Oracle Application Server 10g Release 2 Disaster Recovery

An Oracle White Paper
November 2004

ORACLE

Oracle Application Server 10*g* Release 2 Disaster Recovery

Oracle Application Server 10*g* Release 2 Disaster Recovery

INTRODUCTION

Protecting enterprise application services against disasters has become a business reality today more than ever. An application service usually consists of databases, application servers, and applications. A highly available disaster recovery solution must be a solution for all these system components.

Traditional application disaster recovery solutions rely on third-party remote mirroring products to synchronize independent application servers and databases to a standby site. This negatively affects cost, integration, and operational complexity.

Oracle Application Server Guard in 10*g* Release 2 is the leading disaster recovery solution integrated into an Application Platform Suite that promises lower cost, single-vendor integration, and uncompromising ease of use.

In this paper, we discuss how Oracle Application Server Guard can help meet your disaster recovery needs.

DISASTERS ARE REALITY

Business continuity is a key to many e-business operations. Down time of mission-critical applications translates directly to reduction in productivity, service quality and lost revenue.

Mission-critical application services require both a local high availability solution and a disaster recovery solution. A local high availability solution provides redundancy in one data center. However, failures that impact the entire application services are more and more common as these applications become more complex and interdependent. An effective disaster that disables an application service is not necessarily one that destroys the whole data center (e.g. flood, fire), but is more likely one that disables one particular type of resources. For example, a failure of corporate gateways or ISP network connections, a spread of viruses to all HTTP listener nodes, a misconfiguration, a power outage, and an incorrect patch could all lead to days of complete loss of services.

A disaster recovery solution upholds your application services beyond all types of failures to the primary data center by providing redundancy in a separate, remote

data center and a strategy to synchronize and fail over between these two data centers.

An added benefit of a disaster recovery solution is continued application services during hardware and operating system upgrade and patching. The same disaster recovery strategy can often easily switch over application services between the two data centers.

DISASTER RECOVERY FOR THE ENTIRE APPLICATION ECOSYSTEM

A disaster recovery solution is uniquely distinct from a local high availability solution. While a local high availability solution protects against partial failures of an application system (process failures, system failures, media failures), a disaster recovery solution must protect against a complete data center failure as well as these partial failures.

Just as a local high availability solution provides resource redundancy locally, a disaster recovery solution entails resource redundancy geographically. Hardware, software, network, IT management, and any other resources that constitute the application ecosystem must be redundant.

Oracle Application Server Guard is a disaster recovery solution that is architected with all these resource redundancy in mind. Leveraging Oracle Data Guard, the industry-leading disaster recovery solution for databases, Oracle Application Server Guard is the only comprehensive disaster recovery solution for applications among major application server products.

HOW TO CHOOSE A DISASTER RECOVERY SOLUTION

Choosing the right disaster recovery solution for a given enterprise is often a balance between functionality, cost, and ease-of-use. To make the right choice, we must first understand these three factors and their correlation.

The core functionality of any disaster recovery solution is that should disasters happen, your application services can resume with acceptable data loss and service delay. IT managers measure this functionality by two recovery objectives: Recovery Point Objective and Recovery Time Objective.

Recovery Point Objective (RPO) measures the amount of lost data. Technically, RPO is the distance between the disaster time and the time that data can be reliably recovered to. A smaller RPO leads to less data loss. A zero RPO guarantees zero data loss.

Recovery Time Objective (RTO) measures the delay in services. RTO is the distance between the disaster time and the time that services resume after recovery.

In a real business situation, it is crucial to evaluate the cost and the ease-of-use of a disaster recovery solution in addition to its functionality. A solution with the lowest RPO and RTO is not necessarily the best solution for your business requirements,

as lower RPO and RTO typically come with higher cost and more complicated management.

The cost of a disaster recovery solution generally is from redundant software, redundant hardware, redundant network infrastructure, and additional IT management.

Managing a disaster recovery solution often involves managing products from multiple vendors with more management complexity and steeper learning curve. For example, a disaster recovery solution based on the remote mirroring of storage servers requires special storage hardware management in addition to the application software and hardware management.

The most suitable disaster recovery solution for your business is the one that meets your RTO and RPO requirements with the lowest cost and the best usability.

For example, a stock exchange cannot afford any transaction loss or any downtime during trade hours, hence zero RTO and zero RPO are a must. An online bank cannot tolerate any data loss either, but RTO can be higher as users tolerate some web site downtime. A global Portal service, on the contrary, can lose some recent end user customization but cannot lose service for long. Finally, a reporting system may tolerate both high RPO and high RTO because reports can be regenerated later and are not urgent.

COMMON DISASTER RECOVERY SOLUTIONS FOR APPLICATIONS

A disaster recovery solution for applications must support redundancy of software, hardware, network and other resources. It usually has an architecture as in figure 1.

Regardless of its type of solutions, a disaster recovery solution needs a production site that hosts the production application hardware, software and network, a standby site that hosts the standby application hardware, software and network, and a synchronization network between the two sites.

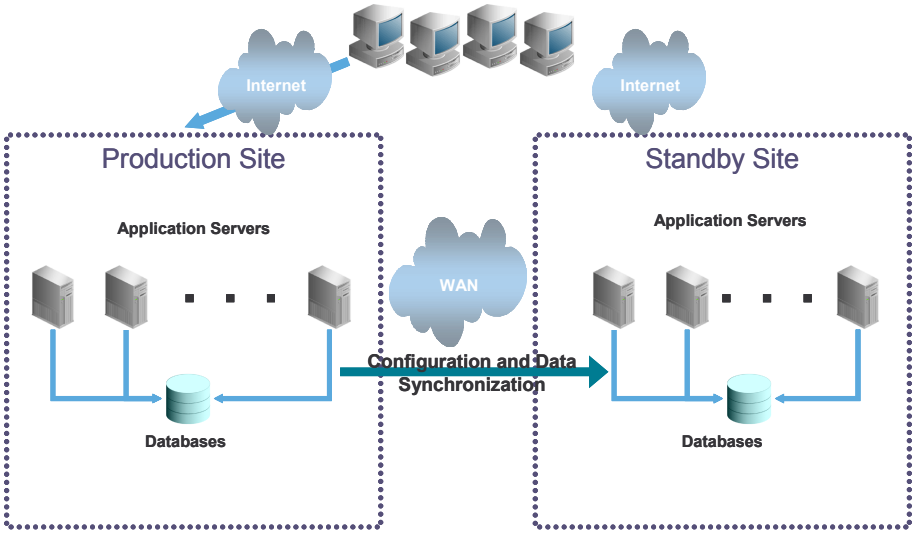

Figure 1: A Common Disaster Recovery Architecture

First the two sites need to have the hardware, software and network properly installed and configured. Second, the standby site needs to be instantiated as the standby to the production site. This instantiation also transports all data and configuration from the production site databases and applications to the standby site so that they are synchronized. Third, after the initial instantiation, as the production site evolves with additional configuration and data changes, you need to synchronize these changes to the standby site constantly. The instantiation and recurring synchronization rely on the synchronization network between the two sites. Finally, during an unplanned downtime, the primary site's entire application services fail over to the standby site, reversing the role of the two sites. Similarly, the two sites can also be switched over for a planned downtime.

IT managers often find two types of disaster recovery solutions:

- Third-Party Remote Mirroring solutions
- Integrated Application Server Disaster Recovery solutions

The major differences are in what manages the instantiation, synchronization, failover and switchover. A third-party remote mirroring solution relies on third-party remote mirroring solutions to perform these tasks. Sometimes the database systems perform the database disaster recovery, and the third-party remote mirroring solution provides disaster recovery for applications installed on regular file systems. Otherwise, the remote mirroring solution handles the disaster recovery process for all the application software and databases.

Host-based remote mirroring transports data from the production site to the standby site with non-application-aware replication software. Storage-based remote mirroring transports data with non-host-aware storage servers.

An integrated application server disaster recovery solution has instantiation, synchronization, switchover, and failover operations built into the application server and database.

Third-party remote mirroring is usually a quick way to provide a disaster recovery solution for application servers. The third-party mirroring products tend to be relatively application agnostic. However, it is a common misconception that a simple combination of third-party remote mirroring and any application makes a good disaster recovery solution. Application servers that are not designed with disaster recovery in mind may have special dependencies on the production site hardware and network environment such as IP addresses that cannot be duplicated in the standby site. Since the remote mirroring products are not aware of these application-specific dependencies, even though the software and data are replicated bit-by-bit to the standby site, the standby application system may not function correctly.

Cost is another important issue with remote mirroring. Proprietary and expensive storage systems, synchronization network and replication software add to licensing cost. Maintaining products from multiple vendors add to the maintenance and

support cost. The total ROI can be significantly lower than an integrated application server disaster recovery solution that only require commodity server, storage, and network for the standby site and synchronization.

More detailed comparison of the two types of solutions is illustrated in the next table.

	Third Party Remote Mirroring	Integrated Application Server Disaster Recovery
Flexibility	Storage-based requires identical proprietary storage systems at both sites. Host-based requires proprietary replication software at both sites.	Can use commodity servers, operating systems, storage, and synchronization network (e.g. TCP/IP). No proprietary third-party replication software or storage systems.
Cost	Extra cost in expensive replication software or Enterprise Storage Systems. Extra capacity licensing based on replication volume.	No extra hardware or software cost for replication.
Integration	Multi-vendor solution. Applications may not recover even with remote mirroring (need better explanation why this may happen).	Single-vendor solution. Fully tested and supported.
Synchronization Network Outage	Replication buffering allows certain period of network outage between the two sites.	Application-aware synchronization can resume after any length of synchronization network outage.
Ease-of-Use	Managing products from multiple vendors. Need special skills in applications, replication software, and storage replication.	Single-vendor management framework. Fewer products to learn.

Third-Party Remote Mirroring and Integrated Application Server Disaster Recovery

ORACLE APPLICATION SERVER GUARD

What Is Oracle Application Server Guard?

Oracle Application Server Guard is the only integrated application server disaster recovery solution among major application server products. Leveraging Oracle Data Guard, the industry-leading disaster recovery solution for databases, Oracle Application Server Guard is an integrated, low-cost, and easy-to-use disaster recovery solution for not just databases, but entire application services.

The Value Proposition

There are three key advantages of Oracle Application Server Guard compared to other application server disaster recovery solutions based on third-party remote mirroring:

- Integrated
- Low cost
- Easy to use

Oracle Application Server Guard is integrated into the Application Platform Suite offered by Oracle Application Server. It provides tools and strategies customized to all components and applications in the suite so that all of them would function correctly after a site failover or switchover.

Oracle Application Server Guard is significantly cheaper than other application servers' mix-and-match approach. Unless you need hardware clusters for a local high availability solution, OracleAS Guard works on commodity hardware and network.

Oracle Application Server Guard is easy to use. Standby instantiation, synchronization, switchover and failover can each be performed with a few commands. No additional software and hardware need to be installed and configured.

Oracle Application Server Guard Architecture

Oracle Application Server Guard requires a standby site that is homogeneous to the production site. The production site Oracle Application Server deployment usually has three major components: Application Middle Tier (Portal, Wireless, OC4J, etc.), Identity Management services (Single-Sign-On services, LDAP services, etc.), and Metadata Repository services.

OracleAS Guard supports a production site deployment with one or more active Application Middle Tier installs and a joined Identity Management services and Metadata Repository services install that is either a single instance install or a cold failover cluster install for local high availability. The standby site needs to have the

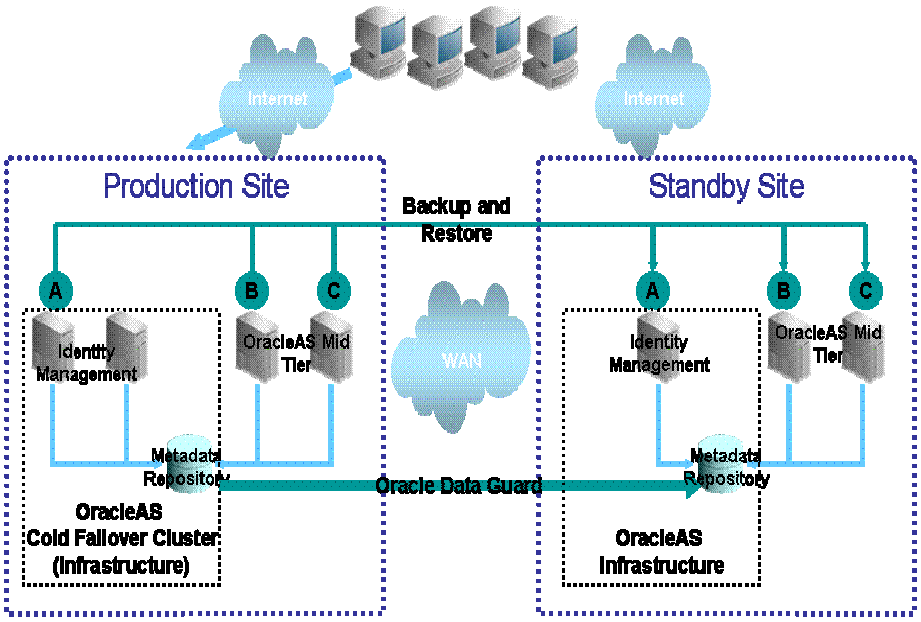

Figure 2: An Oracle Application Server Guard Architecture Example

same number of Application Middle Tier installs and a joined Identity Management and Metadata Repository install (an OracleAS Infrastructure install type) that can be either standalone or cold failover cluster regardless of what the production site has.

The standby site Oracle Application Server installs are normally passive, synchronized and ready to take over the application services. They can co-exist with other Oracle products installs, as long as they do not conflict with the standby OracleAS system during standby installation, instantiation, synchronization, failover and switchover. For example, you can utilize a test system to install the standby software and use the system for testing and standby synchronization at the same time. During site failover/switchover, you should shutdown the test applications before starting the standby Oracle Application Server instances.

Figure 2 shows an Oracle Application Server Guard architecture example. In this example, the production site uses two Middle Tier instances and one CFC Infrastructure instance, while the standby site has two Middle Tier instances and one Infrastructure instance.

Each Oracle Application Server 10*g* Release 2 install includes an Oracle Application Server Guard server (OracleAS Guard server) and a client (OracleAS Guard control, asgctl). The servers collaborate in parallel to perform all the disaster

recovery procedures. The user invokes the client to connect to any server in the production site to initial these procedures with simple OracleAS Guard commands.

To keep the standby site in sync with the production site, OracleAS Guard needs to synchronize data in the Metadata Repository and configuration files in the Application Middle Tier and the Identity Management services.

On the Metadata Repository side, with a site instantiation command from the client, OracleAS Guard servers interact with Oracle Data Guard to instantiate and synchronize the standby Metadata Repository as a physical standby database.

On the Identity Management and Application Middle Tier side, OracleAS Guard uses the Oracle Application Server Backup and Restore tool to assemble and transport configuration changes to the standby site and apply these changes.

The key in Oracle Application Server Guard's integration within the suite is that each synchronization step in Metadata Repository, Identity Management services and Application Middle Tier are performed at the same time across the three components. Site instantiation is also applied across the three components simultaneously. This is crucial for these components to always be in a consistent state, ready to be activated.

Setting Up Oracle Application Server Guard

Using Oracle Application Server Guard has a life cycle of several major steps.

The first step is to prepare the primary site and the standby site. You may set up a production site and a standby site from the scratch, or set up a standby site for an existing production site running the same version of Oracle Application Server.

The two sites need to have the same number of machines allocated for Application Middle Tier with the same physical hostname. Each site needs to have one Oracle Application Server Infrastructure machine or cold failover cluster with the same virtual hostname. Each pair of peer machines need to have homogeneous environments, such as operating system, hardware platform, etc.

You can choose to use the same standby site machines for running other Oracle Application Server instances, such as a test system or a non-critical application. Then the standby site will essentially be a passive, emergency backup site that normally hosts other applications. In order to provide the same level of capacity, you should plan to shutdown the non-critical applications when the standby site is activated.

The second step is to install Oracle Application Server. If your production site is already running, you will only need to install the same set of Oracle Application Server instances on their standby peer machines. It is important that the peer instances use the same configuration, such as ports and Oracle Home path names.

The third step is to instantiate the standby site. You use asgctl to connect to an OracleAS Guard server which in turn interacts with other servers in two sites. In

parallel, these servers set up the standby Metadata Repository database as a physical standby database using Oracle Data Guard, take a full backup of Application Middle Tier and Identity Management configuration, and apply this backup to the standby peer instances. Essentially, the site instantiation links the two sites and does a full synchronization. The following is a command line example of site instantiation:

1. Invoke the OracleAS Guard client, asgctl, and connect to the OracleAS Guard server on the production Infrastructure node prodinfra.

```
> asgctl.sh

ASGCTL > connect asg prodinfra ias_admin/<adminpwd>

Successfully connected to prodinfra:7890
```

2. Specify the primary database, iasdb.

```
ASGCTL> set primary database sys/testpwd@iasdb
```

3. Verify the standby farm is valid for OracleAS Guard through the standby Infrastructure instance at host standbyinfra.

```
ASGCTL> verify farm with standbyinfra
```

4. Instantiate the farm at the secondary site.

```
ASGCTL> instantiate farm to standbyinfra
```

5. Disconnect from the OracleAS Guard server and exit the OracleAS Guard client. The standby site is established and synchronized as of now.

```
ASGCTL> disconnect

ASGCTL> exit

>
```

The fourth step is a recurring step to periodically synchronize the standby site. Synchronization can happen while the production site is running. It can be done at any time, but the less frequently you synchronize, the farther the standby site lags. The following is an example command line sequence. (To save space, identical commands as in the instantiation procedure is not shown.)

1. Invoke the OracleAS Guard client and connect to the OracleAS Guard server.
2. Specify the primary database.
3. Verify the farm.
4. Synchronize the secondary site with the primary site.

```
ASGCTL> sync farm to standbyinfra
```

5. Disconnect from the OracleAS Guard server and exit the OracleAS Guard client.

Since synchronization is a recurring step, you can put these synchronization commands in an asgctl script file and invoke asgctl periodically from a third-party scheduler such as cron in Unix systems.

Oracle Application Server Guard in Planned and Unplanned Downtime

With its seamless integration within Oracle Application Server 10*g*R2 and Oracle Data Guard, OracleAS Guard provides powerful application services recovery, yet at the same it is very simple to operate.

You can use OracleAS Guard for any unplanned downtime, or planned downtime to upgrade hardware or operating systems.

In order to reduce the RTO, your disaster recovery plan should include a plan for WAN DNS switchover. When the standby site is activated, clients must use a new hostname-to-IP mapping to send requests to the new production site. Due to DNS caching, this mapping change will not take effect for your clients immediately. Thus your RTO will be at least as long as one DNS caching TTL for your clients to recognize the new production site. An alternative is to use a third-party global load balancer solution that may recognize DNS changes much faster.

In order to reduce RPO, you should synchronize the two sites more frequently relative to the rate of configuration and data changes in the production site.

In an unplanned downtime, perform site failover.

1. Perform a wide area DNS switchover to direct requests to the standby site, so that DNS changes can start to propagate to your clients as you fail over.
2. Connect to an OracleAS Guard server.
3. Specify that the primary database on the standby site is now identified as the new primary database on this new production site.

```
ASGCTL> set new primary database sys/testpwd@iasdb
```

4. Perform an asgctl failover operation.

```
ASGCTL> failover
```

5. Disconnect and exit.

In a planned downtime, you can perform a site switchover.

1. Reduce the wide area DNS TTL value for the site, so that new DNS lookup values will be cached for a shorter period time until we change the DNS mapping to the standby site.
2. Use asgctl to connect to the Oracle Application Server Guard server on the production Oracle Application Server Infrastructure instance.

3. Specify the primary database.

4. Synchronize the secondary site with the primary site.

```
ASGCTL> sync farm to standbyinfra
```

5. Switchover the farm to the secondary site.

```
ASGCTL> switchover to standbyinfra
```

6. Disconnect and exit.

7. Perform a wide area DNS switchover to direct requests to the new production site.

8. Adjust the wide area DNS TTL to an appropriate value.

CONCLUSION

As the only disaster recovery solution integrated in an Application Platform Suite, Oracle Application Server Guard 10gR2 presents a highly available, low cost, and administrator-friendly alternative to multi-vendor application server disaster recovery solutions. It further fortifies Oracle Application Server's leadership in application platform high availability.

Oracle Application Server 10g Release 2 Disaster Recovery
December 2004
Author: Xiang Liu
Contributing Authors:

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