NETAPP TECHNICAL REPORT

Oracle Fusion Middleware DR Solution Using NetApp Storage

Padmanabhan Sadagopan, NetApp

Michael Doherty, NetApp

Shailesh Dwivedi, Oracle

May 2008 | TR-3672

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This technical report covers replicating Oracle® Fusion Middleware 10*g* using NetApp® SnapMirror®, Snapshot™, and FlexVol® technology. It describes the configuration and steps to create a disaster recovery (DR) solution for Oracle Fusion Middleware 10*g* in a simple, fast, accurate, and cost-effective method.

As companies have become increasingly dependent on data and access to that data, DR has become a major concern. The ability to provide a highly available, 24x7 operation is very desirable. Now more than ever, protecting mission-critical business applications has become a necessity.

NetApp DR solutions are simple to deploy and recover, reducing downtime. They are flexible enough to address a broad range of recovery point objectives ranging from zero data loss to one hour to one day, enabling customers to make the tradeoff between cost and data loss exposure and to replicate over long distances, providing protection from both site and regional disasters.

NetApp DR solutions enable customers to affordably protect more of their business-critical applications.

BACKGROUND

Oracle has started supporting disk replication as a disaster protection solution for Fusion Middleware. NetApp storage systems have proven replication technologies to support such a solution. This paper discusses how NetApp storage's replication features can be used to implement a disaster protection solution for Oracle Fusion Middleware.

DISASTER RECOVERY

To protect against disaster situations such as flood, fire, earthquake, and human acts such as terrorism, a computing environment needs to have a backup arrangement on a secondary site located a significant distance away. The secondary site, also called a standby site, is known as the disaster recovery site. In the event of disaster, the computing infrastructure at the DR site is used to serve the business needs.

Additionally, disaster recovery can refer to how a system is managed for planned outages. For most disaster recovery situations, the solution involves replicating an entire site, not just pieces of hardware or subcomponents. This also applies to the Oracle Fusion Middleware disaster recovery solution.

In the most common configuration, a standby site is created to mirror the production site. Under normal operation, the production site actively services client requests. The standby site is maintained to mirror the applications and content hosted by the production site.

SnapMirror software from NetApp provides a fast, flexible enterprise solution for mirroring or replicating data over local or wide area networks. SnapMirror can be used for:

- Disaster recovery
- Data replication for local read-only access at a remote site
- Application testing on a dedicated read-only mirror
- Data migration between NetApp storage systems

1 ORACLE FUSION MIDDLEWARE

Oracle Fusion Middleware is a preintegrated portfolio of customer-proven software that spans from portals and process managers to application infrastructure, developer tools, and business intelligence. As the industry's most comprehensive family of middleware, it enables your organization to increase its capacity for growth and change; improve insight into business operations; mitigate risk and drive compliance; and connect with customers, partners, and workers.

Providing maximum availability architecture is one of the key requirements for any Oracle Fusion Middleware enterprise deployment. Oracle Fusion Middleware includes an extensive set of high-availability features such as process death detection and restart, server clustering, load balancing, failover, backup and recovery, rolling upgrades, rolling configuration changes, and dynamic discovery, which protect an enterprise deployment from unplanned downtime and minimize planned downtime.

The product binaries and configuration for Oracle Fusion Middleware components and applications get deployed in Oracle Home directories on the middle tier. Additionally, most of the products also have metadata or runtime state data stored in a database repository. Therefore, the Oracle Application Server disaster recovery solution keeps middle tier file system data as well as middle tier data stored in databases at the production site synchronized with the standby site.

The Oracle Application Server disaster recovery solution deployed on NetApp storage supports these methods of providing data protection for Oracle Fusion Middleware data and database content:

- Oracle Fusion Middleware product binaries, configuration, and metadata files: Use NetApp SnapMirror disk replication technology.
- Database content: Use Oracle Data Guard for Oracle Databases.

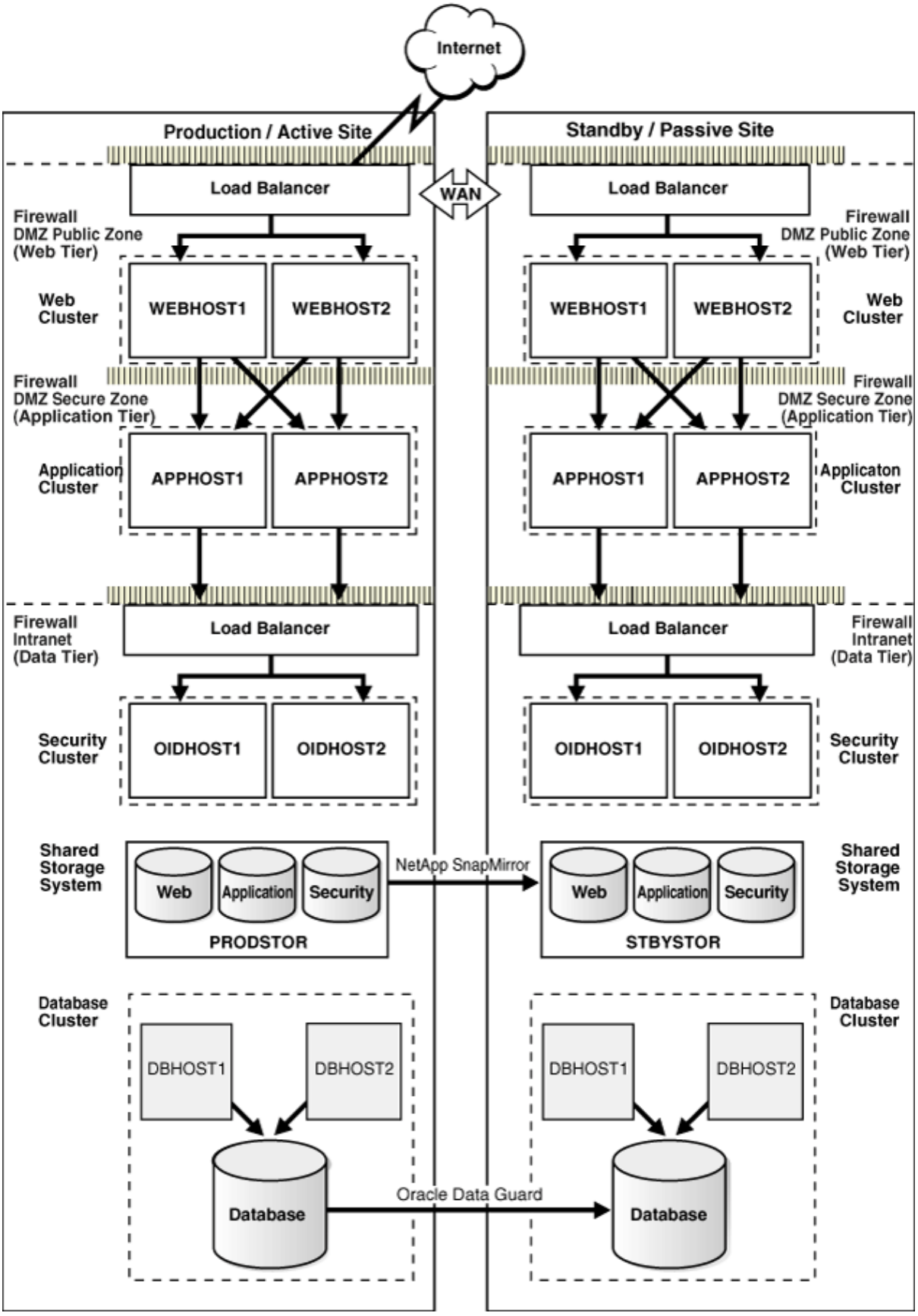
Internet
Production / Active Site
Standby / Passive Site
Load Balancer
WAN
Load Balancer
Firewall
DMZ Public Zone
(Web Tier)
Firewall
DMZ Public Zone
(Web Tier)
Web
Cluster
WEBHOST1
WEBHOST2
WEBHOST1
WEBHOST2
Web
Cluster
Firewall
DMZ Secure Zone
(Application Tier)
Firewall
DMZ Secure Zone
(Application Tier)
Application
Cluster
APPHOST1
APPHOST2
APPHOST1
APPHOST2
Applicaton
Cluster
Firewall
Intranet
(Data Tier)
Load Balancer
Load Balancer
Firewall
Intranet
(Data Tier)
Security
Cluster
OIDHOST1
OIDHOST2
OIDHOST1
OIDHOST2
Security
Cluster
Shared
Storage
System
Web
Application
Security
PRODSTOR
NetApp SnapMirror
Web
Application
Security
STBYSTOR
Shared
Storage
System
Database
Cluster
DBHOST1
DBHOST2
DBHOST1
DBHOST2
Database
Cluster
Database
Oracle Data Guard
Database

Figure 1) Oracle Application Server DR topology.

Key aspects of the solution in Figure 1 are:

- The solution has two sites. The current production site is running and active, while the second site is serving as a standby site and is in passive mode.
- Hosts on each site have mountpoints defined for accessing the shared storage system for the site.
- On both sites, the Oracle Application Server components are deployed on the site's shared storage system. This involves creating all the Oracle home directories, which include product binaries and configuration data for middleware components, in volumes on the production site's shared storage and then installing the components into the Oracle home directories on the shared storage. In Figure 1, a separate volume is created in the shared storage for each Oracle Application Server host cluster (note the Web, Application, and Security volumes created for the Web cluster, application cluster, and security cluster in each site's shared storage system).
- Mountpoints need to be created on the shared storage for the production site. The Oracle Application Server software for the production site will be installed into Oracle home directories using the mountpoint on the production site shared storage. Symbolic links also need to be set up on the production site hosts to the Oracle Application Server home directories on the shared storage at the production site.
- Mountpoints need to be created on the shared storage for the standby site. Symbolic links also need to be set up on the standby site hosts to the Oracle Application Server home directories on the shared storage at the standby site. The mountpoint and symbolic links for the standby site hosts must be identical to those set up for the equivalent production site hosts.
- Disk replication technology: NetApp SnapMirror is used to copy the middle-tier file systems and other data from the production site's shared storage to the standby site's shared storage.
- After disk replication is enabled, application deployment, configuration, metadata, data, and product binary information is replicated from the production site to the standby site.
- It is not necessary to perform any Oracle software installations at the standby site hosts. When the production site storage is replicated at the standby site storage, the equivalent Oracle home directories and data are written to the standby site storage.
- Replication is scheduled at a specified interval. The recommended interval is once a day for the production deployment, where the middle-tier configuration does not change very often. Additionally, you should force a manual synchronization whenever you make a change at the production site (for example, if you deploy a new application at the production site).
- Before forcing a manual synchronization, you should create a Snapshot copy of the site to capture its current state. This helps to ensure that the Snapshot copy gets replicated to the standby site storage and can be used to roll back the standby site to a previous synchronization state, if desired. Recovery to the point of the previously successful replication (for which a Snapshot copy was created) is possible when a replication fails.
- Oracle Data Guard is used to replicate all Oracle Database repositories, including Oracle Application Server repositories and custom application databases.
- User requests are initially routed to the production site.
- When there is a failure or planned outage of the production site, you perform the following steps to enable the standby site to assume the production role in the topology:

1. Stop the replication from the production site to the standby site (when a failure occurs, replication might have already been stopped as a result of the failure).
2. Perform a failover or switchover of the Oracle Databases using Oracle Data Guard.
3. Start the services and applications on the standby site.
4. Use a global load balancer to reroute user requests to the standby site. At this point, the standby site has assumed the production role.

2 NETAPP SNAPMIRROR TECHNOLOGY

NetApp SnapMirror technology provides either synchronous or asynchronous mirroring of data between storage system volumes. Data on the source volume is periodically replicated to the target at a user-definable time interval, with the range being from one minute to one month. At the end of each replication event, the mirror target volume becomes an exact block-for-block copy of the mirror source volume. At that point, the two volumes share identical data content and characteristics. The mirror is initialized by effectively copying the entire source volume to the target volume. Once this initial copy is complete, replication events thereafter copy only changed blocks from the source volume to the target volume. This provides a highly efficient data replication mechanism.

SnapMirror technology is a key component of enterprise data protection strategies. If a disaster occurs at a source site, businesses can access mission-critical data from a mirror on another NetApp storage system, making sure of uninterrupted operation.

The destination NetApp storage system can be located at virtually any distance from the source. It can be in the same building or on the other side of the world, as long as the interconnecting network has the necessary bandwidth to carry the replication traffic that is generated.

SnapMirror technology leverages the WAFL® (Write Anywhere File Layout) Snapshot capability to create and update a copy of a source volume or qtree on a destination NetApp storage system. The mirror copy is accessible to users in read-only mode on the destination NetApp storage system. NetApp FlexClone® technology allows you to create a writable copy, which can be used for any testing without impacting the original replica and replication process.

SnapMirror software makes a baseline transfer of the data (comparable to a full backup for tape backups). The initial transfer can be accomplished either through a network connection or through the restore of a tape on the destination. SnapMirror then updates the mirror by replicating only new or changed data blocks. Mirror copies are consistent because SnapMirror software operates on consistent Snapshot copies.

System administrators specify the intervals, based on RPO requirements, at which SnapMirror Snapshot copies are created and the times at which incremental transfers will occur. Determining this schedule depends upon how much the data changes during the day, how up-to-date the mirror needs to be, CPU usage on the source storage system, and available network bandwidth.

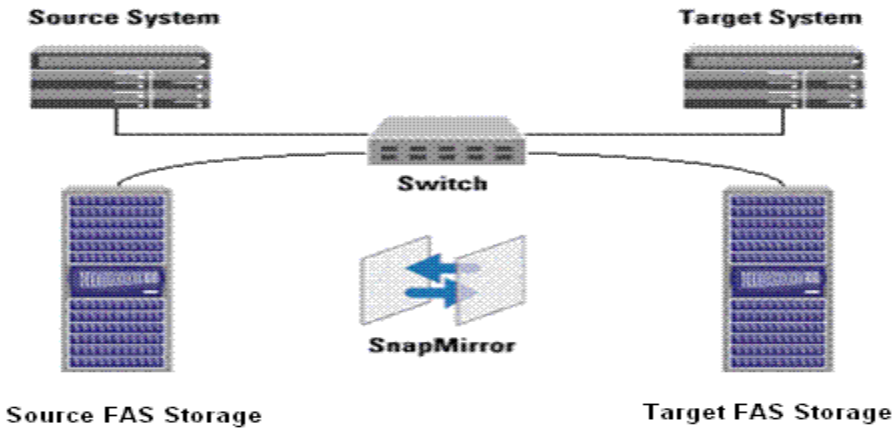

Figure 2) Server and storage: fabric-attached storage (FAS) configuration.

The source server stores all the Oracle Fusion Middleware files in the NetApp FAS2020 storage system, and the target server does the same, but with a separate NetApp FAS2020.

3 TERMINOLOGY

3.1 NetApp Storage Architecture

NetApp storage architecture refers to how Data ONTAP® software provides data storage resources to host or client systems and applications. Data ONTAP distinguishes between the physical layer of data storage resources and the logical layer, which includes the file systems and the data that reside on the physical resources.
The physical layer includes disks, redundant array of independent disks (RAID) groups to which they are assigned, plexes, and aggregates.
The logical layer includes volumes, qtrees, logical unit numbers (LUNs), and the files and directories that are stored in them.

A plex is a collection of one or more RAID groups that together provide the storage for one or more WAFL file system volumes.

An *aggregate* is a collection of one or two plexes, depending on whether you want to take advantage of RAID-level mirroring.

When you create an aggregate, Data ONTAP assigns data disks and parity disks to RAID groups, depending on the options you choose, such as the size of the RAID group (based on the number of disks to be assigned to it) or the level of RAID protection.
You use aggregates to manage plexes and RAID groups because these entities only exist as part of an aggregate. You can increase the usable space in an aggregate by adding disks to existing RAID groups or by adding new RAID groups. Once you've added disks to an aggregate, you cannot remove them to reduce storage space without first destroying the aggregate.

For details, refer to the NetApp Data ONTAP Storage Management Guide at http://now.netapp.com/NOW/knowledge/docs/ontap/rel724/pdfs/ontap/mgmtsag.pdf.

3.2 Volumes

Volumes are file systems that hold user data that is accessible via one or more of the access protocols supported by Data ONTAP, including NFS, CIFS, HTTP, WebDAV, FTP, FCP, and iSCSI.

Each volume depends on its containing aggregate for all its physical storage, that is, for all storage in the aggregate's disks and RAID groups. Aggregates serve as containers for both traditional and flexible volumes, but the relationship of the aggregate to each kind of volume is very different.

A traditional volume is a volume that is contained by a single, dedicated aggregate; it is tightly coupled with its containing aggregate. The only way to grow a traditional volume is to add entire disks to its containing aggregate.

3.3 FlexVol

Flexible volumes allow you to manage the logical layer of the file system independently of the physical layer of storage. Multiple flexible volumes can exist within a single separate, physically defined aggregate structure of disks and RAID groups. Flexible volumes contained by the same aggregate share the physical storage resources, RAID configuration, and plex structure of that aggregate. FlexVol volumes can have their own Snapshot schedule or their own replication schedule.

FlexVol volumes can also be increased or decreased in size on the fly. They also have another very important attribute. Space that is allocated to FlexVol but not used can be taken away on the fly and reallocated to another FlexVol volume that needs it.

FlexVol volumes can also be cloned using FlexClone technology.

Capacity guarantees, a new storage management concept set at the volume level, determine how the aggregate preallocates space to a flexible volume. This allows administrators to effectively implement the concept of "thin provisioning." Thin provisioning enables the administrator to in effect oversubscribe storage safely. Provision your storage space once, and then grow as needed in the aggregate.

When you create a FlexVol volume within an aggregate, you specify the capacity and, optionally, the type of guarantee. There are three types of guarantees, volume, file, and none.

We use the volume guarantee type, which makes sure that the amount of space required by the flexible volume is always available from its aggregate. This is the default setting for flexible volumes.

3.4 Qtrees

A qtree is a logically defined file system that can exist as a special subdirectory of the root directory within either a traditional volume or a flexible volume. You can create up to 4,995 qtrees per volume. Snapshot can be enabled or disabled for individual volumes but not for individual qtrees.

You can create a qtree to assign user- or workgroup-based soft or hard usage quotas to limit the amount of storage space that a specified user or group of users can use on the qtree to which they have access.

You can use qtrees regardless of whether quotas are enabled on your storage system. There are no restrictions on how much disk space can be used by the qtree or how many files can exist in the qtree.

3.5 FlexClone

FlexClone is a very powerful feature introduced in Data ONTAP 7G that adds a new level of agility and efficiency to storage operations by allowing an individual to create an instant writable clone of a flexible volume (FlexVol volume).

A FlexClone volume is a writable point-in-time image of a FlexVol volume or another FlexClone volume. With FlexClone, it takes only a few seconds to create a clone of a FlexVol volume, and such a volume can be created without interrupting access to the parent volume on which the clone is based.

The clone volume uses space very efficiently, allowing both the original FlexVol volume and the FlexClone volume to share common data, storing only the data that changes between the original volume and the clone. This provides a huge potential saving in storage space, resources, and cost. In addition, a FlexClone volume has all the features and capabilities of a regular FlexVol volume, including the ability to be grown or shrunk and the ability to be the source of another FlexClone volume.

FlexClone volumes also enable administrators to access the destination mirror created through the NetApp SnapMirror product. Previously, it was necessary to break the mirror in order to make any changes to the destination copy. With FlexClone, an administrator can now clone a Snapshot copy held in the mirror and make it available for both reading and writing at the remote site while allowing the mirror facility to continue running unaffected.

3.6 Snapshot Technology

A Snapshot copy is a locally retained point-in-time image of data. NetApp Snapshot technology is a feature of the WAFL storage virtualization technology that is a part Data ONTAP, the microkernel that ships with every NetApp storage system.

The WAFL file system can "freeze frame" itself (create a Snapshot copy) at any point in time and make the “frozen” versions of the file system available using "special" subdirectories that appear in the current (active) file system. Each frozen frame version of the file system is called a Snapshot copy.

The high performance of NetApp Snapshot also makes it highly scalable. A NetApp Snapshot copy takes only a few seconds to create—typically less than one second, regardless of the size of the volume or the level of activity on the NetApp storage system. After a Snapshot copy has been created, changes to data objects are reflected in updates to the current version of the objects, as if Snapshot copies did not exist. Meanwhile, the Snapshot version of the data remains completely stable. A NetApp Snapshot copy incurs no performance overhead; users can comfortably store up to 255 Snapshot copies per WAFL volume, all of which are accessible as read-only and online versions of the data.

Data ONTAP deletes Snapshot copies automatically; however, you might want to delete Snapshot copies before the preset time.

Deleting a Snapshot copy frees up disk space, but you should choose carefully which Snapshot copy to delete to free up the disk space you need for the file system.

For details, refer to the NetApp Data Protection Online Backup and Recovery Guide at http://now.netapp.com/NOW/knowledge/docs/ontap/rel7121/html/ontap/onlinebk/snap17.htm.

System administrators use Snapshot copies to facilitate frequent, low-impact, user-recoverable backups of files, directory hierarchies, LUNs, and/or application data.

Snapshot copies provide near-instantaneous, secure, user-managed restores. Users can directly access Snapshot copies to recover from accidental deletions, corruptions, or modifications of their data. Since the security of the file is retained in the Snapshot copy, the restoration is both secure and simple.

3.7 Snapshot Schedule

A Snapshot copy can be scheduled to occur automatically or be created manually. Automatic schedules can be created on an hourly, nightly, or weekly basis. When you install Data ONTAP on a storage system, it creates a default Snapshot schedule. The default Snapshot schedule automatically creates one nightly Snapshot copy Monday through Saturday at midnight, and four hourly Snapshot copies at 8 a.m., noon, 4 p.m., and 8 p.m. Data ONTAP retains the two most recent nightly Snapshot copies and the six most recent hourly Snapshot copies and deletes the oldest nightly and hourly Snapshot copies when new Snapshot copies are created.

3.8 SnapMirror

The Data ONTAP SnapMirror feature allows an administrator to mirror Snapshot copies of volumes or qtrees from a source volume or qtree to a destination volume or qtree. Replication of data can be performed at regular intervals to make the information available at the destination volume or qtree.
The result of this process is an online, read-only volume or qtree that contains the same data as the source at the time of the most recent update.
SnapMirror requires a license code. The basic deployment of SnapMirror consists of the following basic components, source volumes or qtrees, destination volumes or qtrees.

3.9 Modes of SnapMirror

SnapMirror can operate in three different types of modes: asynchronous, synchronous, and semisynchronous, namely SnapMirror Async, SnapMirror Sync, and SnapMirror Semi-Sync.

In asynchronous mode, SnapMirror performs incremental, block-based replication as frequently as once per minute. Because asynchronous replication is periodic, SnapMirror is able to consolidate writes and conserve network bandwidth. There is minimal impact on write throughput and write latency. As soon as data is written to the NVRAM of the source system, applications using this data are free to continue processing, without waiting for the data to reach a destination system. Updates take place in the background, so the application does not experience any additional transaction latency.

SnapMirror in synchronous mode is a mode of replication that sends updates from the source to the destination as they occur, rather than according to a predetermined schedule. This guarantees that data written on the source system is protected on the destination even if the entire source system fails. In synchronous mode, SnapMirror immediately replicates all data written to the source file system. This guarantees zero data loss in the event of a failure, but can have a significant performance impact. It is not necessary or appropriate for all applications.

A semisynchronous mode, which minimizes data loss in a disaster, also minimizes the extent to which replication impacts the performance of the source system and is also provided. Unlike asynchronous mode, which can replicate either volumes or quota trees, synchronous and semisynchronous modes work only with volumes.

As the changes to product binaries and configuration files of Oracle Fusion Middleware are not very frequent, we recommend using the asynchronous mode of SnapMirror.

3.10 Asynchronous Mode

In asynchronous mode, SnapMirror performs incremental, block-based replication as frequently as mentioned in the SnapMirror schedule setup. Performance impact on the source storage system is minimal as long as the system is configured with sufficient CPU and disk I/O resources.

The first and most important step in asynchronous mode involves the creation of a one-time baseline transfer of the entire data set. This is required before incremental updates can be performed. This operation proceeds as follows:

1. The source storage system creates a Snapshot copy (a read-only, point-in-time image of the file system).

2. This Snapshot copy is called the baseline Snapshot copy.

3. All data blocks referenced by this Snapshot copy and any previous Snapshot copies are transferred and written to the target storage system.

4. After initialization is complete, the source and target storage systems will have at least one Snapshot copy in common.

After initialization, scheduled or manually triggered, updates can occur. Each update transfers only the new and changed blocks from the source to the target storage system. This operation proceeds as follows:

1. The source storage system creates a Snapshot copy.

2. The new Snapshot copy is compared to the baseline Snapshot copy to determine which blocks have changed.

3. The changed blocks are sent to the target storage system and written to the file system.

4. After the update is complete, both file systems have the new Snapshot copy, which becomes the baseline Snapshot copy for the next update.

Because asynchronous replication is periodic, SnapMirror is able to consolidate writes and conserve network bandwidth.

3.11 SnapMirror Process

To use volume SnapMirror, you first create a restricted volume to be used as the destination or target for data replication. The destination/target volume is also referred to as the “SnapMirror volume.” The SnapMirror volume must be at least as large as the source and can reside on the same storage system as the source volume or on another storage system.

To replicate data for the first time, the storage system transfers all of the Snapshot copy's data from the source to the target. After the storage system finishes transferring the data, it brings the target volume online. This version of the Snapshot copy acts as the baseline for future incremental changes.

To make incremental changes to the mirror, the storage system creates regular Snapshot copies of the source according to the schedule specified in the /etc/snapmirror.conf file. By comparing the current Snapshot copy with the previous Snapshot copy, the storage system determines what updates it needs to make to synchronize data from the source to the target.

The SnapMirror process is initiated and driven by the target storage system.

To start replication, the target storage system establishes a TCP connection with the source storage system and initiates a transfer of all blocks. This is referred to as a *baseline transfer*, as all blocks of the source file system are transferred to obtain a baseline on the target (mirror).

After the baseline transfer is completed, the target volume on the target storage system is complete, consistent, and identical to the read-only copy on the source storage system. All files and directories on the target storage system are in a state identical to those in the source storage system and can be read and viewed from network clients that are connected to the target storage system.

The SnapMirror volume provides read-only access to users. Note that volume-based SnapMirror also transfers all Snapshot copies not from SnapMirror that existed on the source volume at the time that source storage system Snapshot copy was created.

During the baseline transfer, network clients of the source storage system might have been actively changing the source. However, the WAFL copy-on-write policy means that changes that

occurred during the baseline transfer will reside in a new set of disk blocks. To continue the automated replication process, the changed blocks need to be transferred from the source storage system so that the target storage system can be updated to the new state.

To do this, the target storage system arranges for the source storage system to create another Snapshot copy. It then initiates another TCP connection to transfer just the blocks that changed between the last transfer (the baseline transfer) and the time that the new Snapshot copy was created.

3.12 Volume SnapMirror

SnapMirror volume replication is a block-for-block replication. SnapMirror bypasses the WAFL file system layer and directly performs RAID I/O.

The original source volume must be read-write. The destination or target volumes are read-only. In a cascade, only the original source is read-write.

The target volumes must be restricted before the baseline transfer. It comes online on the completion of the baseline transfer.

A volume SnapMirror update transfers all Snapshot copies created since the last update.

Volume replication may be synchronous or asynchronous. Volume SnapMirror can be used to tape functions and cascading features if you replicate a volume.

There is a restriction that the destination Data ONTAP version must be the same or a higher release than the source Data ONTAP version. The same version is usually recommended so that there are no issues when a reverse resync is required.

4 SYSTEM CONFIGURATION

The following are the hardware and software configurations of the system used for developing this technical report.

4.1 Hardware Configuration

Production and standby hosts:

Dell PowerEdge 2850
Oracle Enterprise Linux® 4.0 update 5
Kernel 2.6.9-55.0.0.0.2: Intel® Xeon(® CPU 3.20GHz: 4CPUs
Memory 12GB

Storage FAS systems: NetApp FAS960/FAS2020

4.2 Software Configuration

NetApp Data ONTAP 7.2 or greater
NFS, FlexClone, and SnapMirror software licenses

Oracle Fusion Middleware 10*g*/Oracle Application Server 10*g* with SOA, IDM, and SSO

- Oracle Fusion Middleware 10.1.3 release
- Oracle Fusion Middleware 10.1.4 Identity Management release
- Oracle Fusion Middleware 10.1.2.x releases

4.3 Storage System Configuration

4.3.1 Update /etc/hosts File

The storage systems must be able to communicate with the application hosts and vice versa. A storage system can communicate to application hosts if there is an entry in its /etc/hosts file for the application hosts or, alternatively, if it uses some other hostname resolution techniques such as NIS or DNS. By default, the /etc/hosts file is checked first for hostname resolution.

The easiest way to update the /etc/hosts file on the storage system is by using the FilerView® GUI tool.

Go to FilerView > Click Network > Click Manage Hosts File

OR

Log into storage systems and enter the commands:

- wrfile /etc/hosts
- 10.61.162.100 *oidhost1.mycompany.com*
- 10.61.162.200 *oidhost2.mycompany.com*
- *Press Ctrl C*
- *That will save the file /etc/hosts.*

You can read the file by issuing the command rdfile /etc/hosts.

Entries made in the /etc/hosts file should look similar to the following one:

[HostIP] [HostName]
Where
• HostIP identifies the IP address assigned to the application hosts.
• HostName identifies the name assigned to the application hosts.

For example:
10.61.162.100 oidhost1.mycompany.com
10.61.162.101 oidhost2.mycompany.com

4.3.2 Enable 'rsh'/'ssh' Access for the Application Hosts

In order to use the 'rsh' (remote shell) command from the application hosts, you need to perform two steps. First, enable the 'rsh', 'ssh' option on the storage system:

Log in to storage system and issue the following commands:

> options rsh.enable on
> options ssh.enable on

Then, add the application hosts and username entry to the /etc/hosts.equiv file found on the storage system. The entry to this file looks somewhat similar to the following:

[*HostName] [UserName]*
Where
• HostName identifies the name assigned to the application host.
• UserName identifies the user name who needs rsh access to the storage system.

For example, to allow 'rsh' command execution from an application host named apphost1 for a user named oracle, you would add the following line to the /etc/hosts.equiv file on the storage system.

oidhost1.mycompany.com orauser
oidhost2.mycompany.com orauser

You can perform the same using FilerView:

Go to FilerView > Click Secure > Manage SSH
Go to FilerView > Click Security > Manage Rsh Access

4.4 Set Up the SnapMirror Relationship

4.4.1 Replication Prerequisites

1. You must purchase a SnapMirror license. If the source and the destination (mirror) will be on different storage systems, you must purchase a separate license and enter a unique SnapMirror license code for each storage system.

2. The source volume must be online.

3. If you are replicating a volume, you must create a restricted volume to be used as the SnapMirror volume. SnapMirror does not automatically create a destination volume as it does for mirror qtrees. This destination volume must not be a storage system's root volume.

4. A destination qtree cannot be /etc. The destination can be a qtree in a root volume, but it cannot be the /etc directory.

5. The capacity of the SnapMirror destination must be greater than or equal to the capacity of the source. The configuration, however, can be different. For example, the RAID group size could be different for two volumes.

4.4.2 Apply License Key for SnapMirror

SnapMirror is a licensed product; therefore, you need to apply the appropriate license key for SnapMirror on the storage systems used in your DR environment.
You can add license keys by executing the following command on a storage system:

license add [KeyCode]
Where *KeyCode* identifies the license key required for SnapMirror.

4.4.3 Allow Access to SnapMirror Destination Storage

In order to start SnapMirror replication, the SnapMirror destination storage system must have access to the SnapMirror source storage system. You can grant access by updating an option named *snapmirror.access*. The default setting for the option *snapmirror.access* is legacy, but you can change this by executing the following command on the source storage system:

options snapmirror.access host=[StorageSystemName]
Where
StorageSystemName identifies the name assigned to the SnapMirror destination storage system.

For example:
options snapmirror.access host=stbystor.netapp.com

4.4.4 Enable SnapMirror

Finally, the SnapMirror feature must be enabled on the both source and the destination storage systems by executing the following command:

options snapmirror.enable on

4.4.5 Identify the Source and Target Volumes

Identify the volumes that are used for the primary and standby application hosts to store Oracle home and other product-specific home directories. The volumes that are used for the primary site are going to be the source, and the volumes that are used for the standby site are going to be the targets for SnapMirror.

For the topology described in Figure 1, we will need to create six volumes for 12 ORACLE_HOMEs. This is because hosts in the deployment are clustered, and the Oracle homes for each of the host pairs need to be installed in the same volume to make sure of consistency. This is due to the fact that SnapMirror guarantees replication consistency at the volume level.

For more topology details, refer to the Oracle Application Server Disaster Recovery Guide.

The table below summarizes the different volumes and mountpoints to be created from the servers.

Usage	Source Volume Name	Mountpoint on Hosts	Target Volume Name
For Mid Tier	VOL_WEBSVR	/u01/app/oracle/webserver	VOL_WEBSVR_M
For SOA suite	VOL_SOA	/u01/app/oracle/soahome	VOL_SOA_M
For OHS	VOL_OHS	/u01/app/oracle/ohshome	VOL_OHS_M
For IM	VOL_IDM	/u01/app/oracle/imhome	VOL_IDM_M
For Midtier Apps	VOL_APPSVR	/u01/app/oracle/appsvrhome	VOL_APPSVR_M
For OID	VOL_OID	/u01/app/oracle/oidhome	VOL_OID_M

**Source and target volume names can be same in the storage.

The table below shows the flexible volume that will be created for each host pair.

Host Pair	Volume for Host Pair
OIDHOST1 and OIDHOST2	/vol/VOLOID
WEBHOST1 and WEBHOST2	/vol/VOL_WEBSVR
APPHOST1 and APPHOST2	/vol/VOL_APPSVR
WEBHOST3 and WEBHOST4	/vol/VOL_OHS
IDMHOST1 and IDMHOST2	/vol/VOL_IDM

Volume Creation

Create all the volumes described above. To create a volume on the storage systems that is in a production (primary) site or DR (standby) site:

Go to FilerView > Click Volumes > Click Add

Now the Volume Wizard, a popup window, appears; follow the screen prompt to complete the steps.

PRODSTOR> vol create oidvol -s volume aggr1 5g

Creation of volume 'oidvol' with size 5g on containing aggregate 'aggr1' has completed.

When you create a volume on the storage system which is in the standby site, make sure the size of the volume is equal to or greater in size than the one on the source storage system.

4.4.6 Configure Replication

For each volume created in the previous section, we need to configure the replication setting. This configuration is specified in the snapmirror.conf file. Hence, create the snapmirror.conf file under the /etc directory, which resides on the destination storage system. It controls where data is copied and how often a mirror is updated. The storage system is not shipped with a default */etc/snapmirror.conf* file. You must use a text editor to create the file. You can modify the */etc/snapmirror.conf* file at any time. The *snapmirror.conf f*ile is in the following format:

source_storage system:[vol|qtree path] destination_storage system:[vol|qtree path] arguments schedule

For example:
*PRODSTOR:oidvol STBYSTOR:mirror_oidvol - 45 10,11,12,13,14,15,16 * 1,2,3,4,5*

Target storage system updates */vol/mirror_oidvol* at 10:45 a.m., 11:45 a.m., 12:45 p.m., 1:45 p.m., 2:45 p.m., 3:45 p.m., and 4:45 p.m., Monday through Friday. The asterisk (*) in this example means that the mirror update schedule is not affected by the day of the month. Note the range syntax used in the second example. In the third example, the *0-55/5* specifies that data is replicated every five minutes every hour.

If SnapMirror is enabled, the changes take effect within two minutes. If SnapMirror is disabled, the changes take effect immediately after you enter the snapmirror on command to enable the feature.

4.4.7 Configure Remote Access

This is achieved by specifying the remote storage names in the snapmirror.access option. The *snapmirror.access* option performs the same function as the */etc/snapmirror.allow* file. This option, when issued on the source storage system, specifies the destination storage systems that are allowed to replicate (pull) data from the source storage system. You can specify hostnames,

IP addresses (including a range of addresses), and/or logical statements to identify destination storage systems. To be compatible with older versions of Data ONTAP, the default value for this option is legacy.

Note: If the */etc/snapmirror.allow* file exists on the source storage system and you also use the snapmirror.access option, the option's settings will take precedence over the file.

Optionally, you can configure remote accessing using the IP address of the target storage system. This can be done by using the *snapmirror.checkip.enable* option in the */etc/snapmirror.allow* file. If desired, you can specify a mix of IP addresses and hostnames, so long as there is no more than one storage system specified on each line in the file. To be compatible with older versions of Data ONTAP, the default value for this option is off.

4.5 Network Configuration

Using a private/dedicated network for SnapMirror is recommended because it can relieve the pressure that SnapMirror puts on a network. It is also recommended to use the throttle option, which can be configured. See NOW™ (http://now.netapp.com) for details.

4.6 Firewall Configuration

SnapMirror uses the typical socket/bind/listen/accept sequence on a TCP socket. SnapMirror source binds on port 10566. The destination storage system contacts the SnapMirror source storage system at port 10566 using any of the permissible, available ports assigned by the system. A firewall set in the customer scenario must allow requests to this port of the SnapMirror source storage system. The source storage system listens on TCP ports 10566 and 10569. The destination storage system listens on TCP ports 10565, 10567, and 10568. Hence, allowing a range of TCP ports from 10565 to 10569 is recommended.

4.7 Protocol Configuration (NFS)

Data ONTAP supports the NFS (Network File System), PC NFS (Personal Computer NFS), CIFS (Common Internet File System), and HTTP (HyperText Transmission Protocol) protocols on a storage system. With Data ONTAP multiprotocol features, a single set of data can be accessed using both NFS and CIFS when the data is protected by Windows® file security or UNIX® file security.

NFS Export

When Data ONTAP receives a mount request from a client, it compares the path name in the mount request to the path names of the exported resources contained in the */etc/exports* file. If Data ONTAP finds a match, the NFS client can mount the resource.

FilerView enables you to insert and modify export lines in the file, add options to an existing line for specific hosts, and delete an existing export line. You can then export all resources for client access. You can access it from FilerView using the below path.

FilerView > NFS > Manage NFS Exports

NFS Wizard -> Export options - Select Anonymous user id, Read-Write Access, Security, Root Access

Follow the screen: Anonymous user id – enter value 0 to give root access to all hosts.

> Click Export All.

The */etc/exports* file contains a list of resources that can be exported. When the storage system is rebooted, Data ONTAP exports all resources in this file.

Mount the Volumes from Application Hosts

For application server files and mountpoints, NetApp supports the use of TCP as the data transport mechanism with the current NFS V3.0 client software on the host. We need to create entries in the /etc/fstab file in primary and standby application hosts for corresponding volumes in the storage.

For example, OIDHOST1 in both primary and standby sites, following entries in */etc/fstab* need to be created.

Sample entries in /etc/fstab:

```
/ prodstor:/vol/oidvol       u01/voloidmount       nfs
rw,hard,nointr,rsize=32768,wsize=32768,vers=3,tcp,noac,timeo=600,actimeo=0 0 0

stbystor:/vol/appvol     /u01/ /volappmount       nfs
rw,hard,nointr,rsize=32768,wsize=32768,vers=3,tcp,noac,timeo=600,actimeo=0 0 0
```


4.8 Creating Directories and Symbolic Links from Application Hosts

Next step is to create the mountpoint from each of the application hosts to their corresponding volume in the storage. Below describes the steps for OIDHOST1 and OIDHOST2, and similar steps to be performed for other hosts. For example:

OIDHOST1: /u01/app/oracle/oid_oh --> /vol/vol_oid_oh1 (Oracle Home)
OIDHOST2: /u01/app/oracle/oid_oh --> /vol/vol_oid_oh2 (Oracle Home)

Let's assume that the /u01 directory already exists on OIDHOST1 and OIDHOST2.

1. A volume for OID has already been created in the storage system referred to as /vol/voloid. Now ORACLE_HOME directories in OIDHOST1 and OIDHOST2.

2. Log in as root on OIDHOST1 and create the following directories:

   prompt> mkdir /u01/app/oracle
   prompt> mkdir /u02/voloidmount

3. On OIDHOST1, add the following entry to the /etc/fstab file so that the /vol/voloid volume on the storage can be mounted to the /u02/storagevol directory on OIDHOST1:

   PRODSTOR:/vol/oidvol /u02/storagevol nfs

hard,nointr,proto=tcp,suid,rw,bg,vers=3,rsize=32768,wsize=32768,actime=0,timeo=600

4. While logged in as root on OIDHOST1, mount the storage volume and export:

   prompt> mount /u02/voloidmount
   prompt> exportfs -a

5. On OIDHOST1, create the Oracle home directory for the Oracle Application Server instance in the storage:

   prompt> cd /u02/voloidmount
   prompt> mkdir oid1_oh

6. On OIDHOST1, create a symbolic link named oid_oh in the /u01/app/oracle directory to the /u02/storagevol/oid1_oh directory on the storage:

   prompt> cd /u01/app/oracle
   prompt> ln -s /u02/voloidmount/oid1_oh oid_oh

7. On OIDHOST1, the following command changes the working directory to the /vol/voloid/oid1_oh directory on the storage.

   prompt> cd /u01/app/oracle/oid_oh

8. On OIDHOST1, run the Oracle Application Server installation procedure. When prompted by the installation procedure to provide an Oracle home directory to install the Oracle Application Server instance into, specify the /u01/app/oracle/oid_oh directory. This is a symbolic link to the vol/voloid/oid1_oh directory on the storage, so the Oracle Internet Directory instance will be installed into the /vol/voloid/oid1_oh directory on the storage.
9. Log in as root on OIDHOST2 and create the following directories:

   prompt> mkdir /u01/app/oracle
   prompt> mkdir /u02/voloidmount

10. On OIDHOST2, add the following entry to the /etc/fstab file so that the /vol/voloid volume on the storage can be mounted to the /u02/storagevol directory on OIDHOST2:

    1.2.3.4:/vol/oidvol /u02/voloidmount nfs
    hard,nointr,proto=tcp,suid,rw,bg,vers=3,rsize=32768,wsize=32768,actime=0,timeo=600

11. While logged in as root on OIDHOST2, mount the storage volume:

    prompt> mount /u02/voloidmount

12. On OIDHOST2, create the Oracle home directory for the Oracle Application Server instance in the storage:

    prompt> cd /u02/voloidmount
    prompt> mkdir oid2_oh

13. On OIDHOST2, create a symbolic link named oid_oh in the /u01/app/oracle directory to the /u02/storagevol/oid2_oh directory on the storage:

```
prompt> cd /u01/app/oracle
prompt> ln -s /u02/voloidmount/oid2_oh oid_oh
```

14. On OIDHOST2, the following command changes the working directory to the /vol/voloid/oid2_oh directory on the storage.

    ```
    prompt> cd /u01/app/oracle/oid_oh
    ```

15. On OIDHOST2, run the Oracle Application Server installation procedure. When prompted by the installation procedure to provide an Oracle home directory to install the Oracle Application Server instance into, specify the /u01/app/oracle/oid_oh directory. This is a symbolic link to the vol/voloid/oid2_oh directory on the storage, so the Oracle Internet Directory instance will be installed into the /vol/voloid/oid2_oh directory on the storage.

Use similar steps to install and configure the necessary Oracle homes for the other host pairs. For details specific to your deployment, check the *Oracle Application Server Disaster Recovery Guide*.

5 PLANNED AND UNPLANNED DOWNTIME

In the event that the DR/standby facility needs to be made operational, applications can be switched over to the servers at the DR/standby site and all application traffic directed to these servers until the primary site is recovered.
Once the primary site is online, SnapMirror can be used to transfer the data efficiently back to the production NetApp storage devices. After the production site takes over normal application operation again, SnapMirror transfers to the DR/standby facility can resume without requiring a second baseline transfer.

5.1 Initiate the Replication Process

Replication process is initiated by performing a *base line transfer* or *snapmirror initialize* command for each volume.

Once the initial transfer is started, changes to the source storage system volume will be replicated to the corresponding target storage system volume in incremental manner. The replication frequency is based on the setup in the */etc/snapmirror.conf* file.

Initialize the Destination or Target Manually

You must use the *snapmirror initialize* command to perform a complete (baseline) transfer of information whenever you start a replication or mirror for the *first time*. This process is known as *initializing a destination or target.*

You can initialize any time you want a baseline transfer, independent of the schedule in the *snapmirror.conf* file.

On the target storage system, enter the following commands:

1. **Restrict the destination volume:**

vol restrict <target volume name>
or
vol restrict <volume name> -mark volume <volume name> restricted

2. **Use the Initialize command (to start the baseline transfer):**

snapmirror initialize <target storage system>
or
snapmirror initialize –S <source storage system>:<volume name> <target storage system>: <volume name>

Example:
snapmirror initialize –S PRODSTOR:oidvol1 STBYSTOR:oidvol1

5.2 Replication Process Update

If there are any changes made to the source storage system, we can propagate the changes to target storage system either by manually forcing the replication for the changed volume or by using automatic update.

Automatic Update

SnapMirror updates the target storage system automatically according to the update schedule you specify in the */etc/snapmirror.conf* file.

Manual Update

You can initiate updates manually with the *snapmirror update* command.

You might need to run an unscheduled update to prevent data loss resulting from a scheduled or threatened power outage or from a target volume being taken offline for maintenance, repair, upgrade, or data migration. You can also include the *snapmirror update* command in an external script if you want to drive updates using that script.

On the target storage system, enter the following commands:

3. Run SnapMirror update:

snapmirror update –S <source storage system>:<volume name> <target storage system>:<volume name>

Example:
snapmirror update –S prodstor:oidvol1 stbystor:oidvol1

Above mentioned examples 1 through 3 are for OID volume; repeat the steps for all volumes created already in "section 4.4.5 - Identify the Source and Target Volumes ".

5.3 Switchover

Switchovers are planned operations done for periodic validation or to perform planned maintenance on the current production site.

During a switchover, the current standby site becomes the new production/primary site, and the current production/primary site becomes the new standby site.

1. Site switchover initiated by user.

2. Shut down applications and databases either manually or using any management or clustering software. (*For more information, refer to the Oracle Application Server Disaster Recovery Guide.)*

3. Perform SnapMirror Update from the target storage system site.

> Remote login (rsh) to the target storage system and perform the following:
>
> *snapmirror update –S <source storage system>:<volume name> -w <target storage system> :<volume name>*
>
> For example:
> *snapmirror update –S prodstor:oidvol1 -w stbystor:oidvol1*

4. Break all SnapMirror relationships between the two sites. This step makes all the target storage system volumes writable. You might need to mount the volumes if it is a fresh install.

> Remote login (rsh) to the target storage system and perform the following:
>
> *snapmirror break <target storage system>:<volume name>*
>
> For example:
> *snapmirror break stbystor:oidvol1*
>
> Also check the status of the volume:
>
> *snapmirror status*
> *vol status <volume name>*

5. Perform any DNS updates if necessary at the standby site (target storage system site).

6. Start applications at the DR site with the DR volumes (target storage system site).

7. Resynchronize the primary site volumes with DR volumes. This will reverse the replication flow, making the primary site as the new target and DR site as the new source. This step will sync up the primary site (PRODSTOR) volume from the newly written data from DR site (STBYSTOR) volume.

> ***Log in to new target system (in this case PRODSTOR) and run the following command:***
>
> *snapmirror resync –S <source storage system>:<volume name> <target storage system> :<volume name>*

For example:
rsh PRODSTOR
snapmirror resync -S stbystor:mirror_oidvol1 -w prodstor:oidvol1

Check the status of the volume:
vol status oidvol1

8. To continue performing updates from DR site (new source – STBYSTOR) to primary site (new target - PRODSTOR), perform *snapmirror update* to reflect the changes.

NOTE:
Repeat steps 1 through 8 in the switchover section to perform switchback to the original production site.
Replace the examples with the correct source and target storage system, as current source storage system will be STBYSTOR and target storage system will be PRODSTOR.

5.4 Failover

The process of making the current standby site the new production site after the production/primary site becomes unexpectedly unavailable (due to a disaster at the production/primary site):

1. Entire primary/production site is unreachable.
2. Break all SnapMirror relationships between the two sites. This step makes all the DR/standby site volumes writable.

Remote login (rsh) to the target storage system (DR site) and perform the following:

snapmirror break <target storage system>:<volume name>

For example:
snapmirror break stbystor:mr_oidvol1

3. Perform any DNS updates if necessary at the DR site.

4. Run the cleanup scripts provided by Oracle Fusion Middleware to remove lock files. (*For more information, refer to the Oracle Application Server Disaster Recovery Guide.)*

5. Start applications at the DR site with the DR volumes. For detailed steps, refer to the *Oracle Application Server Disaster Recovery Guide.*

6. When the primary site is back up, resynchronize the primary site volumes from the DR volumes. This makes the original source volume into a read-only target volume.

   If primary site volume (for example, oidvol1) **is recoverable:**

   If primary site volume is recoverable and its data is intact, to resynchronize primary site with DR site, perform the following:

   Log in to primary site (in this case PRODSTOR) and run the following command:

   snapmirror resync –S <source storage system>:<volume name> <target storage system> :<volume name>

   Example:
   rsh PRODSTOR
   snapmirror resync -S stbystor:mirror_oidvol1 prodstor:oidvol1

   This command makes the primary site volume (oidvol1) a read-only target volume.

   If primary site volume (for example, oidvol1) **is not recoverable:**

   Create a new volume (same name as it was before, for example, oidvol1) on primary site and initialize the new volume from DR site.

   Perform the following command from primary site:

   snapmirror initialize -S stbystor:mirror_oidvol1 prodstor:oidvol1

   This command also makes the primary site volume (oidvol1) a read-only target volume.

7. Update the primary site volume from DR site to transfer the latest data from DR site.

   Log in to primary site (in this case PRODSTOR) and run the following command:

   snapmirror update -S stbystor:mirror_oidvol1 prodstor:oidvol1

NOTE:
To fail back, perform steps 1 through 7 in the failover section and also perform below steps 8 and 9, thus making the original primary site as primary and original DR site as DR site again.

Replace the examples with the correct source and target storage system.

8. Make the primary site volume writable (making it a source again).

Perform the following command from primary site: *snapmirror break prodstor:oidvol1*

9. Make DR site as the target sit: (making it a target again):

Perform the following command from primary site: *snapmirror resync -S stbystor:mirror_oidvol1*

6 BEST PRACTICES

6.1 Manage Snapshot Copies

1. SnapMirror creates Snapshot copies on the source, which are copied to the destination. Do not delete these Snapshot copies because incremental changes to the destination depend on them. If the storage system cannot find the required Snapshot copy, it cannot perform incremental changes to the mirror.
2. The destination must have *snapmirrored* status. If you disable *snapmirrored* status using the vol options or *snapmirror break* command, the destination becomes a read/write volume. Disable the *snapmirrored* status only when you no longer need to update incremental changes from the source, for example, if you used SnapMirror for data migration.
3. Snapshot copies can be deleted using the following command:
   snap delete –v <volume name> <snapshot name>
   or
   You can login to FilerView > Click Snapshots > Click Manage > Select the Snapshot and Click Delete button

6.2 Disable SnapMirror

If you decide that data replication is no longer needed, you can disable replication for the entire storage system. You can disable the feature at any time, even when replication is under way.

To disable SnapMirror on a storage system, enter the following command.

Log in to the storage system and run the following command: *snapmirror off*

Or follow the path to FilerView > SnapMirror > Enable/Disable, in FilerView.

6.3 Replication Restart (Retry)

In SnapMirror, a *retry* is an automatic attempt to start the transfer process after an interruption, whether or not any data was successfully transferred. A *restart* is the resumption of a previous transfer process from a restart checkpoint.
If a transfer fails, because of network failure, for example, SnapMirror automatically *retries* the transfer. SnapMirror makes a restart checkpoint every five minutes during a transfer. If a restart checkpoint exists and conditions are right, SnapMirror *restarts* the previous transfer where it left off. If not, SnapMirror creates a new Snapshot copy and starts a new transfer.
After a reboot, SnapMirror does not automatically retry a transfer that was interrupted, but the next scheduled or manual transfer restarts it at the restart checkpoint, if the checkpoint is still valid.

A restart can happen if all of the following conditions are present:

• A restart checkpoint exists.
• All Snapshot copies being transferred still exist.
• The value for restart mode in the */etc/snapmirror.conf* file is set to *always* or is not set, and the next scheduled update has not arrived.

An initial transfer can be restarted but will not be retried automatically. To restart an initial transfer, you need to reenter the *snapmirror initialize* command. Scheduled incremental updates automatically retry the transfer. A manual update issued from the command line is not retried automatically.

6.4 Notes

SnapMirror works over a TCP/IP connection (also works over Fibre Channel) that uses standard network links. Thus, it allows for maximum flexibility in locating the source and destination storage systems and in the network connecting them. The nature of SnapMirror gives it advantages over traditional mirroring approaches.

The time required for a SnapMirror update is largely dependent on the amount of new data since the last update and, to some extent, on file system size. The worst-case scenario is where all data is read from and rewritten to the file system between updates. In that case, SnapMirror will have to transfer all file blocks. File system size plays a part in SnapMirror performance due to the time it takes to read through the active map files (which increases as the number of total blocks increase).

Another drawback of SnapMirror is that its Snapshot copies reduce the amount of free space in the file system.

On systems with a low rate of change, this is fine, since unchanged blocks are shared between the active file system and the Snapshot copy. Higher rates of change mean that SnapMirror reference Snapshot copies tie up more blocks. By design, SnapMirror only works for whole volumes as it is dependent on active map files for updates. Smaller mirror granularity could only be achieved through modifications to the file system or through a slower, logical-level approach.

7 COMMANDS EXPLAINED

SnapMirror Initialize

SnapMirror initialization (snapmirror initialize) performs the initial baseline transfer. Before initialization the destination volume must be restricted. After initialization the destination volume comes online. Once the initial baseline transfer has been performed and as long as SnapMirror relationship is maintained, the source and destination volumes will always have a common Snapshot copy to use as a baseline for the next update.

The baseline transfer, also called the level 0 transfer, can be performed over the network or using tapes. This transfer replicates the entire data set. Incremental updates transfer only the blocks that changed after the last update. Incremental updates replicate all Snapshot copies that exist on the source at the time of the update.

A Snapshot copy is created by SnapMirror before each transfer, including the baseline transfer. An update transfers all Snapshot copies created since the last update. After the update, the old SnapMirror Snapshot copy is deleted. The SnapMirror Snapshot copy just created is retained to serve as the newest common Snapshot copy for the next update.

If the newest common Snapshot copy is deleted, SnapMirror updates will fail.

Restrict the Volume

Go to FilerView and perform the following:

FilerView > Volume > Choose the volume name from the top down list > Click restrict button.

SnapMirror Break

SnapMirror destination volumes are restricted and not available for writes. In order to make these volumes available for write operations, you need to break the SnapMirror relationship after SnapMirror has completed the transfer of data. You can break a SnapMirror relationship by executing the following command on the SnapMirror destination storage system:

snapmirror break [StandbyStorageSystem]: [DestinationVolumeName]
Where
• *StandbyStorageSystem* identifies the name assigned to the storage system that is used for the standby site.
• *DestinationVolumeName* identifies the name assigned to a volume that is used by the standby site.

If you need to perform these from FilerView:

FilerView > SnapMirror > Manage >
> Select the volume and Quiesce it before you break the relationship.
> Click Quiesce
>Click Manage SnapMirror. Check the state – Quiesced.
>Click Break
Now you get a message like Snapmirror break: Destination voloid is now rewritable.

SnapMirror Update

A SnapMirror update replicates the incremental changes to the source since the last SnapMirror update. An update results in:

- Deletion of all Snapshot copies that have been deleted on the source since the last update
- Creation of new Snapshot copies that have been created on the source since the last update.
- Transfer of all blocks that have changed on the source since the last update

Resynchronizing SnapMirror

The snapmirror resync command can be used to restore or redefine a SnapMirror source or destination relationship that was broken with a snapmirror break command.

Resynchronization may be required when the destination was made writable for some reason (disaster recovery activation, testing) and needs to be made a SnapMirror destination again. The source and destination roles need to be reversed.

Resynchronization requires the source and the destination to have common SnapMirror Snapshot copies. The source and destination agree on the newest common SnapMirror Snapshot copy that the destination will revert back to. The negotiation takes place over the network, but the actual update does not involve data transfer since the Snapshot copy is already available.

8 TROUBLESHOOTING TIPS

Following are the details needed to troubleshoot any issues regarding the filer/storage.

• SnapMirror log files (/etc/logs/snapmirror, /etc/messages) on the source and destination storage systems

• snapmirror.conf file on the destination located under /etc/
• snapmirror status output (primary and standby)
• snap list on primary and standby

Question: Permission denied when you try to create a file in a directory that is already mounted.

For example:
>umount /u01/voloidmount
>mount /u01/voloidmount
>chown aime1:svrtech /u01/voloidmount
>cd /u01/voloidmount
>touch testfile
> Permission denied.

Answer: Make sure hostname entries are correct in /etc/hosts.equiv file in the storage systems.

Question: Trying to do snapmirror initialize command and it failed.
Answer: Check the snapmirror.access option and make sure the storage system name has been specified. Also make sure /etc/hosts has the right entry. Make sure you follow either fully qualified domain name or alias consistently.
For example:
>options snapmirror.access host=stbystor.mycompany.com
>options snapmirror.access host=stbystor.mycompany.com

Page: 34
To check the security permissions, issue a chown command on the physical directory that got mounted to the volume through /etc/fstab entries.

Question: How to verify the volume type?
Answer: There are three types: UNIX (which it should be set to), mixed, or NTFS.
Sample check:

rsh to the target storage system and perform the following: syntax: *qtree status|grep <volume_name>* *<volume_name> unix enabled normal* For example: *qtree status | grep oidvol* *oidvol unix enabled normal*

9 APPENDIX

9.1 Sample Scripts

1) To create volume on the source (primary) storage system.

```
#Script name:  create_source_volume.sh
#
#################
TIMESTAMP=`date +%m%d%y%H%M`; export TIMESTAMP
SCRIPT_HOME=`pwd`;export SCRIPT_HOME
LOG_FILE=$SCRIPT_HOME/create_source_vol.log
echo "Enter volume name>"
read sourcevol
echo "Enter Primary STORAGE name>"
read sourcearray
rsh ${sourcearray} aggr status
echo "Enter Aggregate name to create volume >"
read destaggr
echo "Enter volume size in Gb >"
read volsize
sizetype=g

echo "Enter the hostname(s) you intend to mount volume, if more than 1 use : as separator >"
read desthost
###############
#rsh ${sourcearray} vol create ${sourcevol} -s volume ${destaggr} ${volsize}${sizetype}   >>
$LOG_FILE
#rsh ${sourcearray} snap create ${sourcevol} superfly  >> $LOG_FILE
##rsh ${sourcearray} snap delete ${sourcevol} superfly   >> $LOG_FILE
#echo "Volume created successfully"
rsh ${sourcearray} vol create ${sourcevol} -s volume ${destaggr} ${volsize}${sizetype}
rsh ${sourcearray} snap create ${sourcevol} superfly
rsh ${sourcearray} snap delete ${sourcevol} superfly
# Replace the system created entry in /etc/exports for the newly created
# flexclone by removing the original entry and replacing with an entry
# that has site specific permissions and access
#
rsh ${sourcearray} exportfs -z /vol/${sourcevol}
rsh ${sourcearray} exportfs -p rw,root=${desthost},anon=0 /vol/${sourcevol}

rsh ${sourcearray} vol options ${sourcevol} nosnap on


Below part is optional, this allows users to view actual syntax that was run from above script
###############
# Added spool of actual syntax used during execution
##############
echo "Output values from executed script" > $SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "______Start ** Commands Section  **______" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "rsh ${sourcearray} vol create ${sourcevol} -s volume ${destaggr} ${volsize}${sizetype}" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "rsh ${sourcearray} snap create ${sourcevol} superfly" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "rsh ${sourcearray} snap delete ${sourcevol} superfly" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
```

```
echo "rsh ${sourcearray} exportfs -z /vol/${sourcevol}" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "rsh ${sourcearray} exportfs -p rw,root=${desthost},anon=0 /vol/${sourcevol}" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "rsh ${sourcearray} vol options ${sourcevol} nosnap on" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
echo "_____End ** Commands Section  **_____" >>
$SCRIPT_HOME/volume_${sourcevol}'_'syntax.log
```

2) initiate_mirror.sh

This script will be run only when needed to create a mirror relationship. This script creates volume in the standby storage system as same size as the source volume in the primary storage system and it creates the initial Snapshot copy.

```
##################
TIMESTAMP=`date +%m%d%y%H%M`; export TIMESTAMP
SCRIPT_HOME=`pwd`;export SCRIPT_HOME
echo "---------------------- FSTAB Entries ----------------------"
cat /etc/fstab |grep : | awk '{print $1}'
echo "-----------------------------------------"
echo "Enter source volume name appears after :/vol/<NAME>"
read sourcevol
echo "Enter destination volume prefix ex: mirror_ or clone_ >"
read destvol
echo "Enter Primary STORAGE name>"
read sourcearray
echo "Enter Secondary STORAGE name>"
read destarray
rsh ${destarray} aggr status
echo "Enter Aggregate name to create mirror volume >"
read destaggr
echo " Enter loop checking value in seconds >"
read chksec
###############
# SnapMirror
set -vx

volsize=`rsh ${sourcearray} vol size ${sourcevol} | awk {'print$8'} | cut -d'.' -f1`
rsh ${destarray} vol create ${destvol}${sourcevol} -s volume ${destaggr} ${volsize}
# Optional rsh ${destarray} snap reserve ${destvol}${sourcevol} 0
rsh ${destarray} vol restrict ${destvol}${sourcevol}
rsh ${sourcearray} snap create ${sourcevol} superfly
rsh ${destarray} snapmirror initialize -S ${sourcearray}:${sourcevol} ${destvol}${sourcevol}

ready=0

while [ $ready -ne 1 ];do
        rsh ${destarray} snapmirror status ${destvol}${sourcevol} | grep Idle
if [ $? -eq 0 ];then
        rsh ${destarray} snap list ${destvol}${sourcevol} | grep superfly
        if [ $? -eq 0 ];then
```

```
                ready=1
        fi
else
                sleep ${chksec}
fi
done
rsh ${sourcearray} snap delete ${sourcevol} superfly
```

3) create_snapshot.sh

This script will create a Snapshot copy on source volumes and update the destination volume.

```
TIMESTAMP=`date +%m%d%y%H%M`; export TIMESTAMP
SCRIPT_HOME=`pwd`;export SCRIPT_HOME
read sourcefiler?"Enter Source NetApp Filer >"
rsh $sourcefiler vol status
read sourcevol?"Enter Source NetApp Volume >"
read destfiler?"Enter Destination NetApp Filer >"
rsh $destfiler vol status
read destvol?"Enter Destination NetApp Volume >"
rsh $sourcefiler snap create $sourcevol $sourcevol'_'$TIMESTAMP
rsh $sourcefiler snap list $sourcevol
rsh $destfiler snapmirror update -w $destfiler:$destvol
```


9.2 FilerView

The FilerView administration tool is provided to manage your storage system through a standard Web browser. Using FilerView you choose the portion of the storage system that you want to manage, configure, or monitor, for example:

- Viewing the storage system's system configuration
- Managing disks
- Managing aggregates
- Managing volumes

To access the FilerView tool from the browser, enter http://<filer_name>/na_admin.
For example: http://prodstor.mycompany.com/na_admin.

9.3 References

Oracle Application Server Enterprise Deployment Guide 10*g* Release 3 (10.1.3.3.0)
http://download.oracle.com/docs/cd/E10291_01/core.1013/e10294.pdf

Data ONTAP 7.2.3 Documentation

http://now.netapp.com/NOW/knowledge/docs/ontap/rel723/

Data ONTAP 7.2 Data Protection Online Backup and Recovery Guide
http://now.netapp.com/NOW/knowledge/docs/ontap/rel723/pdfs/ontap/onlinebk.pdf

SnapMirror Best Practices Guide
www.netapp.com/library/tr/3446.pdf

Data ONTAP 7.2 Storage Management Guide
http://now.netapp.com/NOW/knowledge/docs/ontap/rel724/pdfs/ontap/mgmtsag.pdf

10 CONCLUSION

Using NetApp SnapMirror simplifies the Oracle Fusion Middleware replication process; the use of storage-level mirroring allows the copies to be created quickly, efficiently, and independently of the server. This maximizes the resources on the source server available for production/online use. The mirroring can also be started way ahead of time if possible so that only the last incremental changes need to be transferred during cloning, thus shortening the whole process.

This DR solution provides for an optimal process for Oracle Fusion Middleware replication. This in turn gives you flexibility in the frequency of when cloning is done to satisfy the cloning requirements of the enterprise, be it for development, testing, reporting, or whatever the case may be. SnapMirror is easy to set up, configure, and maintain and, most important, is cost-effective as a mirroring solution.

Using NetApp storage systems with the SnapMirror feature in conjunction with the Oracle DataGuard feature greatly simplifies and speeds up the Oracle Fusion Middleware replication process. This provides users with the maximum benefit out of their investment in the overall system.