initiating a fast-start failover. If the observer is unable to 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 7–6 Primary and Standby Databases and the Observer During Fast-Start Failover



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

- 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 Oracle 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 Oracle Data Guard to replace storage-based remote mirroring and maintain a standby database at its recovery site 50 miles away from the primary site. Oracle 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.

7.1.5.2 Overview of Multiple Standby Database Architectures

This architecture is identical to the single-standby database architecture that was described in Section 7.1.5.1, except that there are multiple standby databases in the same Oracle Data Guard configuration. The following list describes some implementations for a multiple standby database architecture:

- Continuous and transparent disaster or high availability protection if an outage occurs at the primary database or the targeted standby database
- Reader farms or lookup databases
- Reporting databases
- Regional reporting or reader databases for better response time
- Synchronous redo transport that transmits to a more *local* standby database, and asynchronous redo transport that transmits to a more *remote* standby database for optimum levels of performance and data protection
- Testing and development clones using snapshot standby databases
- Rolling upgrades

Note that it is possible to convert a physical standby database to a logical standby database or to a snapshot standby database, or you can create additional logical standby databases or snapshot standby databases:

- **Transient logical standby databases** can be used to minimize downtime for database upgrades. Using transient logical standby databases is helpful in Oracle Data Guard architectures where there are no logical standby databases.

  In a multiple standby database environment, you can create a transient logical standby database temporarily (for planned maintenance) and then convert it back to the physical standby database role. For example, you can use transient logical standby databases to minimize downtime for database upgrades, when required. There is no need to create a separate logical standby database to perform upgrades. The high-level steps for rolling upgrades with a transient logical standby database are as follows:

  1. Start performing a rolling database upgrade with the physical standby database.
  2. Temporarily convert the physical standby database to a logical standby database to perform the upgrade. (Note that data type restrictions are limited to the short window of time required to perform an upgrade.)

3. Revert the logical standby database to the physical standby database role.

   See Also: *Oracle Data Guard Concepts and Administration* or *Oracle Database High Availability Best Practices* for step-by-step instructions about performing a rolling upgrade with a transient logical standby database

- **Snapshot standby databases** can be used as clones or test databases to test new functionality and new releases. The snapshot standby database continues to receive and queue redo data so that data protection and the RPO are not sacrificed.

  Snapshot standby databases diverge from the primary database over time because redo data from the primary database is not applied when it is received. Redo Apply does not apply the redo data until you convert the snapshot standby database back into a physical standby database, and all local updates that were made to the snapshot standby database are discarded. Although the local updates to the snapshot standby database cause additional divergence, the data in the primary database is fully protected by the redo log that is located at the standby site.

Figure 7–7 shows the production database at the primary site and multiple standby databases at secondary sites. The figure shows Oracle Database with Oracle Data Guard architecture.

The production database is connected over the network to the physical standby database site and the logical standby database site (the standby databases may be at the same or different sites). The Oracle Data Guard broker communicates with the production database, the physical standby database, and the logical standby database.

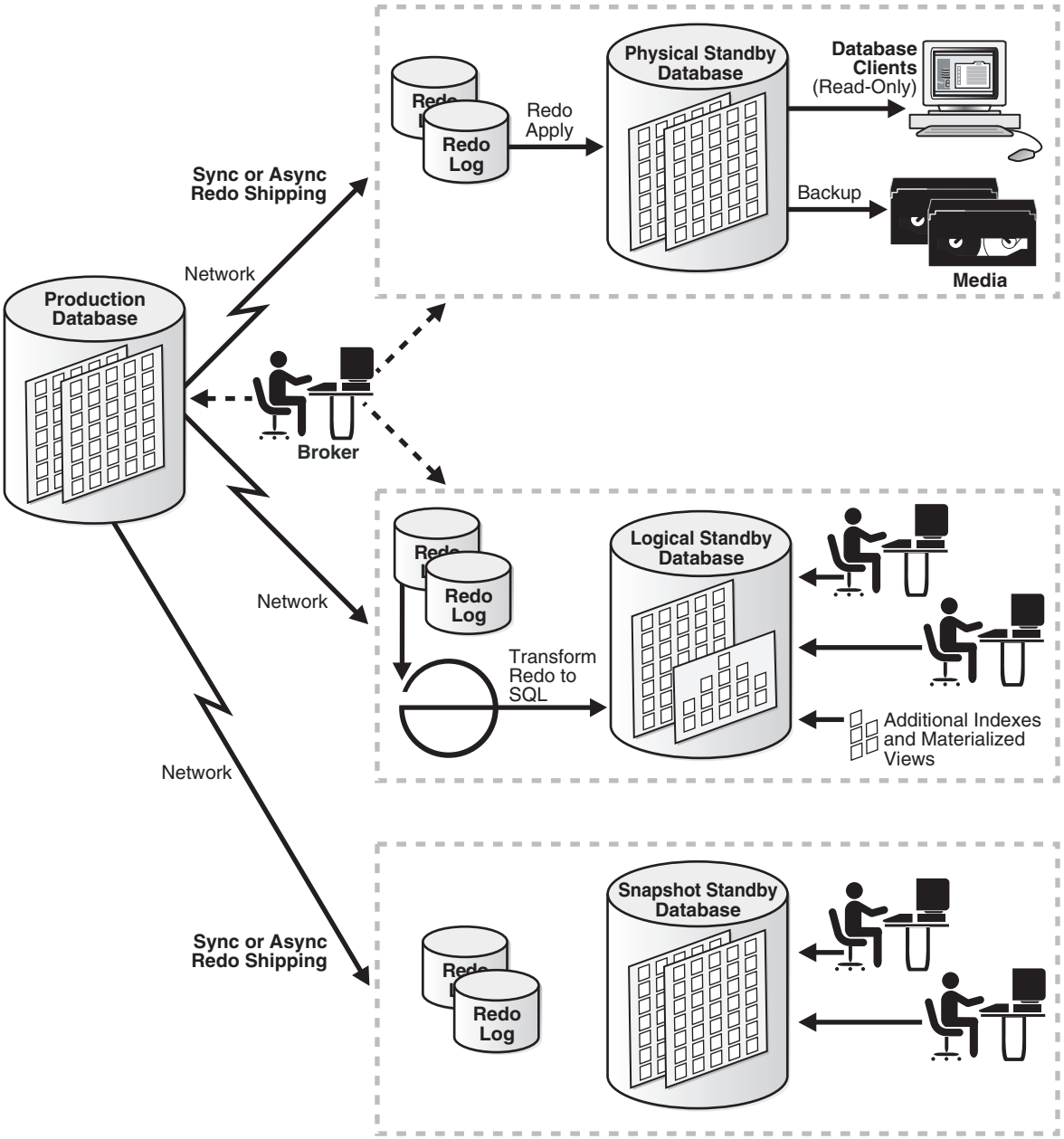
The production database transmits redo data (either synchronously or asynchronously) to redo log files at the physical standby database. Then, the redo data is applied from the logs to the physical standby database, which backs up the redo data to physical media.

At the logical standby database, the redo data is transformed into SQL statements, which are applied to the logical standby database. The logical standby database may contain additional indexes and materialized views. Clients are connected to the logical standby database and can work with its data.

At the snapshot standby database redo data is received, but it is not applied until the snapshot standby database is reconverted to a physical standby database. The figure shows users making local updates to the snapshot standby database. These updates are discarded when the snapshot database is reconverted to a physical standby database.

Also, see Figure 5–2 for another example of a multiple standby database environment.

Figure 7–7 Oracle Database with Oracle Data Guard on Primary and Multiple Standby Sites



See Also:

- *Oracle Data Guard Concepts and Administration* for more information about the various types of standby databases and to find out what data types are supported by logical standby databases
- *Oracle Database High Availability Best Practices* for configuration best practices
- The "Managing Data Guard Configurations Having Multiple Standby Databases - Best Practices" white paper, and other Oracle Data Guard white papers at

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

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

- A world-recognized financial institution uses two remote physical standby databases for continuous data protection after failover. If the primary system should fail, the first standby database becomes the new primary database. The second standby database automatically receives data from the new primary database, insuring that data is protected at all times.
- A nationally recognized insurance provider in the U.S. maintains two standby databases in the same Oracle Data Guard configuration: one physical standby and one logical standby database. Their strategy further mitigates risk by maintaining multiple standby databases, each implemented using a different architectures—Redo Apply and SQL Apply.
- A world-recognized e-commerce site uses multiple standby databases—a mix of both physical and logical databases—both for disaster recovery and to scale out read performance by provisioning multiple logical standby databases using SQL Apply.
- A global provider of information services to legal and financial institutions uses multiple standby databases in the same Oracle Data Guard configuration to minimize downtime during major database upgrades and platform migrations.

Also, for large data centers with a need to support many applications with Oracle Data Guard requirements, you can build an Oracle Data Guard hub to reduce the total cost of ownership.

7.1.5.3 Oracle Data Guard (Standby) Hub

With Database Server Grid and Database Storage Grid (described in Section 5.2 and Section 5.3), 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.

An Oracle 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, an Oracle 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

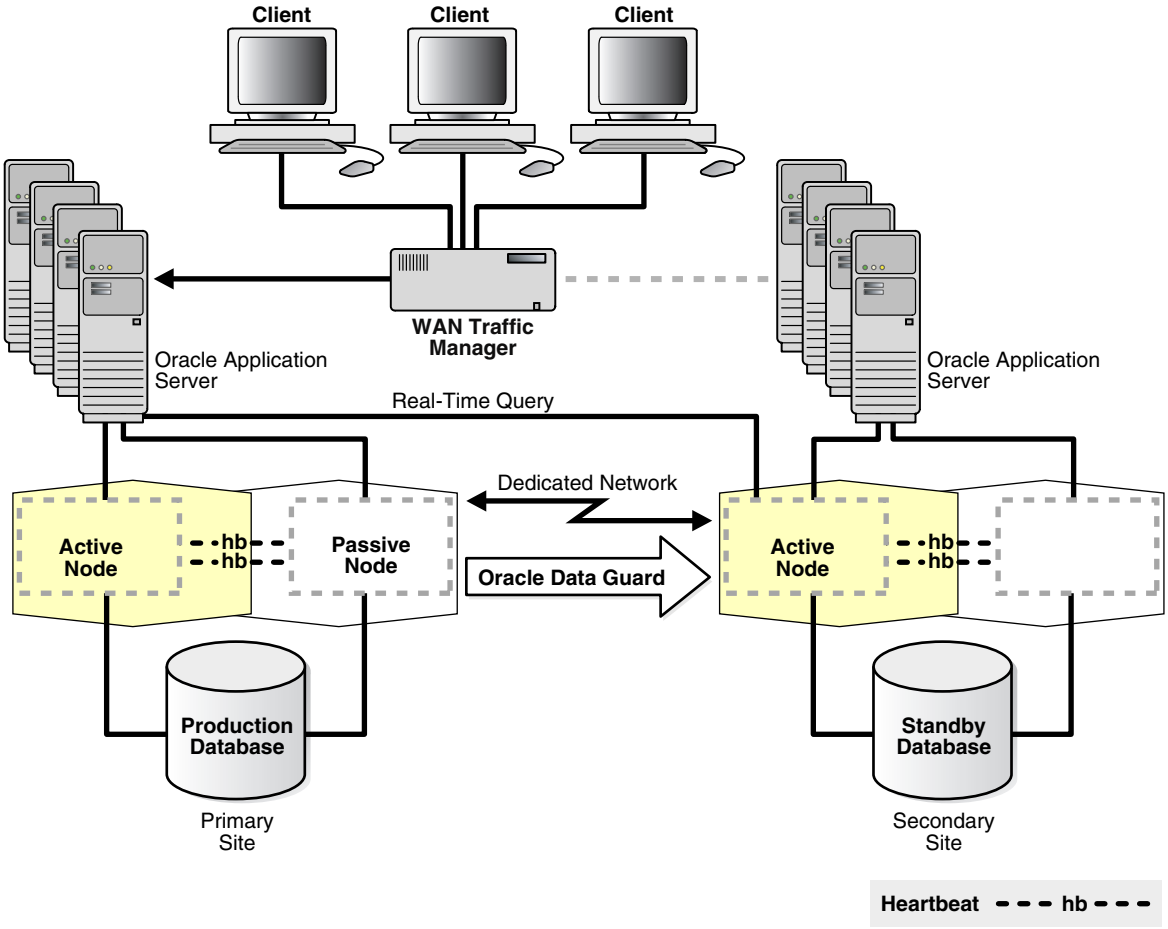
and server resources of a grid instead of building and managing individual servers for each application.

7.1.6 Oracle Database with Oracle Clusterware and Oracle Data Guard

If your business does not require the scalability and additional high availability benefits provided by Oracle RAC, but you still need all the benefits of Oracle Data Guard and cold cluster failover, then Oracle Database with Oracle Clusterware and Oracle Data Guard is a good compromise architecture. Oracle Clusterware cold cluster failover combined with Oracle Data Guard makes a tightly integrated solution in which failover to the secondary node in the cold cluster failover is transparent and does not require you to reconfigure the Oracle Data Guard environment or perform additional steps.

Figure 7–8 shows an Oracle Clusterware and Oracle Data Guard architecture that consists of a primary and a secondary site. Both the primary and secondary sites contain Oracle Application Servers, two database instances, and an Oracle database.

Figure 7–8 Oracle Clusterware (Cold Cluster Failover) and Oracle Data Guard



In Figure 7–8:

- The application servers on the secondary site are connected to the WAN traffic manager by a dotted line to indicate that they are not actively processing client requests at this time. (The application server on the secondary site can be active and processing client requests such as queries if the standby database is a physical

standby database with the Active Data Guard option enabled, or if it is a logical standby database.)

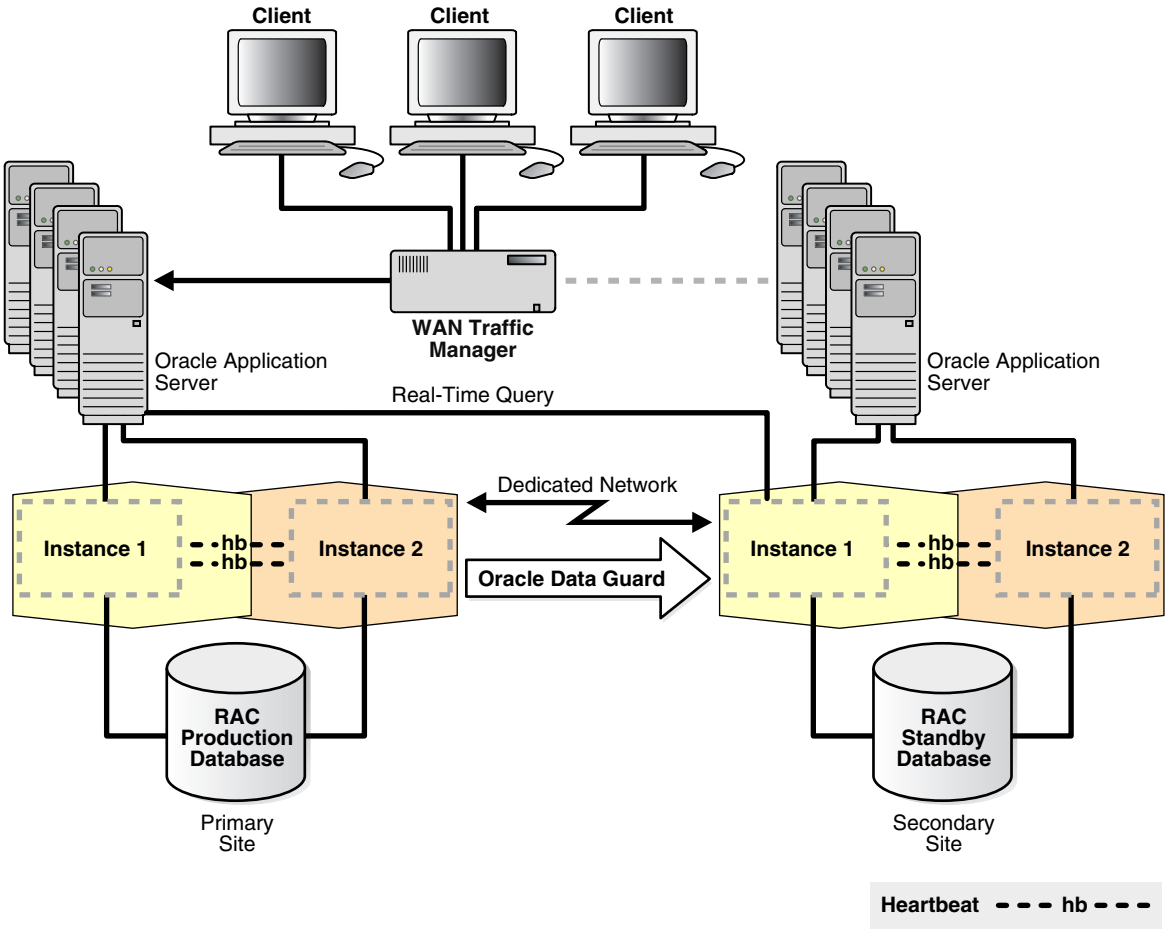
- Oracle Data Guard transmits redo data from the primary database to the secondary site to keep the databases synchronized.
- Oracle Clusterware manages the availability of both the user applications and Oracle databases.
- Oracle Clusterware provides tolerance of node failures, whereas Oracle Data Guard provides additional protection against data corruptions, lost writes, and database and site failures. (See Section 7.1.5 for a complete description.)
- Although cold cluster failover is not shown in Figure 7–8, you can configure it by adding a passive node on the secondary site.

7.1.7 Oracle Database with Oracle RAC and Oracle Data Guard

You can achieve the highest level of availability when using Oracle RAC and Oracle Data Guard and there is no need to make application changes to use these Oracle Database features. The combination of Oracle RAC and Oracle Data Guard provide the most comprehensive architecture for reducing downtime for scheduled outages and preventing, detecting, and recovering from unscheduled outages. This architecture is the recommended configuration for Maximum Availability Architecture (MAA).

To protect against site failures, the MAA recommends that Oracle RAC and Oracle Data Guard reside on separate systems (clusters) and data centers. Figure 7–9 shows the recommended MAA configuration, with Oracle Database, Oracle RAC, and Oracle Data Guard. 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.

Figure 7–9 Oracle Database with Oracle RAC and Oracle Data Guard - MAA



7.1.8 Oracle Database with Oracle Streams

Similar to using Oracle Data Guard in SQL Apply mode, Oracle Streams can capture database changes, propagate them to destinations, and apply the changes at these destinations. Oracle Streams is optimized for replicating data. Oracle Streams can capture changes at a source database, and the captured changes can be propagated asynchronously to replica databases. A logical copy configured and maintained using Oracle Streams is called a *replica*, not a logical standby database, because it provides many capabilities that are beyond the scope of the normal definition of a standby database.

You might choose to use Oracle Streams to configure and maintain a logical copy of your production database. Although using Oracle Streams might require additional work, it offers increased flexibility that might be necessary to meet specific business requirements.

Oracle Database with Oracle Streams provides granularity and control over what is replicated and how it is replicated. It supports bidirectional replication, data transformations, subsetting, custom apply functions, and heterogeneous platforms. It also gives users complete control over the routing of change records from the primary database to a replica database. Oracle Streams can capture data changes at the primary database or downstream at a replica database, thus enabling users to build hub-and-spoke network configurations that can support hundreds of replica databases.

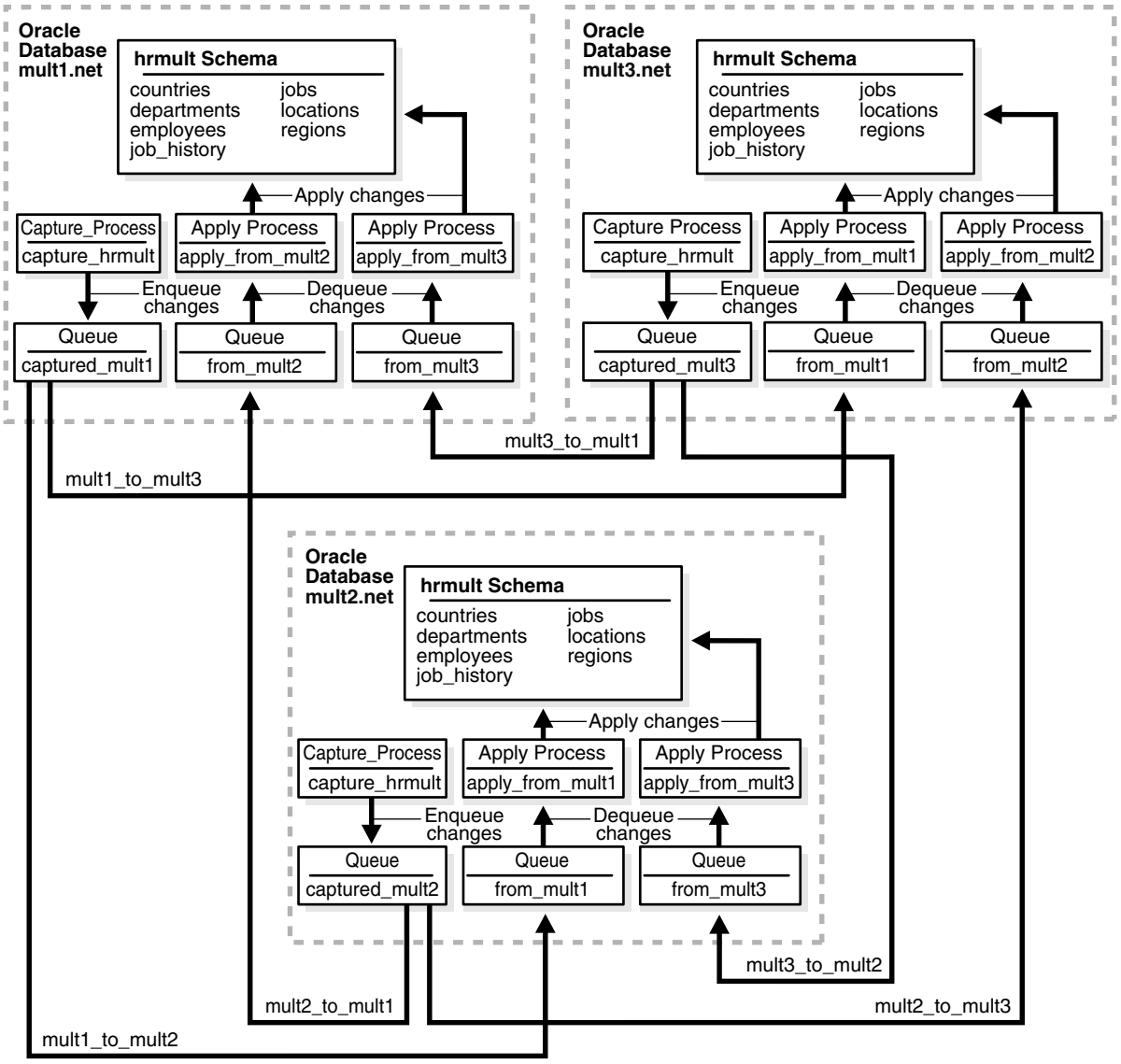
Consider using Oracle Database with Oracle Streams if one or more of the following conditions are true:

- Updates are required on both sites or databases, and the changes must be propagated bidirectionally.
- Site configurations are on heterogeneous platforms.
- Different character sets are required between the primary database and its replicas.
- Fine control of information and data sharing are required.
- More investment and expertise to build and maintain an integrated high availability solution is available.

Figure 7–10 shows a sample Oracle Database using Oracle Streams to replicate data for a schema among three Oracle databases. DML and DDL changes made to tables in the `hr` schema are captured at all databases in the environment and propagated to each of the other databases in the environment.

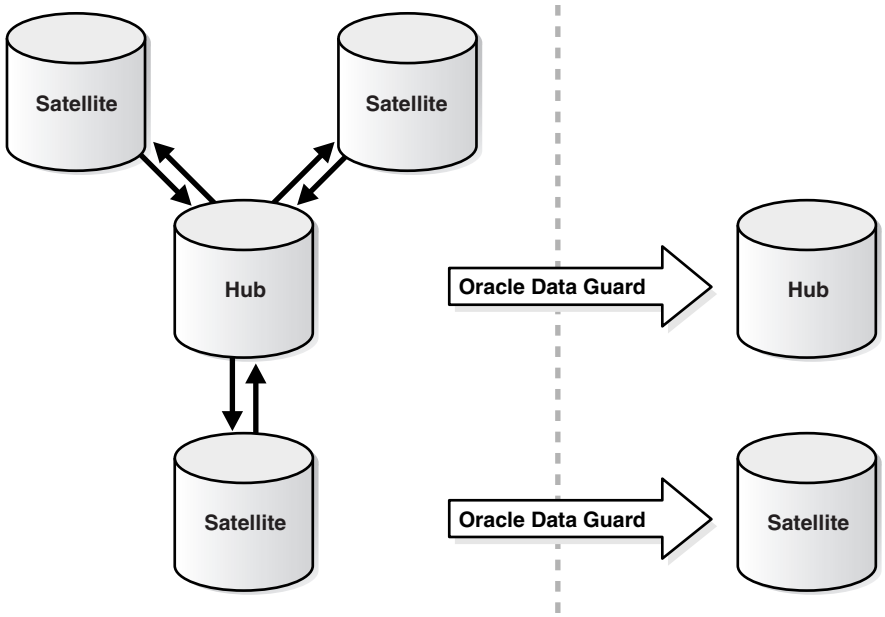
> **See Also:** *Oracle Streams Replication Administrator's Guide* for complete information about constructing multiple-source replication environments using Oracle Streams

Figure 7–10 Oracle Database with Oracle Streams Sharing Data from Multiple Databases



You can configure Oracle Streams with Oracle Data Guard to provide protection for the individual databases in the configuration. Figure 7–11 shows a hub-and-spoke network configuration in which Oracle Data Guard is providing additional data protection for the hub and one of the satellites.

Figure 7–11 Oracle Streams Hub-and-Spoke Network Configuration



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
- Manageability overhead (MO)
- Total cost of ownership (TCO) and return on investment (ROI)

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

Table 7–1 Frequency of Outages

Activity	Outage Frequency
Media or disk failures	High
Application patches	High
Application failures	High
Logical or user failures that manipulate logical data (DMLs and DDLs)	High
Data corruptions and lost writes	Medium
Computer failures	Medium
Database patches	Medium
Hardware patches and upgrades	Low

Table 7–1 (Cont.) Frequency of Outages

Activity	Outage Frequency
Operating system patches and upgrades	Low
Database or application upgrades	Low
Database failures	Low
Platform migrations	Very low
Site failures	Very low

Table 7–2 recommends architectures based on your business requirements for RTO, RPO, MO, scalability, and other factors.

Table 7–2 High Availability Architecture Recommendations

Consider Using ...	Business or Application Impact ...
Oracle Database with Oracle Clusterware (Cold Cluster Failover)	■ Maximum RTO for instance or node failure is in minutes. ■ MO is low. ■ ROI is low. ■ Rolling upgrade and patch capabilities for Oracle Clusterware with zero database downtime
Oracle Database with Oracle Real Application Clusters (Oracle RAC)	■ Maximum RTO for instance or node failure is zero for the database[1]. ■ MO is medium. ■ ROI is high. ■ Workload Management and Quality of Service Management ■ 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[2]. ■ ROI is medium. ■ Additional protection from data center failure with special considerations that are documented in Section 7.1.4 ■ 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

Table 7–2 (Cont.) High Availability Architecture Recommendations

Consider Using ...	Business or Application Impact ...
Oracle Database with Oracle 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. ■ MO is low. ■ ROI is high. ■ Rolling upgrade for system, clusterware, database, and operating system ■ Off-load read-only, reporting, testing and backup activities to the standby database. For physical standby databases, 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 For logical standby databases, this solution: ■ Provides the simplest form of one-way logical replication ■ 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 Clusterware and Oracle Data Guard	■ All of the business benefits of Oracle Clusterware (cold cluster failover) and Oracle Data Guard ■ MO is low. ■ ROI is medium.
Oracle Database with Oracle RAC and Oracle Data Guard	■ All of the business benefits of Oracle RAC and Oracle Data Guard ■ MO is medium. ■ ROI is high.

Table 7–2 (Cont.) High Availability Architecture Recommendations

Consider Using ...	Business or Application Impact ...
Oracle Database with Oracle Streams	■ Maximum RTO for instance or node failure is in seconds to minutes. ■ Maximum RTO for data corruption, cluster, database, or site failures is in seconds to minutes. ■ MO is high[2]. ■ ROI is high. ■ Rolling upgrade for system, clusterware, operating system, database, and application. ■ Support for bidirectional replication and updating anything and anywhere. ■ Support for heterogeneous platforms, versions, and character sets. ■ Support for fine-grained, n-way multimaster, hub-and-spoke, or many-to-one replication architectures. ■ Flexible propagation and management of data, transactions, and events. ■ With Oracle RAC integration, database scalability is possible.

[1] Database is still available, but a portion of the application connected to the failed system is temporarily affected.

[2] Architectures for which the MO is high might require additional time and expertise to build and maintain, but offer increased flexibility and capabilities required to meet specific business requirements.

Table 7–3 identifies the additional capabilities provided by the architectures that build on Oracle Database and attempts to label each architecture with its greatest strengths.

Table 7–3 Additional Capabilities of High Level Oracle High Availability Architectures

Oracle High Availability Architecture	Key Characteristics and Additional Capabilities
Oracle Database (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 Automatic Storage Management and Automatic Storage Management Cluster File System (ACFS) tolerate storage failures and optimize storage performance and utilization. ▪ 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 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
Oracle Database with Oracle Clusterware (Cold Cluster Failover)	▪ All of the benefits of Oracle Database ▪ Automatic and fast failover for computer failure ▪ Minimum rolling upgrade capabilities for system, clusterware, and operating system[3]
Oracle Database with Oracle RAC on Extended Clusters Database Grid with site failure protection	▪ All of the benefits of Oracle Database ▪ Protection from site failure
Oracle Database with Oracle Data Guard Simplest high availability, data protection, and disaster-recovery solution	▪ All of the benefits of Oracle Database ▪ Automatic and fast failover features protect against computer failure, storage failure, data corruption, and database failures ▪ Protection from site failure ▪ Rolling upgrade to update system, clusterware, database, and operating system software[1] ▪ Off-load backups to the standby database ▪ Off-load read and reporting workload to the standby database ▪ Comprehensive lost write protection

Table 7–3 (Cont.) Additional Capabilities of High Level Oracle High Availability Architectures

Oracle High Availability Architecture	Key Characteristics and Additional Capabilities
Oracle Database with Oracle Clusterware and Oracle Data Guard High availability solution with added data and disaster recovery protection.	▪ The sum of benefits of Oracle Clusterware with Oracle Data Guard
Oracle Database with Oracle RAC and Oracle Data Guard Best high availability, data protection and disaster-recovery solution with scalability built in	▪ The sum of benefits of Oracle RAC with Oracle Data Guard
Oracle Database with Oracle Streams[2] Bidirectional replication and information management	▪ Replica database (or databases) are available for read/write use ▪ Provides heterogeneous platform support ▪ Fast failover for computer failure and storage failure ▪ Protection from site failure ▪ Minimizes downtime for computer or site maintenance and database and application upgrades
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) with integrated Oracle client failover ▪ FAN with integrated Oracle client failover for pooled resources and third-party vendor middle tiers ▪ FAN with integrated Oracle client failover, including Java applications using UCP with Oracle RAC and Oracle Data Guard. Applications can easily mask failures to the end user. ▪ Workload Management and Quality of Service Management ▪ Zero downtime with Grid Control provisioning ▪ Rolling upgrade for system, clusterware, operating system, CPUs, and some Oracle interim patches[3]
Oracle Database with Oracle RAC on Extended Clusters Database Grid with site failure protection	▪ All of the benefits of Oracle Database ▪ Protection from site failure
Oracle Database with Oracle Data Guard Simplest high availability, data protection, and disaster-recovery solution	▪ All of the benefits of Oracle Database ▪ Automatic and fast failover for computer failure, storage failure, data corruption, for configured ORA- errors or conditions and database failures ▪ Protection from site failure ▪ Rolling upgrade for system, clusterware, database, and operating system[4] ▪ Off-load backups to the standby database ▪ Off-load read and reporting workload to the standby database ▪ Only comprehensive lost write protection

Table 7–3 (Cont.) Additional Capabilities of High Level Oracle High Availability Architectures

Oracle High Availability Architecture	Key Characteristics and Additional Capabilities
Oracle Database with Oracle Clusterware and Oracle Data Guard High availability solution with added data and disaster recovery protection.	▪ The sum of benefits of Oracle Clusterware with Oracle Data Guard
Oracle Database with Oracle RAC and Oracle Data Guard Best high availability, data protection and disaster-recovery solution with scalability built in	▪ The sum of benefits of Oracle RAC with Oracle Data Guard
Oracle Database with Oracle Streams[5] Bidirectional replication and information management	▪ Replica database (or databases) available for read/write use ▪ Heterogeneous platform support ▪ Fast failover for computer failure and storage failure ▪ Protection from site failure ▪ Minimum downtime for computer or site maintenance and database and application upgrades

1 Rolling upgrades with Oracle Data Guard incur minimal downtime.
2 The initial investment to build a robust solution is well worth the long-term flexibility and capabilities that Oracle Streams delivers to meet specific business requirements.
3 Rolling upgrades with Oracle Clusterware and Oracle RAC incur zero downtime.
4 Rolling upgrades with Oracle Data Guard incur minimal downtime.
5 The initial investment to build a robust solution is well worth the long-term flexibility and capabilities that Oracle Streams delivers to meet specific business requirements.

Table 7–4 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

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

Table 7–4 Attainable Recovery Times for Unplanned Outages

Outage Type	Oracle Database	Cold Cluster	Oracle RAC and RAC on Extended Clusters	Oracle Data Guard	Oracle RAC and Oracle Data Guard	Oracle Streams
Site failure	Hours to days	Hours to days	No downtime[4] if the outage is limited to one building Hours to days if the outage affects both building	Seconds to a minute[1]	Seconds to a minute[1]	No downtime[2]
Computer failure	Minutes to hours[3]	Minutes	No downtime[4]	Seconds to a minute	No downtime[4]	No downtime[4]

Table 7–4 (Cont.) Attainable Recovery Times for Unplanned Outages

Outage Type	Oracle Database	Cold Cluster	Oracle RAC and RAC on Extended Clusters	Oracle Data Guard	Oracle RAC and Oracle Data Guard	Oracle Streams
Storage failure	No downtime[5]	No downtime[5]	No downtime[3]	No downtime[3]	No downtime[3]	No downtime[3]
Human error	< 30 minutes[6]	< 30 minutes[6]	< 30 minutes[4]	< 30 minutes[4]	< 30 minutes[4]	< 30 minutes[4]
Data corruption	Potentially hours[7]	Potentially hours[7]	Potentially hours[7]	Zero downtime[8]	Zero downtime	Seconds to a minute

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 a 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 the age of the backup used for recovery and the number of log changes scanned to make the corrupt data consistent with the database.

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 7–5 compares the attainable recovery times of each Oracle high availability architecture for all types of planned downtime.

Table 7–5 Attainable Recovery Times for Planned Outages

System Change or Data Change	Outage Type	Oracle Database	Oracle RAC	Oracle Data Guard	MAA	Oracle Streams
System change - Dynamic Resource Provisioning	--	No downtime	No downtime	No downtime	No downtime	No downtime
System change - Rolling Upgrade	System-level upgrade	Minutes to hours	No downtime	Seconds to 5 minutes	No downtime	No downtime
System change - Rolling Upgrade	Clusterwide or sitewide upgrade	Minutes to hours	Minutes to hours	Seconds to 5 minutes	Seconds to 5 minutes	No downtime[1]
System change - Rolling Upgrade	Storage Migration	No downtime[2]	No downtime[2]	No downtime[2]	No downtime[2]	No downtime[2]
System change - Rolling Upgrade	Database one-off patch	Minutes to an hour	No downtime[3]	Seconds to 5 minutes	No downtime[3]	No downtime

Table 7–5 (Cont.) Attainable Recovery Times for Planned Outages

System Change or Data Change	Outage Type	Oracle Database	Oracle RAC	Oracle Data Guard	MAA	Oracle Streams
System change - Rolling Upgrade	Database patch set and version upgrade	Minutes to hours	Minutes to hours	Seconds to 5 minutes	Seconds to 5 minutes	No downtime[1]
System change - Rolling Upgrade	Platform migration	Minutes to hours	Minutes to hours	Minutes to hours	Minutes to hours	No downtime[1]
Data change	Online Reorganization and Redefinition	No downtime	No downtime	No downtime[4]	No downtime[4]	No downtime[4]
Application changes	Outages that are fixed by Editions	No downtime	No downtime	No downtime	No downtime	No downtime

[1] Applications (or a portion of an application) connected to the system that is being maintained may be temporarily affected.

[2] Oracle ASM automatically rebalances stored data when disks are added or removed while the database remains online. For storage migration, you are required to use both storage arrays by Oracle ASM temporarily.

[3] For qualified one-off patches only

[4]Tables can be reorganized online using the `DBMS_REDEFINITION` package. However, the online changes are not supported by SQL Apply or data capture, and therefore the effects of this subprogram are not visible on the logical standby database or replica database. For more information, see *Oracle Data Guard Concepts and Administration* or the *Oracle Streams Replication Administrator's Guide*.

7.3 Integrating Application Server High Availability

Flexible and automated high availability solutions ensure that applications you deploy on Oracle Application 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 Application Server High Availability Architectures
- Redundant Architectures
- High Availability Services in Oracle Application Server

7.3.1 Oracle Application Server High Availability Architectures

Oracle Application 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 Application Server local high availability architectures include several active-active and active-passive architectures for the OracleAS middle-tier and the OracleAS Infrastructure. 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.

Building on top of the local high availability solutions is the Oracle Application Server disaster recovery solution. This unique solution combines the proven Oracle Data Guard technology in Oracle Database with advanced disaster recovery technologies in the application realm to create a comprehensive disaster recovery solution for the entire application system. Disaster recovery solutions typically set up two homogeneous sites, one active and one passive. Each site is a self-contained system. The active site is generally called the production site, and the passive site is called the standby site. During normal operation, the production site services requests; in the event of a site failover or switchover, the standby site takes over the production role and all requests are routed to that site. To maintain the standby site for failover, not only must the standby site contain homogeneous installations and applications, data and configurations must also be synchronized constantly from the production site to the standby site. Oracle Application Server instances can be installed in either site as long as they do not interfere with the instances in the disaster recovery setup. Configurations and data must be synchronized regularly between the two sites to maintain homogeneity.

7.3.2 Redundant Architectures

Oracle Application 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 Application 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 Application Server. The configuration can be an active-active configuration using Oracle Application Server Cluster or an active-passive configuration using Oracle Application Server Cold Cluster Failover.

7.3.3 High Availability Services in Oracle Application Server

The *Oracle Application Server High Availability Guide* describes the following high availability services in Oracle Application Server in detail:

- Process death detection and automatic restart
- Configuration management
- State replication
- Server load balancing and failover
- Backup and recovery
- Disaster recovery

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 Application Server, Oracle Enterprise Manager, and Oracle Applications on the MAA Web site at:

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

Glossary

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 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 Active Data Guard option may also be referred to as "real-time query" in other documentation.

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 is unable to 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 11*g* with Oracle RAC and Oracle Database 11*g* with Data Guard.

The MAA environment consists of a site containing an Oracle RAC primary database and a second site containing a 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 server processes

The Data Guard network server processes, also referred to as LNS*n* 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.

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.

Index

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W