Maximum Availability Architecture: Overview

An Oracle White Paper
July 2002

ORACLE

Maximum Availability Architecture: Overview

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 simple, 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. It is being validated by the Oracle Server Technologies High Availability Systems Group and is being validated and deployed in numerous customer sites around the world.

This paper is targeted toward database administrators, system administrators, and architects with an understanding of Oracle Server, Real Application Clusters and Data Guard terminology. Please refer to *Oracle9i Database Concepts*, the Oracle Data Guard documentation set, and the Real Application Clusters documentation set.

INTRODUCTION

This paper provides an executive overview of MAA. The complete MAA description is provided in another Oracle white paper, *Maximum Availability Architecture*. *Maximum Availability Architecture* includes the following sections:

- Overview of MAA and its components
- Configuration best practices in building MAA
- Detailed descriptions of outages and solutions
- Restoring full database fault tolerance
- MAA within your data center

The overview provides an executive view of the architecture and its components. The configuration best practices section describes what needs to be implemented and why. The outage and solutions section justifies this architecture by providing the best solutions for a list of scheduled and unscheduled outages. After a failover operation or after resolving a database outage, use the section about restoring full database fault tolerance. It describes how to restore complete high availability to the database. Finally, in the section about MAA in the data center, MAA is described with some context to real world data centers and multiple production databases rather than in the simplified form of just one database or application.

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. Future revisions will contain configuration descriptions for

deploying the Oracle9*i* Application Server, Oracle's Customer Relationship Management (CRM) application, and Oracle's Enterprise Resource Planning (ERP) applications. It will also leverage the latest, tested high availability Oracle features.

This architecture was validated using Oracle 9*i* Release 2.

ARCHITECTURE OVERVIEW

MAA provides a simple, 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:

- Redundant middle or application tier
- Redundant network infrastructure
- Redundant storage infrastructure
- Real Application Clusters (RAC) to protect from host and instance failures
- Oracle Data Guard (DG) to protect from human errors and data failures and recover from site failures
- Sound operational practices

Figure 1 provides an overview of the architecture.

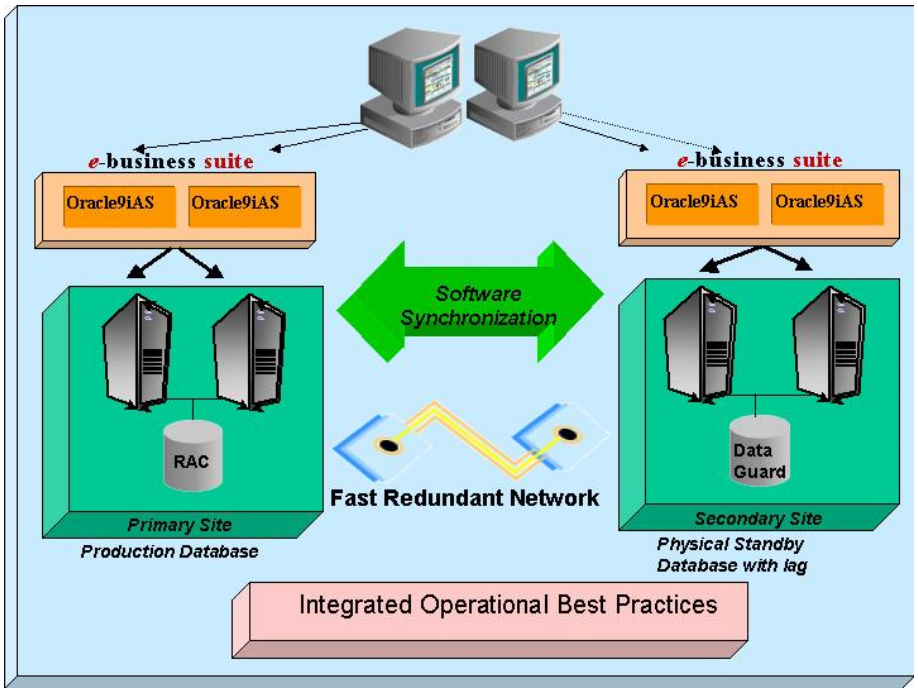

Figure 1: Maximum Availability Architecture Overview

Figure 1 illustrates identically configured sites. Each site consists of redundant components and redundant routing mechanisms, so that requests are always serviceable even in the event of a failure. Most outages are resolved locally. Client requests are always routed to the site playing the production role. After a failover or switchover operation occurs due to a serious outage, client requests are routed to another 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 a 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 switchover and failover functions allow the roles to be traded between sites. We have defined two 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.

RAC and 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, the Data Guard Broker GUI, and a command-line interface. In MAA, Data Guard provides automatic remote archiving, managed recovery, and role management between the production database and the physical standby databases. A 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. In Figure 1, the standby database is configured with a lag to prevent the application of user error or corruption in the production database to the standby database.

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. We advocate identical site configurations to ensure that performance is not sacrificed. In addition, this allows processes and procedures to be kept the same between sites, making operational tasks much easier to maintain and execute. Furthermore, 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 as illustrated in the above diagram.

The following sections give a brief overview of each component of MAA:

Application Tier

The application tier consists of at least one set of computers that host application services such as CRM or Self Service Applications. The functionality for these services can be distributed across multiple machines. For example, the application tier functionality can be implemented with a machine hosting the web server in addition to a server hosting the application specific processes such as one for servlets, another for Enterprise Java Beans (EJBs), and another for the infrastructure database. Together, this set of servers provides the application service to the clients and is the building block of the application tier. Redundant sets of hosts for each service exist in the application tier to provide resiliency as well as scalable

load balancing. They are collectively referred to as a server farm. The server farm usually has either a hardware or software-based load balancer that distributes incoming client requests across multiple hosts in the server farm.

MAA has a separate application tier server farm at each site. The server farm at the primary site is fully replicated in terms of hardware, OS, application server software, and application-specific software at the secondary site. They are configured similarly in all respects. Each server farm has a hardware or software-based load balancer facing the clients. 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).

In Figure 1, if the primary site fails, then the server farm at the secondary site needs 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.

Because each server farm is redundant on multiple sets of machines, problems and outages within a site of individual application tier hosts are transparent to the client. Automatic detection by the monitoring infrastructure and, where applicable, restart of a failed component of the application tier ensure near uninterrupted availability of a application services.

Network Infrastructure

All network components (router, firewalls, load balancers) should be 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 Figure 2: Network Failover Routes, you can see that each network device is part of a redundant pair, and multiple failover routes are available in case of any single point outage.

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 serious outage. In the event of a primary site failure, a wide-area traffic manager is used to direct traffic to the secondary site. [See Figure 2– Tier 1.]

Load balancers disguise the application server farm 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. [See Figure 2– Tier 2.]

The application servers use configuration information for Oracle Net Services to connect to the database. This information can be stored in a centralized 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.

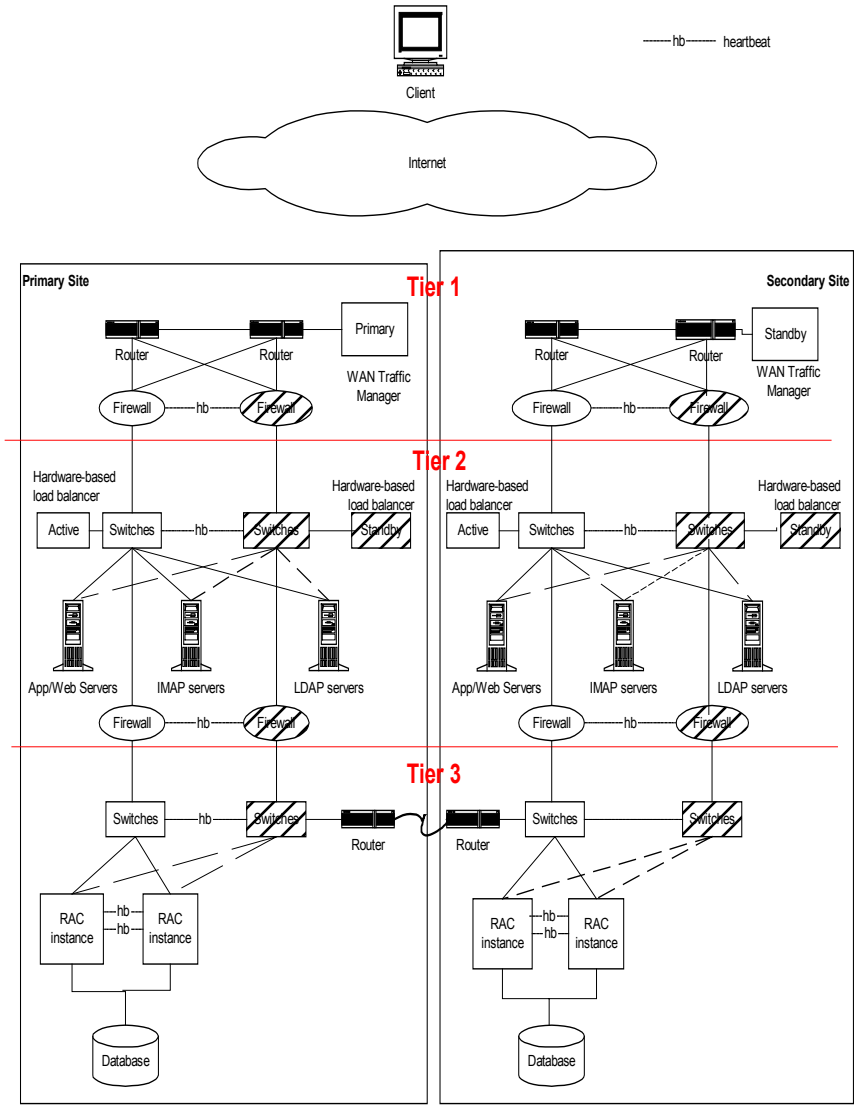

Figure 2: Maximum Availability Network Configuration

Storage Infrastructure

Several storage options exist today and have a variety of features and options. From a high availability perspective, the following features are required:

- Full redundancy for all hardware components
- Online parts replacement (hot swappable parts)
- Online patch application
- Hardware mirroring and striping capabilities

- Mirrored write cache with battery backup
- Load balancing and failover capabilities across all host bus adaptors

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.

Data Guard and the Secondary Site

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.

Oracle Data Guard manages the two databases by providing remote archiving, managed recovery, switchover and failover features. In MAA, the production database resides within a Real Application Cluster 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 and failover operations. 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. The secondary site protects from site failures and includes the primary site equivalents:

- Redundant middle or application tier
- Redundant network infrastructure
- Redundant storage infrastructure
- Identical Real Application Clusters environment to protect from host and instance failures when in the production role
- Sound operational practices
- Oracle Data Guard to protect from human errors and data failures when in the standby role

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 remote archiving 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 should be 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. 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.

Operational Best Practices

An architecture that contains all the necessary hardware and software features without sound operational practices will ultimately fail to meet availability service levels. Operational best practices provide the greatest impact on availability by:

- Preventing outages
- Detecting potential problems
- Recovering from outages within a tolerated MTTR

Operational best practices have been 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*.

Maximum Availability Architecture
July 2002
Author: Lawrence To, High Availability Systems Group
Contributing Authors: Andrew Babb, Cathy Baird, Pradeep Bhat, Ray Dutcher, Wei Hu, Susan Kornberg, Juan Loiaza, Ashish Prabhu, Doug Utzig, 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