

Maximum Availability Architecture: Overview

An Oracle White Paper
October 2002

Maximum Availability Architecture: Overview

| | |
|---|----|
| Abstract | 3 |
| Introduction..... | 3 |
| Architecture overview | 3 |
| Secondary Site..... | 4 |
| Highly Available Database..... | 5 |
| Highly Available Application Tier | 7 |
| Network Infrastructure | 8 |
| Sound Best Practices | 9 |
| Configuration Best Practices | 9 |
| Operational Best Practices..... | 9 |
| Outages and Solutions..... | 10 |
| Restoring Fault Tolerance..... | 11 |
| Conclusion..... | 12 |

Maximum Availability Architecture: Overview

ABSTRACT

Oracle and its partners provide all the ingredients and components to build a highly available architecture. However, choosing and implementing the architecture that best fits your availability requirements can be a daunting task. This architecture must encompass redundancy across all components, achieve fast client failover for all types of outages, and provide protection from user errors, corruptions, and site disasters, while being easy to deploy, manage, and scale.

This paper describes a technical architecture that removes the complexity of designing a highly available (HA) architecture for your business. *Maximum Availability Architecture* (MAA) is a straightforward, redundant, and robust architecture that prevents, detects, and recovers from different outages within a small mean time to recovery (MTTR), as well as preventing or minimizing downtime for maintenance. This architecture is a complete solution consisting of proven Oracle HA technology and exemplifies the Oracle Unbreakable architecture. It is validated by the Oracle Server Technologies High Availability Systems Group and is being validated and deployed in numerous customer sites around the world.

INTRODUCTION

This paper provides an overview of the Maximum Availability Architecture. The complete MAA description is presented in an Oracle white paper titled *Maximum Availability Architecture*, which is available on the Oracle Technology Network at http://otn.oracle.com/deploy/availability/pdf/MAA_WP.pdf. *Maximum Availability Architecture* includes the following sections:

- *Overview of MAA and its components* - provides an executive view of the architecture and its components
- *Configuration best practices in building MAA* - describes what needs to be implemented and why
- *Detailed descriptions of outages and solutions* - justifies the architecture by providing the best solutions for a list of scheduled and unscheduled outages
- *Restoring full database fault tolerance* – explains how to restore complete high availability to the database after a failover operation or after resolving a database outage

Maximum Availability Architecture provides detailed configuration best practices and solutions to help prevent and repair a wide range of different outages across the entire architecture. The core content focuses on configuring and maintaining a highly available database within a three-tier architecture, leveraging the latest, validated high availability Oracle features.

This architecture was validated using Oracle9i Release 2 and Oracle9iAS Release 2.

ARCHITECTURE OVERVIEW

MAA provides a straightforward, redundant, and robust architecture that prevents different outages or recovers from an outage within a small mean time to recovery (MTTR). The goal is that most outages have no impact or minimal impact to

availability while catastrophic outages can be repaired in less than 30 minutes. It encompasses the following main components, illustrated in figure 1:

- Secondary Site
- Highly Available Database Tier
- Highly Available Application Tier
- Redundant Network Infrastructure

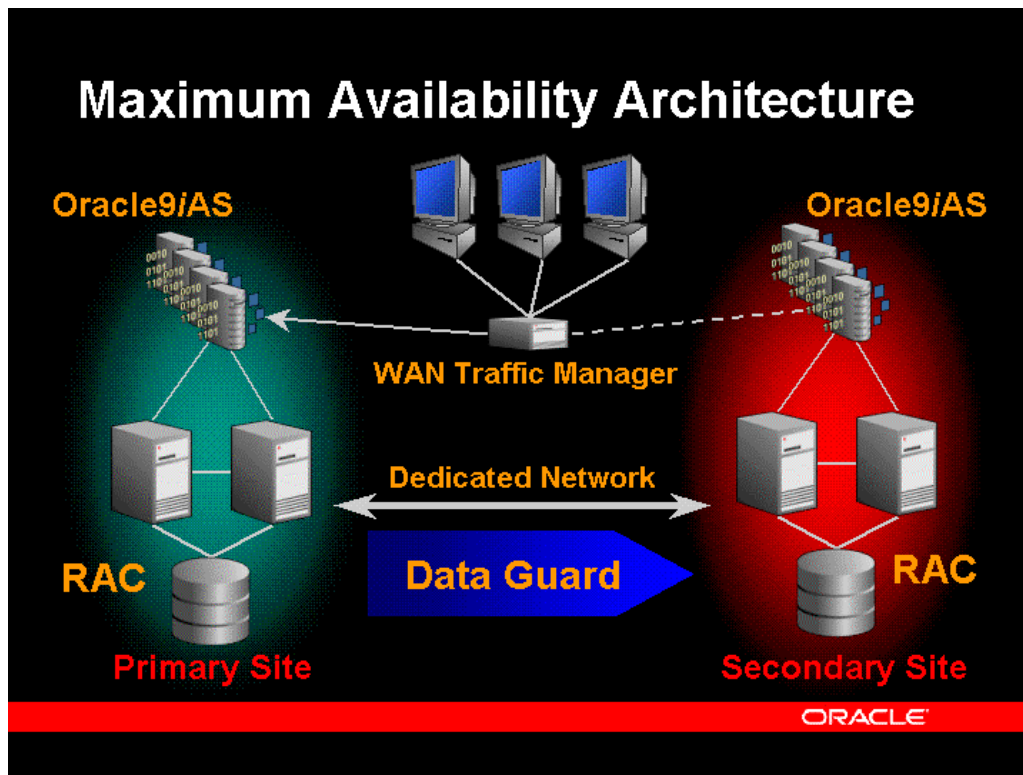


Figure 1: Maximum Availability Architecture Overview

Secondary Site

The principal architectural characteristic of MAA is a primary site and a secondary site that are identically configured. Each site consists of redundant components and redundant routing mechanisms, so that requests are always serviceable, even in the event of a failure. Client requests are always routed to the site playing the production role. After a switchover or failover operation occurs, client requests are re-routed to the secondary site that assumes the production role. Each site contains a set of application servers or mid-tier servers. The site playing the production role contains the production database using RAC to protect from host and instance failures. The site playing the standby role contains a physical standby database managed by Data Guard. Data Guard Role Management Services allow for a change in the role of a database from a physical standby database to a primary database, or from a primary database to a physical standby database using either a switchover or a failover operation. There are two defined roles:

- The production role acts as the production database
- The standby role acts as the physical standby database

Initially in Figure 1, the primary site contains the production database and plays the production role, and the secondary site contains the physical standby and plays the standby role. The roles switch after a scheduled switchover operation or an unplanned failover operation. A detailed description of these operations and when they should occur appears in the outage and solution section of *Maximum Availability Architecture*. Even though roles can change, the primary and secondary site labels are constant.

You need to ensure that the application failover policies and infrastructure allow you to fail over to the secondary site within an acceptable MTTR while maintaining tolerable performance at the site. *Maximum Availability Architecture* advocates identical site configurations to ensure that performance is not sacrificed. In addition, this allows consistent processes and procedures between sites, making operational tasks much easier to maintain and execute. Furthermore, identical site configurations ensure that upgrades and software changes on the primary site are also propagated to the secondary site and vice versa. Customers should repeat primary site upgrade steps on the secondary site or copy the changes directly to the secondary site. In all cases, remote software synchronization needs to be maintained manually or with third party solutions to keep software synchronized.

A secondary site that is identical to the primary site allows predictable performance and response time after failing over or switching over from the primary site. An identical secondary site also allows for procedures, processes, and overall management to be the same between sites that are set up identically. The secondary site is leveraged for primary site-wide failures, all unplanned outages that are not resolved automatically or quickly on the primary site, and many planned outages when maintenance is required on the primary site.

Highly Available Database

Real Application Clusters (RAC) and Oracle Data Guard provide the basis of the MAA solution. RAC allows multiple database instances to share the same database, providing optimal performance, scalability, and availability gains. Using a Data Guard physical standby database provides disaster recovery and protection from user error and data corruption. Data Guard also contains a management infrastructure that encompasses remote archiving, managed recovery, physical standby databases, logical standby databases, and the Data Guard Broker GUI and command-line interface. In MAA, Data Guard provides automatic remote archiving, managed recovery, and role management between the production database and the physical standby database at the secondary site. A physical standby database is a copy of the production database that is updated by applying the production database's redo data. The physical standby database is a duplicate database that is mounted and in recovery mode. It is configured with a lag to prevent the application of user error or corruption in the production database to the standby database.

Real Application Clusters

RAC uses two or more nodes or machines, each running an Oracle instance that accesses a single database residing on shared-disk storage. In a RAC environment, all active instances can concurrently execute transactions against the shared database. RAC automatically coordinates each instance's access to the shared data to provide data consistency and data integrity.

RAC provides the following key benefits:

- Availability – provides near-continuous access to data with minimal interruption from hardware and software component failures
- Scalability – allows nodes to be added to the cluster to increase processing capabilities without having to redistribute data or alter the user application
- Manageability – provides a single system image to manage

RAC allows continuous data *availability* in the event of component, instance, or node failure. If an instance or node fails, the surviving instances automatically perform recovery for the failed instance and continue to provide database service. User data is always accessible if there is at least one available instance running in the cluster. Along with effectively handling unscheduled outages (e.g., instance or node failures), RAC gives the administrator the ability to perform scheduled maintenance on a subset of nodes or components of the cluster while continuing to provide service to users.

RAC automatically harnesses the processing power of additional nodes as they are brought into the cluster, thus providing *scalability*, potentially without downtime. With RAC's Cache Fusion architecture, it is not necessary to re-partition data or modify an application to take advantage of additional CPU power or additional I/O and network bandwidth made available when nodes are added to or removed from the cluster.

RAC also can automatically balance new database connection requests among the available instances, based on lowest processing load and fewest connections. Because of an instance's ability to provide load data to listeners and to cross-register with remote listeners, each listener is aware of all services, instances, dispatchers, and their current loads regardless of their location. Thus a listener can send an incoming client request for a specific service to the least-loaded node, instance, or dispatcher.

A key component to RAC availability and scalability is the private interconnect. The interconnect is a communication facility that links the nodes in the cluster, routing messages, data, and other cluster communications traffic to coordinate each node's access to shared resources. For high availability, the interconnect must be redundant such that a single link failure, from a failed adapter, cable or switch, does not isolate one node from the rest of the cluster. To ensure scalability, particularly with the Cache Fusion architecture, the interconnect must be a high-bandwidth, low-latency link. Ideally, the cluster can fully utilize the redundant links and balance loads across the multiple interconnect paths.

When maintaining a RAC environment, since it is a single database accessed by multiple instances, a single system image is preserved across the cluster for all database operations, which simplifies *manageability*. DBAs perform configuration, HA operations, recovery, and monitoring functions once. Oracle then automatically distributes the management functions to the appropriate nodes. This means the DBA manages *one* virtual server.

Implementing the Real Application Clusters Guard (RACG) feature of RAC provides an enhanced HA solution by coupling the availability advantages of RAC with integrated monitoring, connection failover, and hardware clustering. RACG is intended for environments with the strictest availability requirements.

Oracle Data Guard

Oracle Data Guard is software that maintains a real-time copy of a production database, called a standby database. In MAA, the standby database is kept on the site with the standby role and can be used for disaster recovery. However, if the sites are identical and the physical location of the production database is transparent to the user, the production and standby roles can switch between sites easily for many different types of unplanned or planned outages, in addition to providing disaster recovery.

Data Guard manages the two databases by providing online log transport services, managed recovery, switchover and failover features. In MAA, the production database utilizes RAC, and the physical standby database resides on an identical cluster at the secondary site. Initially, the physical standby database resides on the secondary site. However, the primary and secondary sites can switch roles easily with Data Guard switchover for planned outages, and Data Guard failover for unplanned outages. A secondary site that is identical to the primary site allows predictable performance and response time after failing over or switching over from the primary site. An identical secondary site also allows for procedures, processes, and overall management to be the same between sites that are set up identically. The secondary site is leveraged for all unplanned outages that are not resolved automatically or quickly on the primary site and for many planned outages when maintenance is required on the primary site.

Data Guard with physical standby database provides the following benefits:

- *Availability* – provides protection from human errors, data failures and primary site failures, provides switchover operations for primary site maintenance, and different database protection modes to minimize or create no data loss environments
- *Manageability* – provides framework for log transport services and managed standby recovery, contains role management services such as switchover and failover and allows you to offload backups and read only activities from the production database

A specified delay of redo application at the standby database is configured to ensure that a logical corruption or error, such as dropping a table, will be detected before the change is applied to the standby database. In addition, adequate monitoring and detection also need to be in place to ensure errors are detected within the specified lag interval. The standby database can be configured with a zero data or transaction loss capability. Using the standby database, most database failures are resolved faster than by using on-disk backups since the amount of database recovery is dramatically reduced.

Highly Available Application Tier

The application tier provides application level services to clients. At both the primary and secondary site, MAA has a separate application tier comprised of a redundant set of servers that run Oracle9iAS Containers for J2EE (OC4J) front ended by Oracle9iAS Web Cache cluster and load balancers as depicted in Figure 2.

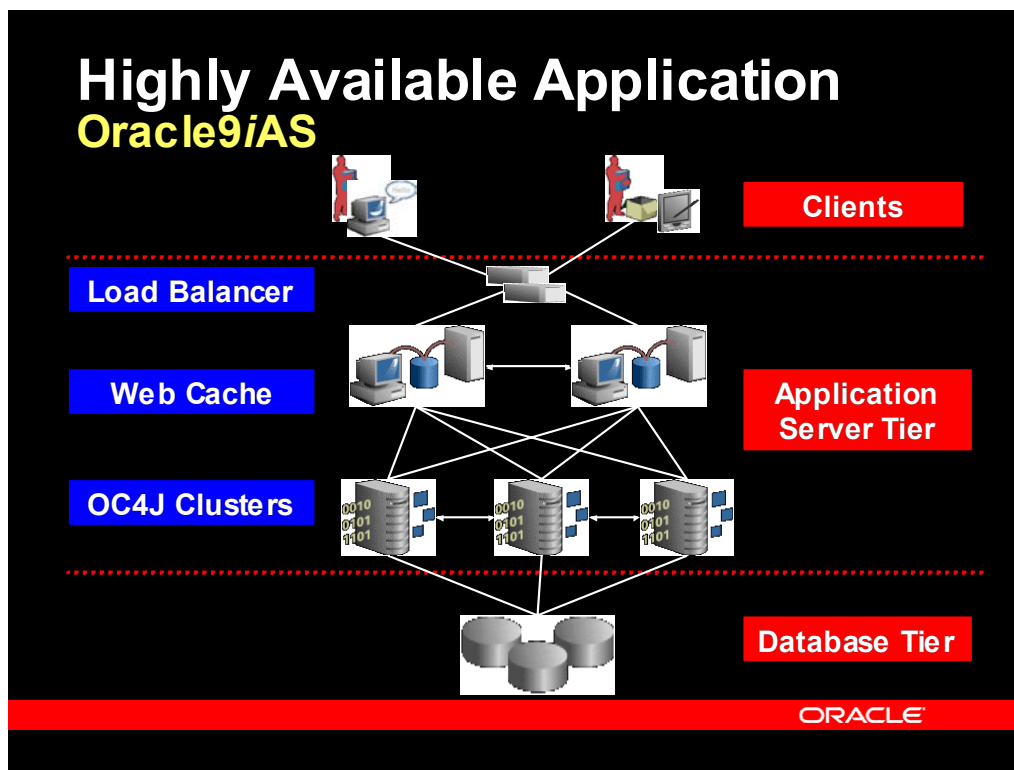


Figure 2: Highly Available Application Tier

The servers that run OC4J provide the application service to the clients and are the building block of the application tier. OC4J is J2EE compliant and includes a JSP translator, a Servlet container, an EJB container, and several other Java specifications. OC4J clustering connects servers running individual containers to act in tandem to provide greater

scalability and availability than a single instance can provide. Additional OC4J instances can be added or removed transparently to increase scalability or do system maintenance. The OC4J cluster at the primary site is identical in terms of hardware, OS, application server software, and application-specific software at the secondary site. They are configured similarly in all respects.

In addition to OC4J clustering, having Oracle9iAS Web Cache Cluster and load balancers in front of the OC4J clusters can increase the performance, scalability and availability of a web site. By storing frequently accessed URL's in memory, Web Cache eliminates the need to repeatedly process those requests on the application web server. In addition, multiple instances of Web Cache can be configured to run as members of a cache cluster, providing higher scalability of the website - by allowing more content to be cached and allowing for more client connections; and by providing higher availability - by detecting failures and failing over of caches. The Web Cache Clusters are configured similarly in all respects at the primary and the secondary site. The network infrastructure directs all client requests to the application servers at the primary site containing the production role. The application tier accesses a RAC database at the primary site in the database tier using Oracle Net Services (optionally using Oracle Internet Directory servers to lookup a database service).

Because each OC4J instance and cache cluster member is redundant on multiple sets of machines, problems and outages within a site are transparent to the client. The Oracle Process Management and Notification (OPMN) monitoring infrastructure ensures near uninterrupted availability of an application's services by automatically detecting and restarting failed components of the application tier or the Web Cache.

In Figure 1, if the primary site fails, then the servers at the secondary site need to be activated. The network then directs all subsequent client requests to the application tier at the secondary site. The client redirection is discussed in "Network Infrastructure" and the "Application Failover Configuration Best Practices" section of *Maximum Availability Architecture*. The database at the secondary site that was in the standby role takes over the production role and becomes the active production data server under the new configuration.

Network Infrastructure

All network components (router, firewalls, load balancers) are implemented in a fault tolerant fashion so that there is no single point of failure. The network must have the ability to automatically reroute traffic across redundant links and devices, providing at least one alternative route around any single failed component. This way the application is always accessible. This also allows you to take the network components offline, one at a time, and minimize planned downtime.

In MAA, two identically configured sites are created to provide a high availability environment - the primary site and the secondary site. Traffic is directed to the secondary site when the primary site cannot provide the service due to a planned or unplanned outage. In the event of a primary site outage, a wide-area traffic manager is used to direct traffic to the secondary site.

Load balancers disguise the application server instances and present a single IP address to the end user clients. The load balancers receive all client requests and then evenly distribute the load across the middle-tier application servers. If one of the application servers fails, the load balancer redistributes all subsequent client requests to any (appropriate) surviving application servers. A backup load balancer is also required for redundancy. With two load balancers, one is configured as a standby load balancer and will become active only if the primary load balancer becomes unavailable.

The application servers use configuration information from Oracle Net Services to connect to the database. This information can be stored in a centralized Lightweight Directory Access Protocol (LDAP)-compliant directory server (such as Oracle Internet Directory, which should also be redundant). This is easier to maintain than a local tnsnames.ora files on each host.

SOUND BEST PRACTICES

An architecture that contains all the necessary hardware and software features without sound best practices will ultimately fail to meet availability service levels. Best practices provide the greatest impact on availability by preventing outages, detecting outages quickly, recovering from outages within a tolerated MTTR, and restoring fault tolerance promptly. Best practices have been categorized into the following areas:

- Configuration best practices
- Operational best practices
- Outages and solutions
- Restoring fault tolerance

Each category is introduced below. Refer to the Maximum Availability Architecture paper for additional details.

Configuration Best Practices

Proper configuration is crucial for deploying and operating a highly available environment that meets service levels and MTTR targets. The Configuration Best Practices section of *Maximum Availability Architecture* contains specific details for Oracle software and guidelines for non-Oracle software and hardware. Non-Oracle software includes storage, hardware and operating systems, and network.

MAA describes specific Oracle features to use, parameters to set, and points to consider when multiple, valid HA options are available, with a heavy focus placed on database configuration best practices that affect the performance, availability and MTTR of your system. These practices are identical for the production and standby databases. Some of the options may affect performance levels, but are considered necessary to reduce or avoid outages. The minimal performance impact is outweighed by the reduced risk of corruption or the performance improvement for recovery.

For example, *Maximum Availability Architecture* recommends using the fast-start checkpointing feature to control the amount of time required to recover from an instance failure. A thorough discussion is presented highlighting the points to consider when choosing a proper setting for the `fast_start_mttr_target` INIT.ORA parameter, along with test results to validate the recommendations.

Configuration also covers guidelines for non-Oracle software and hardware, particularly as they directly affect Oracle software. For example, MAA recommends that disks in the storage array be configured using the Stripe and Mirror Everything (SAME) methodology for a simple, efficient, and highly available storage configuration. Another example is utilizing a storage array that supports Oracle's Hardware Assisted Resilient Data (HARD) initiative for the highest level of data protection. In order to prevent corruptions before they happen, Oracle tightly integrates with advanced storage devices to create a system that detects and eliminates corruptions before they happen. Oracle has worked with leading storage vendors to implement Oracle's data validation and checking algorithms in the storage devices themselves.

Operational Best Practices

Operational best practices are categorized into logistical and technical components. The logistical component includes those practices that are the foundation of managing the IT infrastructure and are geared towards process and policy management. Some of the logistical best practices include having sound change management, backup and recovery planning, disaster recovery planning, scheduled outage planning, adequate staff training, thorough documentation practices, and sound security policies and procedures. These processes and policies allow IT to prevent most problems from occurring and provide recovery plans when a problem does occur.

The technical component covers the specific technical detail and infrastructure used to prevent, detect, and resolve a problem. Technical best practices include the following:

- QA and test systems to allow for thorough testing before deployment
- Redundant, secure system stack to prevent single point of failures and malicious acts from causing downtime
- A monitoring infrastructure to quickly detect, prevent, notify, and possibly resolve problems
- Automated recovery infrastructure to resolve the most common outages

Maximum Availability Architecture focuses on the technical best practices in configuring a resilient architecture, which will prevent most outages. For more information on the logistical best practices and other technical best practice components including prevention and detection of outages, please refer to the Operational Best Practices appendix of *Maximum Availability Architecture*.

Outages and Solutions

Continued availability is a fundamental requirement of many applications today. Businesses operate on a continual basis and in fact, many exist on the premise of 24x7 availability of their applications. Downtimes, intentional or otherwise, have a large opportunity cost in revenue and quality of service. MAA provides a recovery process and architectural framework to manage each outage and minimize the downtime associated with each outage. The Outages and Solutions section of *Maximum Availability Architecture* provides an outage decision tree for unscheduled and scheduled outages.

Unscheduled outages are unanticipated failures in any part of the technology infrastructure supporting the application, including hardware such as host machines, storage, switches, cables, cards, software (operating system, Oracle database and application server, application code), network infrastructure, naming services infrastructure, front-end load balancers and the current production site. The monitoring and HA infrastructure should provide for rapid detection and recovery from such failures.

Scheduled outages are planned outages. They are required for regular maintenance of the technology infrastructure supporting the application and include tasks such as hardware maintenance, repair and upgrades, software upgrades and patching, application changes and patching, and changes to improve performance and manageability of systems. Scheduled outages should be scheduled to occur at times best suited for continual application availability.

For each outage in the decision tree, the detailed recovery steps are described. The outage decision tree is divided as follows for both the production and standby sites:

- Unscheduled Outages on the Production Site
- Scheduled Outages on the Production Site
- Unscheduled Outages on the Standby Site
- Scheduled Outages on the Standby Site

The Unscheduled Outages on the Production Site table below is provided below. Some outages are handled automatically without any loss of availability. For example, instance failure is managed automatically by RAC. Other outages require multiple recovery steps. For example, when a site failover occurs, the outage decision matrix states that a Data Guard failover and a site failover must occur. MAA uses a variety of recovery options. These recovery options use a combination of Oracle product features and the infrastructure to prevent and minimize downtime and data loss. Multiple recovery options for each outage are described wherever relevant.

| Scope of Outage | Reason for Outage | Recovery Steps |
|-----------------|-------------------|----------------|
|-----------------|-------------------|----------------|

| Scope of Outage | Reason for Outage | Recovery Steps |
|-------------------------|----------------------|--|
| Site | Site failure | 1. Database failover 2. Site failover |
| Any application server | Node failure | Managed automatically by redundant nodes in the application server farm |
| All application servers | Complete failure | 1. Database switchover 2. Site failover |
| Database | Node failure | Managed automatically by RAC |
| Database | Instance failure | Managed automatically by RAC |
| Database | Cluster-wide failure | 1. Database failover 2. Site failover |
| Database | User error | 1. Database forced failover 2. Site failover OR Local object recovery |
| Any tier | Component failure | Managed automatically by redundant components |

Table 1: *Unscheduled Outages on the Production Site*

Restoring Fault Tolerance

Whenever a component within MAA fails, then the full protection, or fault tolerance, of MAA is compromised and possible single points of failure exist until the component is repaired. Restoring MAA to full fault tolerance to reinstate full MAA protection requires repairing the failed component. While full fault tolerance may be sacrificed during a scheduled outage, the method of repair is well understood because it is planned, the risk is controlled, and it ideally occurs at times best suited for continued application availability. However, for unscheduled outages, the risk of exposure to a single point of failure must be clearly understood.

The Restoring Fault Tolerance section of *Maximum Availability Architecture* focuses on describing the steps in restoring database fault tolerance and tie directly back to the solutions employed in dealing with the outages described above. The database tier fault tolerance restoration processes are detailed in the following sections:

- Restoring Failed Nodes or Instances within a RAC Cluster
- Restoring Standby Database after a Failover with Complete Recovery
- Restoring Standby Database after a Forced Failover
- Instantiating Initial Standby Database
- Restoring Fault tolerance after Dual Failures

CONCLUSION

The *Maximum Availability Architecture* is Oracle's top high availability solution that provides:

- Architectural components to protect your data
- Configuration best practices to achieve high availability
- Detailed descriptions of outages and solutions for quickly recovering from outages
- Restoring full database fault tolerance for continued protection

MAA embraces high availability so that any failure is handled transparently or with a thorough, automated recovery procedure that can achieve a low MTTR. After setting up MAA with the operational and configuration best practices, we recommend using the outage decision matrix and the detailed solutions to build a complete high availability solution for all potential outages. The complete deployment should be rehearsed to ensure that the required MTTR is met. After automation and some testing, you should be able to meet high availability requirements by leveraging the MAA practices such as:

- Fast site failover using a wide area traffic manager to reroute clients to the secondary site
- Transparent application tier failover with load balancers and mid-tier application server farms
- Host and instance failover with Transparent Application Failover and Real Application Clusters
- Database role reversal between primary and secondary sites for scheduled maintenance using Data Guard switchover
- Database failover to the standby database to protect from user errors, data errors and site disasters using Data Guard failover

MAA will be continually enhanced with knowledge from customer implementation experiences and with the results of further internal testing and validation. Future projects will include Oracle features such as logical standby databases, and features not yet released. MAA enhancements will be validated using different application infrastructures such as Oracle's Mail Server application, Oracle's Customer Relationship Management (CRM) application, and Oracle's Enterprise Resource Planning (ERP) application.

The Server Technologies High Availability Systems Group's charter is to build, test, validate and design high availability solutions. One of the goals of this group is to simplify customer HA deployments and ensure completeness. This requires integrated Oracle HA solutions that are easy to deploy and easy to manage and enable customers to meet their service levels. Enhancements will continue to make Oracle HA solutions more transparent and manageable.

The complete MAA description is presented in an Oracle white paper titled *Maximum Availability Architecture*, which is available on the Oracle Technology Network at http://otn.oracle.com/deploy/availability/pdf/MAA_WP.pdf.



Maximum Availability Architecture

October 2002

Author: High Availability Systems Group, Server Technologies

Contributing Authors: Andrew Babb, Cathy Baird, Pradeep Bhat, Ray Dutcher, Wei Hu, Susan Kornberg, Juan Loiaza, Ashish Prabhu, Lawrence To, Doug Utzig, Jim Viscusi, Shari Yamaguchi

Oracle Corporation

World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065

U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation

All rights reserved.