Abstract

This document describes information and best practices related to configuring replication in NetApp® ONTAP®.
# Table of Contents

1 **Overview** .................................................................................................................. 5  
   1.1 Purpose and Intended Audience .................................................................................. 5  
   1.2 SnapMirror Overview ............................................................................................... 5  
   1.3 SnapVault .................................................................................................................. 8  
   1.4 Unified Data Protection .............................................................................................. 9  

2 **Requirements** ............................................................................................................ 9  
   2.1 SnapMirror Technology Requirements ..................................................................... 9  
   2.2 Clustered Data ONTAP Overview .............................................................................. 10  

3 **Architecture: Network Configuration for Replication Between Different Clusters** .......... 11  
   3.1 Intercluster Networking ............................................................................................ 12  
   3.2 Cluster Peering .......................................................................................................... 12  
   3.3 Cluster Peer Requirements ......................................................................................... 12  
   3.4 Intercluster Multipathing and Network Redundancy .................................................. 13  
   3.5 Network Connections for Intercluster SnapMirror ..................................................... 16  
   3.6 Determining Whether to Share or Dedicate Ports for Replication ............................ 17  
   3.7 Configuring Intercluster LIFs to Share Data Ports ..................................................... 18  
   3.8 Configuring Intercluster LIFs to Use Dedicated Ports .............................................. 18  
   3.9 Intercluster SnapMirror Throttle ................................................................................ 18  
   3.10 Firewall Requirements for Intercluster SnapMirror .................................................. 18  

4 **Interoperability** .......................................................................................................... 19  

5 **SVM Peering** ............................................................................................................. 19  

6 **SnapMirror Data Protection Relationships** .................................................................. 19  
   6.1 Networks Used for SnapMirror Data Protection Relationships ................................ 20  
   6.2 SnapMirror Data Protection Relationships ................................................................ 22  
   6.3 Scheduling SnapMirror Updates .............................................................................. 25  
   6.4 Converting a SnapMirror Relationship to a SnapVault Relationship ....................... 25  

7 **SnapMirror Load-Sharing Mirror Relationships** ........................................................ 28  

8 **SnapMirror Unified Replication** .................................................................................. 28  
   8.1 Default Policies .......................................................................................................... 28  
   8.2 Use Cases .................................................................................................................. 29  
   8.3 Technology Implications ........................................................................................... 32  
   8.4 Configuring SnapMirror: Unified Replication ............................................................ 33
8.5 Converting Default SnapMirror to SnapMirror: Unified Replication ........................................... 33
8.6 Secondary Snapshot Management: Unified Replication .............................................................. 35
8.7 Configuring Cascades: Unified Replication .................................................................................... 37

9 Storage Virtual Machine Disaster Recovery ...................................................................................... 38
9.1 Overview ....................................................................................................................................... 38
9.2 Options .......................................................................................................................................... 39
9.3 Requirements ............................................................................................................................... 40
9.4 Use Cases ...................................................................................................................................... 40

10 SnapMirror and Data ONTAP Feature Interaction ........................................................................ 41
10.1 SnapMirror and Snapshot Copies ............................................................................................... 41
10.2 SnapMirror and Qtrees ............................................................................................................... 41
10.3 SnapMirror and FlexClone .......................................................................................................... 42
10.4 SnapMirror and Infinite Volume ................................................................................................. 42
10.5 SnapMirror and NetApp Storage Efficiency .................................................................................. 43
10.6 SnapMirror and Volume Move .................................................................................................... 43
10.7 SnapMirror for Disk Shelf Failure Protection ............................................................................... 43
10.8 SnapMirror and Volume Autosize .............................................................................................. 44
10.9 SnapMirror and Network Data Management Protocol ............................................................... 44
10.10 SnapMirror and Data ONTAP Version Dependencies ............................................................... 45

11 Performance .................................................................................................................................. 46
11.1 SnapMirror and Network Compression ....................................................................................... 46
11.2 SnapMirror Throttling ................................................................................................................. 48
11.3 SnapMirror Sizing Recommendations .......................................................................................... 49
11.4 Concurrent Replication Operations ............................................................................................ 49
11.5 Recommended Replication Intervals ........................................................................................... 50
11.6 Network Sizing Requirements .................................................................................................... 50

12 Troubleshooting Tips ..................................................................................................................... 51
12.1 Troubleshooting Cluster Peer Relationships ........................................................................... 51
12.2 Troubleshooting SVM Peer Relationships ................................................................................. 52
12.3 Understanding SnapMirror Relationship Status ....................................................................... 52
12.4 Troubleshooting SnapMirror Relationships .............................................................................. 53

13 Best Practices for DR Configurations ............................................................................................ 55

14 Configuration and Failover for Disaster Recovery ....................................................................... 56
14.1 Environment Failover Requirements and Assumptions .............................................................. 56
1 Overview

There are several approaches to increasing data availability in the face of hardware, software, or even site failures. Backups provide a way to recover lost data from an archival medium (tape or disk). Redundant hardware technologies also help mitigate the damage caused by hardware issues or failures. Mirroring provides a third mechanism to facilitate data availability and minimize downtime. NetApp SnapMirror® technology offers a fast and flexible enterprise solution for mirroring or replicating data over local area networks (LANs) and wide area networks (WANs). SnapMirror is a key component in enterprise data protection (DP) strategies.

1.1 Purpose and Intended Audience

This document is intended for individuals who administer, install, or support clustered Data ONTAP operating systems and who intend to configure and use SnapMirror technology for data replication.

This document assumes that the reader has an understanding of the following processes and technologies:

- Storage systems administration: working knowledge of clustered Data ONTAP operating system operational processes
- Storage systems administration: working knowledge of NetApp features such as NetApp Snapshot™ copies, NetApp FlexVol® volumes, NetApp FlexClone® volumes, and NetApp Infinite Volumes
- General knowledge of disaster recovery (DR) and data replication solutions
- Familiarity with the ONTAP 9.3 Data Protection Guide on the NetApp Support site

1.2 SnapMirror Overview

SnapMirror software is a NetApp disk-to-disk replication solution that is built into Data ONTAP. Replication can be performed within the same cluster or remotely to another cluster. NetApp clustered Data ONTAP provides integrated data replication technologies for creating replica copies that can be used for DR, to offload tape backup processes from the primary, to distribute datasets to other locations, and to create read/write clones for test and development environments. The relationships created are referred to as data protection (DP) relationships. With SnapMirror, you can reduce total cost of ownership (TCO), making it easier to justify the DR investment by putting your DR site to active business use.

The SnapMirror technology in Data ONTAP 8.2 used a block-level replication engine. With Data ONTAP 8.3, a logical replication engine can also be used to perform SnapMirror DR tasks. These are referred to as extended data protection (XDP) relationships combined with a policy type of async-mirror and policy MirrorAllSnapshots (predefined). The logical replication engine provides many benefits:

- Single baseline copy for DR and backup
- Reduced network traffic
- Local Snapshot copies on the secondary
- Version flexible replication to simplify system upgrades

In general, a SnapMirror license must be installed on both source and destination clusters to enable the SnapMirror feature on your NetApp system.

For an overview of clustered Data ONTAP replication, refer to Figure 1.
Integrated Data Protection

Integrated Data Protection can be used to create an on-disk, quickly accessible history of application-consistent Snapshot copies that eliminates the concept of traditional backup windows. NetApp SnapMirror then replicates the history of Snapshot copies to the destination volumes, which can be used for backup, DR, or test and development.

SnapMirror replication is efficient because it only replicates the blocks that have changed or have been added since the previous update. Additional efficiency is gained when SnapMirror is combined with NetApp storage efficiency technologies. Compression and data deduplication technologies can result in telecommunication savings and significant storage capacity savings.

SnapMirror for Disaster Recovery

SnapMirror technology is an integral part of DR plans. If critical data is replicated to a different physical location, a serious disaster does not have to result in extended periods of unavailable data. Clients can access replicated data across the network until the damage caused by the disaster is repaired. Application servers at the recovery site can access replicated data to restore operations for business-critical applications for as long as necessary to recover the production site. Recovery might include recovery from corruption, natural disaster at the production site, accidental deletion, and so on.

In cases where a disaster requiring a failover occurs and the primary storage is not completely lost, SnapMirror provides an efficient means of resynchronizing the primary and DR sites. When the primary site is back online, SnapMirror resynchronizes the two sites, transferring only changed or new data back to the primary site from the DR site by simply reversing the SnapMirror relationships. After the primary production site resumes normal application operations, SnapMirror transfers to the DR facility resume without requiring another complete data transfer.
NetApp FlexClone Technology for Disaster Recovery Testing and Application Test/Development

NetApp FlexClone technology can be used to quickly create a read-write copy of a SnapMirror destination FlexVol volume, eliminating the need for additional copies of the data. For example, a 10GB FlexClone volume does not require another 10GB FlexClone volume. It requires only the metadata needed to define the FlexClone volume. FlexClone volumes only store data that is written or changed after a clone is created.

Data Distribution and Remote Data Access

SnapMirror technology can be used to distribute large amounts of data throughout the enterprise, enabling access to data at remote locations. Remote data access provides faster access to data by clients in the remote locations. It also allows more efficient and predictable use of an expensive network and server resources because WAN usage occurs at a predetermined replication time. Storage administrators can replicate production data at a specific time to minimize overall network utilization.

Backup Offloading and Remote Tape Archiving

SnapMirror technology can also be used for backup consolidation and for offloading tape backup overhead from production servers. This approach facilitates centralized backup operations, reducing backup administrative requirements at remote locations. Because NetApp Snapshot technology eliminates the traditional backup window on the primary storage system, offloading tape backup to a SnapMirror destination, as shown in Figure 2, dramatically reduces the overhead of backup operations on production storage systems.

Unified Architecture Flexibility

Starting with the clustered Data ONTAP 8.1 operating system, SnapMirror technology can be used between NetApp FAS systems. Systems with different performance characteristics and different costs can be deployed at the primary and DR sites. For example, depending on the capabilities required, the DR site might contain a lower-end platform, SATA disk versus Fibre Channel (FC) disk, or the iSCSI or Fibre Channel over Ethernet (FCoE) protocol versus FC. Figure 2 illustrates the flexibility within a unified architecture.
A unified architecture, from low-end platforms to high-end platforms, also allows system administrators to learn and use the same management and monitoring paradigm.

1.3 SnapVault

SnapVault® software (introduced in Data ONTAP 8.2) is a NetApp disk-to-disk backup solution that is built into NetApp Data ONTAP. SnapVault software allows you to replicate your data to a secondary volume and to retain the data for a longer period of time than you might on your primary volume.

Although former 7-Mode users will find similarities, major enhancements were made in this version of SnapVault software. One major advance is the ability to preserve storage efficiencies on primary data during SnapVault transfers. This technical report covers the details of storage efficiency preservation as well as other topics that changed from the 7-Mode version of SnapVault software.

One important architectural change is that SnapVault software in a clustered Data ONTAP storage operating system replicates at the volume level as opposed to the qtree level, as in 7-Mode SnapVault software. This fact means that the source of a SnapVault software relationship must be a volume and that volume must replicate to its own volume on the SnapVault secondary.

The new SnapVault technology developed in Data ONTAP 8.2 uses the logical level replication engine. These are referred to as extended data protection (XDP) relationships combined with a policy type of vault and policy XDPDefault (predefined). A logical replication engine provides the flexibility of version-independent backup replication.

Starting with ONTAP 9.0, a SnapMirror license is required to be installed on both source and destination clusters to enable the SnapVault feature on your NetApp system. Further details about SnapVault can be found in the SnapVault Best Practices Guide (TR-4183).
1.4 Unified Data Protection

The unified data protection or unified replication solution introduced in Data ONTAP 8.3 provides the combined benefits of SnapMirror and SnapVault. It’s a standardized multipurpose replication solution that can protect mission-critical business with simple, efficient replication for disaster recovery and extend the value of replicated data to accelerate business needs. It utilizes the logical replication engine, and the relationships are created using XDP with policy type mirror-vault and policy MirrorAndVault.

Functionality available in clustered Data ONTAP 8.3 and later versions removes the limitation of the destination controller needing to have a clustered Data ONTAP major version number equal to or higher than the major version of the source controller, allowing customers to have nondisruptive upgrades. In addition, the functionality only requires a single baseline and reduces the number of secondary Snapshot copies needed on the destination. Section 9 of this document and the SnapMirror Unified Replication FAQ on the Field Portal contain additional details about SnapMirror unified replication.

2 Requirements

2.1 SnapMirror Technology Requirements

SnapMirror technology in the clustered Data ONTAP operating system provides asynchronous volume-level replication based on a configured replication update interval. SnapMirror uses NetApp Snapshot technology as part of the replication process.

Clustered Data ONTAP 8.1 and later provides the following replication capabilities:

- **Data protection mirrors.** Replication to create a backup copy within the same cluster (intracluster) or to create a DR copy in a different cluster (intercluster).
- **Load-sharing mirrors.** Replication from one volume to multiple volumes in the same cluster to distribute a read-only workload across a cluster. Starting with ONTAP 9.1, load-sharing mirrors are being deprecated for data volumes and need to use a special option to enable it.

Basics of SnapMirror Replication

When the scheduler triggers a replication update, the following operations are performed:

1. A new Snapshot copy is created on the source volume.
2. The block-level difference between the new Snapshot copy and the last replication Snapshot copy is determined and then transferred to the destination volume. This transfer includes other Snapshot copies that were created between the last replication Snapshot copy and the new one.
3. When the transfer is complete, the new Snapshot copy exists on the destination volume.

A SnapMirror destination volume is available for read-only access if it is shared using the Common Internet File System (CIFS) protocol or exported using the Network File System (NFS) protocol. A logical unit number (LUN) in the replicated volume can be made available to a client that supports connection to read-only LUNs.
Replication occurs at the volume level. Qtrees can be created in the clustered Data ONTAP operating system and replicated along with the replicated volume. However, individual qtrees cannot be separately replicated.

Data protection (DP) relationships can be resynchronized in either direction after a failover without recopying the entire volume. If a relationship is resynchronized in the reverse direction, only new data written since the last successful synchronization Snapshot copy is sent back to the destination.

Starting with clustered Data ONTAP 8.2, a cluster administrator can delegate the management of SnapMirror relationships to a storage virtual machine (SVM) administrator.

2.2 Clustered Data ONTAP Overview

Some basic terms used in clustered Data ONTAP include:

- **Clustered Data ONTAP.** The Data ONTAP operating mode that supports interconnection of nodes into a cluster.
- **Node.** A single NetApp controller, one of a high-availability (HA) pair.
- **Cluster.** One or more nodes that are interconnected and managed as a single system.
- **Cluster interconnect.** A dedicated high-speed, low-latency, private network used for communication and replication between nodes in the same cluster.
- **Data network.** The network used by clients to access data.
- **Management network.** The network used for administration of the cluster, SVM, and nodes.
- **HA interconnect.** The dedicated interconnect between two nodes in one HA pair.
- **HA pair.** Two nodes configured in a pair for HA.
- **Physical port.** A physical port such as e0e or e0f or a logical port such as a virtual LAN (VLAN) or an interface group (ifgrp).
- **Virtual port:**
  - **Ifgrp.** A collection of physical ports combined to create one logical port used for link aggregation.
  - **VLAN.** A VLAN subdivides a physical network into distinct broadcast domains. As a result, traffic is completely isolated between VLANs unless a router (layer 3) is used to connect the networks. In clustered Data ONTAP, VLANs subdivide a physical port into several separate virtual ports, allowing for one of the key components of our secure multi tenant messaging: isolation of data.
- **Logical interface (LIF).** A LIF is an IP address or a worldwide port name (WWPN) that is associated with a port. It is associated with attributes such as failover groups, failover rules, and firewall rules. A LIF communicates over the network through the port (physical or virtual) to which it is currently bound.
- **Intercluster LIF.** A LIF that is used for cross-cluster communication, backup, and replication. You must create an intercluster LIF on each node in the cluster before a cluster peering relationship can be established. These LIFs can only fail over to ports in the same node. They cannot be migrated or failed over to another node in the cluster.
- **Intercluster network.** The network used for communication and replication between different clusters.
- **SVM.** A logical storage server that provides data access to LUNs and/or a network-attached storage (NAS) namespace from one or more LIFs.

There are multiple types of networks in a clustered Data ONTAP solution, as shown in Figure 33 and Figure 44. It is important to understand for what each network type is used.

The cluster interconnect network is a dedicated, high-speed, low-latency private network used for communication and replication between nodes in the same cluster. This configuration is a redundant back-end network that cannot be used or shared for client access to data or for managing the cluster, nodes, or SVMs. Client access to data occurs on the data network. Management of the cluster, nodes,
and SVMs occurs on the management network. The data and management networks might share the same ports or physical network. However, the data and management networks must be a different physical network than the cluster interconnect network.

Figure 3) Cluster interconnect and data and management networks.

An intercluster network is a network that allows communication and replication between two different clusters operating in clustered Data ONTAP, as shown in Figure 4. This network might be a network consisting of dedicated physical ports but could also be a network sharing ports with the data and/or management networks. The intercluster network is discussed in detail in the following section.

Figure 4) Intercluster network.

3 Architecture: Network Configuration for Replication Between Different Clusters

- **Cluster peering.** The act of connecting two clusters to allow replication to occur between them.
- **Intercluster LIFs.** Logical network interfaces used for intercluster communication.
- **Intercluster ports.** Ports dedicated to intercluster replication.

Clusters must be joined in a peer relationship before replication between different clusters is possible. Cluster peering is a one-time operation that must be performed by the cluster administrators.
Cluster peering must be performed because this configuration defines the network on which all replication between different clusters occurs. Additionally, starting in clustered Data ONTAP 8.2, SVMs must be joined in a peer relationship before replication between different SVMs is possible.

For additional information regarding intercluster networking, see the data protection guide appropriate for the version of clustered Data ONTAP you are running. Here is:

For example, the information relevant to clustered Data ONTAP 8.3.2 can be found on page 52 here.

### 3.1 Intercluster Networking

Cluster peer intercluster connectivity consists of intercluster LIFs that are assigned to network ports:

- Intercluster LIFs are node scoped. Therefore, when the port hosting an intercluster LIF fails, the LIF can fail over to only another intercluster-capable port on that node, as defined by the LIF’s failover policy. At least one intercluster LIF is required per node for replication between clusters. Maintain consistent settings between the intercluster LIFs (same maximum transmission units [MTUs], flow control, Transmission Control Protocol [TCP] options, and so on).
- SnapMirror replication over an FC network is not available in clustered Data ONTAP.
- If a node fails while an intercluster SnapMirror transfer is in progress, the transfer automatically continues using an intercluster LIF on the surviving node of the HA pair. In clustered Data ONTAP 8.2, the same transfer does not automatically continue after the storage failover (SFO) of the destination. If SFO happens on the source, the transfer continues. However, replication as such continues automatically from the surviving node.

For additional information regarding intercluster networking, see the data protection guide appropriate for the version of clustered Data ONTAP you are running. Here is the URL for Data ONTAP 8 documentation: http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=30092

For example, the information relevant to clustered Data ONTAP 8.3.2 can be found on page 53 here.

### 3.2 Cluster Peering

After the intercluster LIFs have been created and the intercluster network has been configured, cluster peers can be created. A cluster peer is a cluster that is allowed to replicate to or from another cluster.

Establishing cluster peering is a one-time operation that must be performed by the cluster administrators. A peer relationship can be created in two ways. In one method, a peer relationship is created by a cluster administrator who has security credentials (a cluster admin login and password) for the other cluster. The other method allows two administrators who do not want to exchange cluster admin passwords to peer their clusters. In this method, each administrator enters the cluster peer create command specifying intercluster IP addresses of the other cluster.

A cluster can be in a peer relationship with up to 256 clusters, allowing multiple clusters to replicate between each other.

### 3.3 Cluster Peer Requirements

Cluster peer requirements include the following:

- The time on the clusters must be in sync within 300 seconds (five minutes) for peering to be successful. Cluster peers can be in different time zones.
- At least one intercluster LIF must be created on every node in the cluster.
- Every intercluster LIF requires an IP address dedicated for intercluster replication.
- The correct MTU value must be used on the network ports that are used for replication. The network administrator can identify which MTU value to use in the environment. The default value of 1,500 is correct for most environments.
• All paths on a node used for intercluster replication should have equal performance characteristics.
• The intercluster network must provide connectivity among all intercluster LIFs on all nodes in the cluster peers. Every intercluster LIF on every node in a cluster must be able to connect to every intercluster LIF on every node in the peer cluster.

Note: See TR-4182, section 3.7 for in-depth information regarding cluster peering requirements and specific cluster peering configuration information.

3.4 Intercluster Multipathing and Network Redundancy

NetApp clustered Data ONTAP 8.1 provides the following capabilities to configure two kinds of multipathing for intercluster SnapMirror replication:

• **Active-passive.** Replication in which a particular path is used unless that path fails, in which case a different path is used.
• **Active-active.** Replication in which multiple paths are actively used at the same time. If one path fails, the surviving paths remain active, and all replication transfers continue.

**Active-Passive Intercluster Multipathing in Data ONTAP**

In many ways an intercluster LIF behaves in the same way as a LIF used for CIFS or NFS in terms of active-passive failover, except that an intercluster LIF cannot fail over to a port in a different node. The initial placement of a LIF on a specific port determines which port is used by that LIF. If ports are redundant for failover on the same node, the active path is the port where the initial LIF was placed. The passive path is any port where the LIF may fail over.

Therefore, it can be said that a properly configured redundant LIF provides active-passive multipathing, as shown in Figure 5.

Figure 5) Active-passive multipathing.

Communication on an intercluster LIF occurs on only the port to which the LIF is assigned unless that port fails, which causes the LIF to move to another surviving port in that LIF’s failover group.
To configure active-passive multipathing, assign an intercluster LIF to an intercluster-capable port and make sure that another intercluster-capable port is configured that is capable of supporting that connection. Make sure that the LIF’s failover policy is configured such that the LIF’s failover group contains the necessary ports to allow failover, as shown in Figure 6.

**Active-Active Intercluster Multipathing in Clustered Data ONTAP**

Active-active multipathing requires the configuration of additional intercluster LIFs on a node. SnapMirror uses all available intercluster LIFs on the source and destination nodes to send and receive data for all transferring SnapMirror relationships between those two nodes. If two intercluster LIFs are configured, and two ports are available for intercluster communication, then one LIF can be assigned to each port, and SnapMirror simultaneously uses both ports, as shown in Figure 7.

Starting in clustered Data ONTAP 8.2, SnapMirror multipathing with different types and speeds of networks is supported, without adversely affecting replication performance on the faster ports.

Communication occurs on both ports because an intercluster LIF is assigned to each port. If a port fails, the LIF that was on the failed port moves to another surviving port in that LIF’s failover group. Depending on the number of ports in the failover group, multiple LIFs can now share a port, as shown in Figure 8.
To configure two-path active-active multipathing for SnapMirror, create two intercluster LIFs and assign one LIF to each port. Make sure that each LIF’s failover policy is configured such that the LIF’s failover group contains the necessary ports to allow failover.

Depending on the replication workload between any given pair of source and destination nodes, it might be necessary to configure multiple paths on the source and destination nodes. There are no special configuration settings necessary to apply to each SnapMirror relationship to use the multipath connection. All SnapMirror relationships are automatically multiplexed across the available LIFs on the source and destination nodes.

### Switch-Based Link Aggregation for Multipathing

As mentioned earlier in this document, an intercluster LIF can be assigned to any kind of port in the system, including a logical port such as an ifgrp. An ifgrp supports switch-based link aggregation. Multiple physical ports can be configured into an ifgrp, and then the intercluster LIF can be assigned to that ifgrp port. The switch ports can then be combined using link aggregation technology as a method of providing multipathing and/or redundancy.

Switch-based link aggregation does not guarantee that multiple physical paths in the ifgrp are used simultaneously. For example, assume that a single intercluster LIF is configured on both the source and destinations nodes. Therefore, each node has one IP address to use for intercluster communication and a two-port ifgrp. If the ifgrp is using an IP hash-based method of load balancing, then there is only one pair of source and destination IP addresses on which to perform the load balancing hash. The link might place all connections between these two nodes on the same path within that port group.

Keep in mind that replication can take place between multiple nodes. For example, one node might replicate different volumes to different nodes in the remote cluster. Each node has different intercluster LIFs, which have different pairs of source and destination IP addresses that enable multiple paths within the link to be used for that particular source node.

If switch-based link aggregation is used to allow multiple physical paths in the ifgrp to be used when replicating between two particular nodes, additional intercluster LIFs can be configured on either of the two nodes. Data ONTAP automatically establishes a connection between every LIF on the source and destination node for SnapMirror. This approach provides additional combinations of source and destination IP addresses for the load balancing hash, which could be placed on different paths within the link. However, in this example the purpose of configuring multiple LIFs on one node is to enable multiple paths to be used for replication between any two particular nodes. This precaution is likely not necessary in many WAN replication scenarios because WAN bandwidth might be significantly less than the bandwidth of the combined links in the ifgrp. Enabling multiple paths between two particular nodes might not be beneficial, because many nodes must share the WAN bandwidth anyway.
Best Practice

When using switch-based link aggregation, create the ifgrp with a `multimode_lacp` mode and set the distribution function of the ifgrp to `port`. Using the port value for the distribution function configures the ifgrp to distribute connections across paths by hashing the source/destination IP address, as well as the port used. This practice does not guarantee that connections are evenly distributed across all paths in the ifgrp, but it does allow use of multiple physical links in the ifgrp.

3.5 Network Connections for Intercluster SnapMirror

In clustered Data ONTAP, the number of intercluster LIFs determines the number of TCP connections established between the source and destination nodes for SnapMirror. TCP connections are not created per volume or per relationship.

Remember that starting in clustered Data ONTAP 8.2, Data ONTAP establishes at least 12 intercluster TCP connections for sending data. A minimum of 12 TCP connections is created for sending data, as shown in Figure 9. This is true even if both the source and destination nodes have only one intercluster LIF and enough connections are created so that all intercluster LIFs on both the source and destination nodes are used.

Figure 9) TCP connections with one intercluster LIF.

If the source node, destination node, or both nodes are configured with 2 intercluster LIFs, then Data ONTAP establishes 12 TCP connections for sending data. However, instead of both connections using the same LIFs, one connection uses one LIF pair, and the other connection uses the other LIF pair, as shown in Figure 10. This example shows different combinations of intercluster LIFs that produce 12 intercluster TCP connections.

Note: It is not possible to select a specific LIF pair to use for a certain TCP connection. They are managed automatically by Data ONTAP.
After scaling past 12 intercluster LIFs on a node, Data ONTAP creates additional intercluster TCP connections, creating enough so that all intercluster LIFs are used.

The creation of additional intercluster TCP connections continues as more intercluster LIFs are added to either the source or the destination node. A maximum of 24 intercluster connections is currently supported for SnapMirror on a single node in Data ONTAP.

**Best Practice**

Although it is not required, the same number of intercluster LIFs can be configured on both the source and destination nodes for operational consistency. Multiple intercluster LIFs can be created to enable active-active multipathing across multiple physical paths, as described in the section titled “Switch-Based Link Aggregation for Multipathing.”

For example, if a node is configured with four 1 Gigabit Ethernet (GbE) ports for intercluster replication, then four intercluster LIFs are required, one assigned to each port to make sure all paths are used to provide bandwidth beyond just one GbE link.

### 3.6 Determining Whether to Share or Dedicate Ports for Replication

There are a number of configurations and requirements to consider when determining whether to share or dedicate ports for replication. They include:

- **LAN type.** GbE or 10GbE connectivity.
- **Available WAN bandwidth (compared to LAN bandwidth).** The WAN can act as a throttle if there is significantly less available WAN bandwidth than LAN bandwidth.
- **Replication interval.** Replication during nonproduction hours might have an irrelevant impact on the data network.
- **Change rate.** The amount of data required for replication may not interfere with client data access.
- **Number of ports used by the solution.** Dedicating ports for replication requires additional switch ports and cable runs.

Refer to the appropriate system administration guide for cluster administrators version relevant to the version of clustered Data ONTAP you are running for additional information to help determine which option is best for your environment. Click here for the clustered Data ONTAP 8.3 version. Page 317 begins the conversation regarding this step.
3.7 Configuring Intercluster LIFs to Share Data Ports

Refer to the appropriate system administration guide for cluster administrators version relevant to the version of clustered Data ONTAP you are running for a complete list of steps for configuring intercluster LIFs to share data ports. Click here for the clustered Data ONTAP 8.3 version. Page 319 begins the conversation regarding this step.

3.8 Configuring Intercluster LIFs to Use Dedicated Ports

Refer to the appropriate system administration guide for cluster administrators version relevant to the version of clustered Data ONTAP you are running for a complete list of steps for configuring intercluster LIFs to use dedicated data ports. Click here for the clustered Data ONTAP 8.3 version. Page 319 begins the conversation regarding this step.

Best Practice

As intercluster LIFs become available or unavailable, the list of active IP addresses can change. The discovery of active IP addresses is automatic in certain events, such as when a node reboots. The –peer-addrs option requires only one remote cluster address to be provided. However, if the node hosting that address is down and it becomes unavailable, then the cluster peer relationship might not be rediscovered. Therefore, it is a best practice to use at least one intercluster IP address from each node in the remote cluster, so, in the event of a node failure, the peer relationship remains stable.

3.9 Intercluster SnapMirror Throttle

To limit the amount of bandwidth used by intercluster SnapMirror, apply a throttle to intercluster SnapMirror relationships. When creating a new relationship, a throttle can be set through the command line by adding the –throttle option and a value in kilobytes, by modifying an existing relationship with the snapmirror modify command. In this example, a 10MB throttle is applied to an existing relationship using the snapmirror modify command.

```
cluster02::> snapmirror modify -destination-path cluster02://vs1/vol1 -throttle 10240
```

To change the throttle of an active SnapMirror relationship, terminate the existing transfer and restart it to use the new value. SnapMirror restarts the transfer from the last restart checkpoint using the new throttle value, rather than restarting from the beginning.

Note: Starting with clustered Data ONTAP 8.2.1, intracluster throttle is supported, and it works exactly the same way as intercluster throttle.

3.10 Firewall Requirements for Intercluster SnapMirror

Open the following ports on the intercluster network between all source and destination nodes for intercluster replication:
• Port 11104
• Port 11105

Clustered Data ONTAP uses port 11104 to manage intercluster communication sessions. It uses port 11105 to transfer data.

4 Interoperability

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

5 SVM Peering

SVM peering is the act of connecting two SVMs to allow replication to occur between them (starting in clustered Data ONTAP 8.2). Cluster peering must be configured to allow any replication to occur between different clusters. In clustered Data ONTAP 8.1, any SVM could replicate data to any other SVM in the same cluster or any cluster peer. Control of replication security could be maintained at only a clusterwide level. Starting in clustered Data ONTAP 8.2, more granularity in SnapMirror security is provided. Replication permission must be defined by peering SVMs together.

Best Practice

Name an SVM with a unique fully qualified domain name (FQDN): for example, dataVserver.HQ or mirrorVserver.Offsite. SVM peering requires unique SVM names, and using FQDN naming style makes it much easier to make sure of uniqueness.

For additional information regarding SVM peering, see the data protection guide appropriate for the version of clustered Data ONTAP you are running. Here is the URL for Data ONTAP 8 documentation: http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=30092

For example, the information relevant to clustered Data ONTAP 8.3.2 can be found on page 73 here.

6 SnapMirror Data Protection Relationships

Clustered Data ONTAP 8.1 and later versions provides two types of SnapMirror relationships: DP mirrors and load-sharing (LS) mirrors. DP mirrors are discussed in this section. LS mirrors are discussed in a later section.

DP mirrors can be performed as intercluster or intracluster:

• **Intercluster DP mirrors.** Replication between volumes in two different SVMs in different clusters operating in clustered Data ONTAP. They are primarily used for providing DR to another site or location.

• **Intracluster DP mirrors.** Replication between two volumes in different SVMs in the same cluster or between two volumes in the same SVM. They are primarily used for maintaining a local backup copy.

DP mirror relationships have the same characteristics regardless of whether intracluster or intercluster is being replicated. These characteristics include:

• DP mirror relationships are created and managed on the destination cluster.

• DP mirror relationship transfers are triggered by the scheduler in the destination cluster.
• Each DP mirror destination volume is a separate SnapMirror relationship that is performed independently of other DP mirror volumes. However, the same clustered Data ONTAP schedule entry can be used for different DP mirror relationships.

• Destination volumes for both DP- and LS-type mirrors must be created with a volume type (-type option) of DP. The storage administrator cannot change the volume -type property after the volume has been created.

• DP mirror destination volumes are read-only until failover.

• DP mirror destination volumes can be failed over using the SnapMirror break operation, making the destination volume writable. The SnapMirror break must be performed separately for each volume.

• DP mirror destination volumes can be mounted into an SVM namespace while still read-only, but only after the initial transfer is complete.

• An intercluster DP mirror destination volume cannot be mounted in the same namespace as the source volume, because intercluster DP mirror relationships are to a different cluster and therefore to a different SVM, which is a different namespace.

• An intracluster DP mirror destination volume can be mounted in the same namespace as the source volume if both the source and destination volumes exist in the same SVM. However, they cannot be mounted to the same mount point.

• LUNs contained in DP mirror destination volumes can be mapped to igroups and connected to clients. However, the client must be able to support connection to a read-only LUN.

• DP mirror relationships can be managed using the clustered Data ONTAP command line interface (CLI), NetApp OnCommand® System Manager 3.0, and NetApp OnCommand Unified Manager 6.0.

• If an in-progress transfer is interrupted by a network outage or aborted by an administrator, a subsequent restart of that transfer can automatically continue from a saved restart checkpoint.

Clustered Data ONTAP 8.2 and later versions provides an additional SnapMirror relationship: XDP vault. For more information about SnapVault in clustered Data ONTAP 8.2, refer to TR-4183.

6.1 Networks Used for SnapMirror Data Protection Relationships

Intercluster and intracluster DP SnapMirror relationships are different based on the network that is used for sending data. Intercluster DP SnapMirror relationships use the intercluster network defined by intercluster LIFs. Figure 11 illustrates an intercluster network for SnapMirror.
Intracluster DP mirror relationships use the cluster interconnect, which is the private connection used for communication between nodes in the same cluster. Figure 12 illustrates a cluster interconnect for intracluster SnapMirror.

Figure 12) Cluster interconnect for intracluster SnapMirror.
6.2 SnapMirror Data Protection Relationships

After the cluster peer relationship and SVM peer relationship have been successfully created between the two clusters, create the intercluster SnapMirror relationships. A peer relationship is not required to mirror data between two SVMs in the same cluster or between two volumes in the same SVM.

Both the source and destination SVMs must have the same language type setting to be able to replicate between them (in clustered Data ONTAP 8.1). Starting with clustered Data ONTAP 8.1.1, source and destination volumes must have the same language type. An SVM language type cannot be changed after it has been created.

Intercluster SnapMirror relationships are primarily used to provide DR capability in another site or location. If all necessary volumes have been replicated to a DR site with SnapMirror, then a recovery can be performed so that operations can be restored from the DR site.

The creation of SnapMirror relationships in clustered Data ONTAP does not depend on SVM host name to IP address resolution. Whereas the cluster names are resolved through the peer relationship, the SVM names are internally resolved through the clusters. The host names of the source and destination SVM and cluster are used to create SnapMirror relationships in clustered Data ONTAP. It is not necessary to use the IP address of a LIF.

Intercluster SnapMirror Requirements

Complete the following requirements before creating an intercluster SnapMirror relationship:

- Configure the source and destination nodes for intercluster networking.
- Configure the source and destination clusters in a peer relationship.
- Create a destination SVM that has the same language type as the source SVM. Volumes cannot exist in clustered Data ONTAP without an SVM (in clustered Data ONTAP 8.1).
- Both the source and destination SVM can have different language types, but the source and destination volumes must have the same language type. The SVM language type can be set only at the time of SVM creation (starting in clustered Data ONTAP 8.1.1).
- Configure the source and destination SVM in a peer relationship.
- Create a destination volume with a type of DP, with a size equal to or greater than that of the source volume.
- Assign a schedule to the SnapMirror relationship in the destination cluster to perform periodic updates. If any of the existing schedules are not adequate, a new schedule entry must be created.

SVM Fan-Out and Fan-In

It is possible to fan out or fan in volumes between different SVMs. For example, multiple different volumes from a single SVM in the source cluster might be replicated with each volume replicating into a different SVM in the destination cluster, referred to as fan-out. Alternatively, multiple different volumes might also be replicated, each existing in a different SVM in the source cluster, to a single SVM in the destination cluster, referred to as fan-in.

Best Practice

When replicating to provide DR capabilities, mirror all required volumes from a given SVM in the source cluster to a particular matching SVM in the destination cluster. Design considerations that determine that a given set of volumes should reside in the same SVM should also apply to keeping those same volumes in a like SVM at a DR site. In order for different volumes to be accessible in the same namespace, they must exist in the same SVM (an SVM is a namespace).
Volume Fan-Out and Fan-In

For SnapMirror DP relationships, a single NetApp FlexVol volume can be replicated to up to eight different destination volumes. Each destination volume can exist in a different SVM, or all can exist in the same SVM. This approach is referred to as volume fan-out. Volume fan-in, which is replication of multiple different volumes into the same destination volume, is not possible.

Cascade Relationships or Multihop Replication

Starting in clustered Data ONTAP 8.2, SnapMirror relationships can be cascaded. The cascade relationships could be a mix of SnapMirror, SnapVault, or unified replication relationships. From ONTAP 9.0, a cascade of unified replication and/or SnapVault (XDP to XDP) is also supported.

Cascading is defined as replicating from established replicas. Suppose there are three storage systems: A, B, and C. Replicating from A to B and from B to C is considered a cascade configuration.

An example cascade configuration with two hops is shown in Figure 13.

Figure 13) Cascaded volume replication using SnapMirror.

The function of this deployment is to make a uniform set of data available on a read-only basis to users from various locations throughout a network and to allow updating that data uniformly at regular intervals.

Snapshot copy behaviors:

1. SnapMirror creates a soft lock on the Snapshot copy of the source volume (snapmirror tag).
2. The destination system carries an extra Snapshot copy.

Dual-Hop Volume SnapMirror

This configuration involves volume SnapMirror replication among three clusters:

vs1_src:vol1 > vs1_dest:vol1 > vs1_backup:vol1

Note: In the preceding configuration, vs1_src:vol1 to vs1_dest:vol1 and vs1_dest:vol1 to vs1_backup:vol1 transfers can occur at the same time.
Table 1) Snapshot copy propagation for dual-hop volume SnapMirror.

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Snapshot Copies on cluster01</th>
<th>Snapshot Copies on cluster02</th>
<th>Snapshot Copies on cluster03</th>
</tr>
</thead>
</table>
| 1) After volume initialization on cluster02 | hourly.2013-02-26_1505 snapmirror.filev1 | hourly.2013-02-26_1505 snapmirror.filev1 | |}
| 2) Volume SnapMirror update on cluster02 | hourly.2013-02-26_1505 snapmirror.filev2 | hourly.2013-02-26_1505 snapmirror.filev1 snapmirror.filev2 | |}
| 4) Volume SnapMirror update on cluster02 | hourly.2013-02-26_1505 snapmirror.filev2 snapmirror.filev3 | hourly.2013-02-26_1505 snapmirror.filev1 snapmirror.filev2 snapmirror.filev3 | hourly.2013-02-26_1505 snapmirror.filev1 snapmirror.filev2 |

Snapshot copy behaviors to note:

- There is an extra Snapshot copy on cluster02 (destination) after the first SnapMirror update (step 2).
- Cluster03 also has the same number of Snapshot copies as cluster02 after step 3 because there is a volume SnapMirror relationship between cluster02 and cluster03 systems.
- A new soft lock exists on cluster02 after step 3 because cluster02 is now the volume SnapMirror source for cluster03.
- After step 4, the source cluster, cluster01, contains two SnapMirror Snapshot copies. This arrangement is because the Snapshot copy `snapmirror.filev2` is locked by cluster02 because it is required to continue to perform SnapMirror updates with cluster03. This Snapshot copy on cluster01 system is also used to perform SnapMirror resync with the cluster03 system in case the cluster02 system meets disaster.
- After an update is performed on cluster03 (step 5), the soft lock now exists on the latest SnapMirror Snapshot copy, `snapmirror.filev3`, because this copy is the new baseline SnapMirror Snapshot copy between cluster02 and cluster03 systems.

Seeding Intercluster SnapMirror Relationships

The term seeding refers to the initial transfer of data for a newly created SnapMirror relationship.

When a new SnapMirror relationship is created using the `snapmirror create` command, an initial transfer is not automatically performed. The `create` command simply establishes the relationship and the metadata that defines it. Follow the `snapmirror create` command with the `snapmirror initialize` command to perform the initial transfer. Alternatively, use the `snapmirror initialize` command alone to perform the initial transfer as soon as the relationship is created. If the SnapMirror relationship does not exist, then the `initialize` command creates the relationship and performs the initial transfer.

NetApp OnCommand System Manager 3.0 provides the option of initializing a relationship using the SnapMirror relationship create wizard. Managing SnapMirror with System Manager is described later in this document.
6.3 Scheduling SnapMirror Updates

Clustered Data ONTAP has a built-in scheduling engine similar to cron. Periodic replication updates in clustered Data ONTAP can be scheduled by assigning a schedule to a SnapMirror relationship in the destination cluster. Create a schedule through the command line using the job schedule cron create command. This example demonstrates the creation of a schedule called Hourly_SnapMirror that runs at the top of every hour (on the zero minute of every hour).

```
cluster02::> job schedule cron create Hourly_SnapMirror -minute 0
cluster02::> job schedule cron show
Name                Description
----------------- ------------------------------------------
5min               @:00,:05,:10,:15,:20,:25,:30,:35,:40,:45,:50,:55
8hour              @2:15,10:15,18:15
Hourly_SnapMirror  @:00
avUpdateSchedule   @2:00
daily              @0:10
hourly             @:05
weekly             Sun@0:15
```

The schedule can then be applied to a SnapMirror relationship at the time of creation using the –schedule option or to an existing relationship using the snapmirror modify command and the –schedule option. In this example, the Hourly_SnapMirror schedule is applied to an existing relationship.

```
cluster02::> snapmirror modify -destination-path cluster02://vs1/vol1 -schedule Hourly_SnapMirror
```

Schedules can also be managed and applied to SnapMirror relationships using NetApp OnCommand System Manager 3.0.

6.4 Converting a SnapMirror Relationship to a SnapVault Relationship

One scenario in which you would want to convert an existing SnapMirror relationship to a SnapVault relationship: an existing customer using SnapMirror in clustered Data ONTAP 8.1 wants to use SnapVault in clustered Data ONTAP 8.2 for longer retention.

Upgrade your source and destination clusters to clustered Data ONTAP 8.2. Your existing SnapMirror relationships continue to remain cluster scope and behave as they did in clustered Data ONTAP 8.1. They do not benefit from the scalability improvements unless they are deleted and recreated. However, both clustered Data ONTAP 8.1 and clustered Data ONTAP 8.2 use the block-level engine for mirrors, and it is important to note that no rebaseline is required, only resync.

Figure 14 outlines an example based on the details from the preceding paragraph. Cluster peering and SVM peering have already been set up in this example.
It consists of the following steps:

1. Delete mirror (DR) relationship.
2. Break the mirror destination.
3. Create an XDP (vault) relationship between the same endpoints.
4. Perform resync between the endpoints. This resync converts a DR destination to a vault destination without having to do a rebaseline. (Metadata rebuild will take 10-12 minutes per TB of source data).

Create a Volume on the Primary Cluster

```
Primary::> vol create -vserver vs1P -volume voll_vs1P -aggregate aggr1_Primary_01
          -size 10GB (volume create)
[Job 81] Job succeeded: Successful
```

Create a DP Volume on the Remote Cluster

```
Remote::> vol create -vserver vs1R -volume voll_vs1R -aggregate aggr1_Remote_01
          -size 10GB -type DP (volume create)
[Job 81] Job succeeded: Successful
```

Create a SnapMirror Relationship Between the Volumes on the Primary and the Remote Clusters

```
Remote::> snapmirror create -source-path vs1P:voll_vs1P -destination-path vs1R:voll_vs1R -type DP
          -schedule daily
Operation succeeded: snapmirror create the relationship with destination vs1R:voll_vs1R.
```

```
Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source Path</th>
<th>Destination Type</th>
<th>Mirror State</th>
<th>Relationship</th>
<th>Total Progress</th>
<th>Last Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:voll_vs1P</td>
<td>DP</td>
<td>vs1R:voll_vs1R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Initialize the SnapMirror Relationship

Remote::> snapmirror initialize -destination-path vs1R:vol1_vs1R
Operation is queued: snapmirror initialize of destination vs1R:vol1_vs1R.

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last</th>
<th>Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td>vs1R:vol1_vs1R</td>
<td>Snapmirrored</td>
<td>Idle</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Conversion of SnapMirror to SnapVault

SnapMirror Delete

Remote::> snapmirror delete -destination-path vs1R:vol1_vs1R
Operation succeeded: snapmirror delete the relationship with destination vs1R:vol1_vs1R.

SnapMirror Break

Remote::> snapmirror break -destination-path vs1R:vol1_vs1R
[Job 128] Job succeeded: SnapMirror Break Succeeded

SnapVault Create

Remote::> snapmirror create -source-path vs1P:vol1_vs1P -destination-path vs1R:vol1_vs1R -type XDP
Operation succeeded: snapmirror create the relationship with destination vs1R:vol1_vs1R.

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last</th>
<th>Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td>vs1R:vol1_vs1R</td>
<td>Broken-off</td>
<td>Idle</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

SnapMirror Resync

Remote::> snapmirror resync -destination-path vs1R:vol1_vs1R

Warning: All data newer than Snapshot copy
snapmirror.3fd9730b-8192-11e2-9caa-123478563412_2147484699.2013-02-28_10732 on volume vs1r:vol1_vs1r will be deleted. Verify there is no XDP relationship whose source volume is "vs1R:vol1_vs1R". If such a relationship exists then you are creating an unsupported XDP to XDP cascade.
Do you want to continue? [y|n]: y
[Job 133] Job succeeded: SnapMirror Resync Transfer Queued

Remote::> snapmirror show
After completing the preceding steps, adjust the schedules and policies accordingly to keep desired Snapshot copies on the vault destination. Also, you cannot make a SnapVault destination volume read/write for use as a DR volume.

7 SnapMirror Load-Sharing Mirror Relationships

SnapMirror LS mirrors increase performance and availability for NAS clients by distributing an SVM namespace root volume to other nodes in the same cluster. They also distribute data volumes to other nodes in the cluster to improve performance for large read-only workloads. Starting with ONTAP 9.1, LS mirrors have been deprecated for data volumes and are only supported with SVM root volumes.

Note: SnapMirror LS mirrors are only capable of supporting NAS (CIFS/NFSv3). LS mirrors do not support NFSv4 clients or SAN client protocol connections (FC, FCoE, or iSCSI). However, you can use NFSv4 and LS mirrors in the same environment. NFSv4 never uses an LS mirror; instead, it always uses the source volume.

For additional information regarding LS mirrors, see the “Creating and Initializing Load-Sharing Mirror Relationships” guide in the ONTAP 9 Documentation Center.

8 SnapMirror Unified Replication

While backups provide operational recovery for data, DR is necessary to protect the critical data against possible catastrophic natural events or human-caused disasters. Backup means creating a copy of the necessary data. DR means the copy of the data must be sent to another location. Traditionally, backup and DR were achieved separately by using two different sets of data copies. Replication essentially serves the purpose of vaulting the data or providing DR capabilities. However, with unified replication, backup and DR can be achieved together.

SnapMirror unified replication refers to the use of SnapMirror with the same (unified) logical replication engine as in NetApp SnapVault technology. This unified relationship type is designated extended data protection (XDP) and provides single baseline functionality at the volume level, drastically reducing storage and network bandwidth, which translates immediately into cost savings.

The relationships are created with type XDP, policy type mirror-vault, and policy MirrorAndVault (predefined). The policy can always be modified to include custom rules for backing up specific application-consistent Snapshot copies.

Functionality available in clustered Data ONTAP 8.3 and later versions removes the limitation of the destination controller needing to have a clustered Data ONTAP major version number equal to or higher than the major version of the source controller, allowing customers to have nondisruptive upgrades. In addition, the functionality reduces the number of secondary Snapshot copies needed on the destination.

8.1 Default Policies

Three additional SnapMirror policies are defined for replication starting in clustered Data ONTAP 8.3. They are:

- **MirrorLatest.** A Snapshot copy of the active file system is created and transferred from the source to the destination.
- **MirrorAllSnapshots.** This option is similar to default SnapMirror. All source Snapshot copies, including the active file system SnapMirror Snapshot copy, created are transferred from the source to the destination.

- **MirrorAndVault.** This option gives the capability of disaster recovery and backup in a single volume. Retention is set similar to vault policies with SnapMirror labels and their corresponding keep count. In addition to transferring Snapshot copies' matching labels, the active file system SnapMirror Snapshot copy is also created and transferred.

### 8.2 Use Cases

In today's always-on global business environment, data is the key to everything. It needs to be protected, and there needs to be a mechanism to quickly recover that data if damaging natural or human-made events occur. Customers are looking to maximize their investments and get the most out of their IT infrastructure. Effective data protection strategy is vital to prevent operations from being brought to a standstill, which could result in lost productivity and revenue and damage to a customer's reputation.

While NetApp does provide data protection on the primary system using Snapshot technology as well as RAID, SnapMirror and MetroCluster™ help to meet the secondary protection needs.

The need for continuous availability for mission-critical applications can be met by solutions such as MetroCluster. However, not all applications require such a high level of continuous availability. Moreover, customers prefer to reuse their secondary facility for business intelligence or development and testing, thus turning their backup and DR solution into a business accelerator.

Unified replication is able to deliver application-consistent as well as crash-consistent recovery across local and remote sites. It replicates only what is needed, so its impact on the system is minimal. It replicates only the changed blocks, maintaining storage efficiencies on the primary system and further reducing network requirements with native network compression. SnapMirror maintains application consistency when used with SnapManager®, SnapProtect®, or other application-aware backup management solutions.

Depending on the need of the customer, required RPO settings can be configured using the unified replication RPO calculator.

With unified replication, the total cost of ownership can be reduced. The secondary storage investment can be justified because the secondary site can be put for active business use. Using FlexClone technology, near-instantaneous and space-efficient copies of secondary data can be created and used without any negative effect on the production system.

Unified replication also allows data distribution across different locations. It allows fan-out (one-to-many) capability so that the same data can be distributed globally. Cascades are also supported, which allows data replication for disaster recovery from one FAS/All Flash FAS (AFF) to another FAS/AFF and the ability to further back it up to another third-party system (Altavault®, StorageGRID®, or cloud). It can also be leveraged to create storage tiers between SATA and SSD or different backup tiers wherein a secondary may store weekly data and a tertiary may store monthly data.

Unified replication is designed to meet recovery requirements in a timely manner. Frequent replication transfers can be enabled by using schedule intervals of as low as 5 minutes. Selective Snapshot copies can be transferred from the primary to secondary using the policy-driven mechanism. Different types of Snapshot copies can be identified based on their SnapMirror label, and they can have different retention settings. Additional Snapshot copies can be created on secondary as well.

The different policies just listed meet different needs.

**MirrorLatest** helps to take crash-consistent Snapshot copies on the primary and replicates those. Its policy type is async-mirror and has a rule sm_created. At every SnapMirror schedule, a Snapshot copy whose name is prefixed with ‘snapmirror’ is created on the active file system of the primary, and the same is transferred to the secondary. This approach provides a crash-consistent recovery point.
<table>
<thead>
<tr>
<th>Vserver Policy</th>
<th>Policy Number</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>Vserver 1</td>
<td>MirrorLatest</td>
<td>async-mirror</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MirrorLatest**

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Snapshot Copies on cluster01</th>
<th>Snapshot Copies on cluster02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) After volume initialization on cluster02</td>
<td>hourly.2017-01-07_1505 snapmirror.filev1</td>
<td>snapmirror.filev1</td>
</tr>
<tr>
<td>2) Volume SnapMirror update on cluster02</td>
<td>hourly.2017-01-07_1505 <em>snapmirror.filev1</em> hourly.2017-01-07_1605 snapmirror.filev2 hourly.2017-01-07_1705 snapmirror.filev2</td>
<td>snapmirror.filev1 snapmirror.filev2</td>
</tr>
</tbody>
</table>

**MirrorAllSnapshots** is similar to the DP-style SnapMirror relationship wherein all Snapshot copies are transferred from the primary to the secondary. Its policy type is async-mirror and has rules sm_created (to allow crash-consistent Snapshot copy creation on the primary) and all_source_snapshots (to specify that all Snapshot copies need to be transferred).

<table>
<thead>
<tr>
<th>Vserver Policy</th>
<th>Policy Number</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>Vserver 1</td>
<td>MirrorAllSnapshots</td>
<td>async-mirror</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vserver Policy</th>
<th>Policy Number</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>Vserver 1</td>
<td>MirrorAllSnapshots</td>
<td>async-mirror</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 16) Snapshot copy propagation for SnapMirror with policy MirrorAllSnapshots.

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Snapshot Copies on cluster01</th>
<th>Snapshot Copies on cluster02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) After volume initialization on cluster02</td>
<td>hourly.2017-01-07_1505 snapmirror.filev1</td>
<td>hourly.2017-01-07_1505 snapmirror.filev1</td>
</tr>
</tbody>
</table>

MirrorAndVault is the default unified replication policy. It is a unification of SnapMirror and SnapVault policies to allow mirroring the latest active file system (crash-consistent recovery point) as well backup daily and weekly Snapshot copies (application-consistent recovery points). Its policy type is mirror-vault and has rule sm_created as well as daily and weekly. This policy can be modified to add/remove the application-specific rules (snapmirror-labels), or a new variant of this policy can always be created.

<table>
<thead>
<tr>
<th>Vserver Policy</th>
<th>Policy Number</th>
<th>Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>vs-1</td>
<td>MirrorAndVault</td>
<td>mirror-vault</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 17) Snapshot copy propagation for SnapMirror with policy MirrorAndVault.

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Snapshot Copies on cluster01</th>
<th>Snapshot Copies on cluster02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) After volume initialization on cluster02</td>
<td>daily.2017-01-01_0005 weekly.2017-01-01_0005 snapmirror.filev1</td>
<td>snapmirror.filev1</td>
</tr>
</tbody>
</table>
## 8.3 Technology Implications

Currently in Data ONTAP there are two ways to create replicas:

- **Block-based replication.** Refers to replication of the on-disk format of the source volume expressed in terms of blocks of a stable point-in-time Snapshot copy of a file system to a destination volume (DP relationships).

- **Logical replication.** Refers to replication of the data in a source volume expressed in terms of logical units of file data, inode data, and directory entries, captured in a stable point-in-time Snapshot copy of the source file system, independent of the on-disk layout of both the source and destination file systems (XDP relationships).

With unified replication, SnapMirror and SnapVault use the same (storage-efficient) logical replication engine. On the secondary system, there is a single baseline copy of the volume being replicated, and all Snapshot copies are associated with this volume regardless of how they were created or replicated. Therefore, data recovery to the source system can be made from any Snapshot copy on the secondary system.

Storage efficiency features such as block sharing and compression allow a volume effectively to hold far more data than the space actually used. Unless this efficiency is preserved during replication, the replica might inflate to an intolerably large size (not to mention the time needed to transfer it). In extreme but plausible cases, it might not be possible to create a replica at all. Storage efficiency is an important part of this logical replication engine. An inability to create efficient replicas largely nullifies its value.

Additional storage efficiency can always be applied on the secondary. This capability is independent of the primary storage settings.

In addition to asymmetric storage efficiency support, the logical replication engine also supports Data ONTAP version independence (where the destination version can be higher or lower than the source) and Snapshot copy independence (where the destination can support a greater number of Snapshot copies.)

### Timeline

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Snapshot Copies on cluster01</th>
<th>Snapshot Copies on cluster02</th>
</tr>
</thead>
</table>
than the source). This approach allows for reduced downtime because the controllers on either side can be upgraded at any time. Also, other issues of managing complex replication topologies are reduced.

The performance characteristics are also similar to those of the original block replication engine used by SnapMirror (DP relationships).

In comparison to traditional backup software, the replication engine only transfers the differences between two Snapshot copies from the primary to the secondary. This incremental-only transfer leads to a lot of savings in terms of storage and network bandwidth.

Overall, this unified replication (SnapMirror) provides powerful data management capabilities for virtualization, protecting critical data while providing the flexibility to move data between locations and storage tiers, including cloud service providers. SnapMirror technology can directly replicate data between FAS and AltaVault or SolidFire®. It can also be used to replicate data to StorageGRID or other cloud providers using ONTAP Cloud.

8.4 Configuring SnapMirror: Unified Replication

The following is an example of how unified replication can be configured with the MirrorAllSnapshots policy from the CLI:

```bash
cluster02:> snapmirror create -source-path svmA:srcvolA -destination-path svmB:dstvolB -type XDP -policy MirrorAllSnapshots
```

8.5 Converting Default SnapMirror to SnapMirror: Unified Replication

Consider the following scenario:

An existing customer using SnapMirror in clustered Data ONTAP 8.2 wants to use SnapMirror unified replication in clustered Data ONTAP 8.3 to use a single destination volume for disaster recovery and backup.

Upgrade your source and destination clusters to clustered Data ONTAP 8.3. Cluster peering and SVM peering have already been done.

The following are the steps for the conversion:

1. Delete mirror (DR) relationship.
2. Break the mirror destination.
3. Create an XDP relationship between the same endpoints with one of the default SnapMirror unified replication policies.
4. Perform resync between the endpoints. This resync converts the relationship to a SnapMirror unified replication configuration without having to do a rebaseline.

Create a Volume on the Primary Cluster

```bash
Primary::> vol create -vserver vs1P -volume vol1_vs1P -aggregate aggr1_Primary_01 -size 10GB (volume create)
[Job 81] Job succeeded: Successful
```

Create a DP Volume on the Remote Cluster

```bash
Remote::> vol create -vserver vs1R -volume vol1_vs1R -aggregate aggr1_Remote_01 -size 10GB -type DP (volume create)
[Job 81] Job succeeded: Successful
```
Create a SnapMirror Relationship Between the Volumes on the Primary and the Remote Clusters

Remote::> snapmirror create -source-path vs1P:vol1_vs1P -destination-path vs1R:vol1_vs1R -type DP -schedule daily
Operation succeeded: snapmirror create the relationship with destination vs1R:vol1_vs1R.

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last</th>
<th>Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vs1R:vol1_vs1R</td>
<td>Uninitialized</td>
<td>Idle</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 entries were displayed.

Initialize the SnapMirror Relationship

Remote::> snapmirror initialize -destination-path vs1R:vol1_vs1R
Operation is queued: snapmirror initialize of destination vs1R:vol1_vs1R.

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last</th>
<th>Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vs1R:vol1_vs1R</td>
<td>Snapmirrored</td>
<td>Idle</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 entries were displayed.

Conversion of Default SnapMirror to SnapMirror Unified Replication

SnapMirror Delete

Remote::> snapmirror delete -destination-path vs1R:vol1_vs1R
Operation succeeded: snapmirror delete the relationship with destination vs1R:vol1_vs1R.

SnapMirror Break

Remote::> snapmirror break -destination-path vs1R:vol1_vs1R
[Job 128] Job succeeded: SnapMirror Break Succeeded

SnapVault Create

Remote::> snapmirror create -source-path vs1P:vol1_vs1P -destination-path vs1R:vol1_vs1R -type XDP -policy MirrorLatest
Operation succeeded: snapmirror create the relationship with destination vs1R:vol1_vs1R.

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last</th>
<th>Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vs1R:vol1_vs1R</td>
<td>Broken-off</td>
<td>Idle</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SnapMirror Resync

Remote::> snapmirror resync -destination-path vs1R:vol1_vs1R

Warning: All data newer than Snapshot copy
snapmirror.3fd9730b-8192-11e2-9ca9-123478563412_2147484699.2013-02-28_10372 on volume vs1r:vol1_vs1r will be deleted.
Verify there is no XDP relationship whose source volume is
"vs1R:vol1_vs1R". If such a relationship exists then you are creating
an unsupported XDP to XDP cascade.
Do you want to continue? {y|n}: y
[Job 133] Job succeeded: SnapMirror Resync Transfer Queued

Remote::> snapmirror show

<table>
<thead>
<tr>
<th>Source Path</th>
<th>Destination Path</th>
<th>Mirror</th>
<th>Relationship</th>
<th>Total</th>
<th>Last Progress</th>
<th>Healthy</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1P:vol1_vs1P</td>
<td>XDP</td>
<td>vs1R:vol1_vs1R</td>
<td>Snapmirrored</td>
<td>-</td>
<td>true</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

After completing the preceding steps, you can adjust the schedules and policies accordingly to keep desired Snapshot copies on the destination.

**Note:** The SnapMirror unified replication relationship cannot be converted back to a default SnapMirror relationship unless a rebaseline is created to a new destination volume.

For additional information, refer to the unified replication FAQ on the Field Portal.

8.6 Secondary Snapshot Management: Unified Replication

Current SnapVault or MirrorAndVault implementation does not delete any Snapshot copies from primary after transfer. It is made up of two modules: a Snapshot policy attached to the primary volume and a SnapMirror policy configuration attached to the secondary volume. The Snapshot policy controls creation and retention of backup copies on the primary. Alternatively, the admin can use an external application such as NetApp SnapManager or even a homegrown script to create and manage backup copies on the primary. The SnapVault configuration defines a schedule/frequency for transferring the backup copies from primary to secondary. In addition, it defines retention time for the backup copies on the secondary. Because the minimum retention count for any Snapshot copy is 1, there is always one backup Snapshot copy of each type on the primary. Thus a Snapshot copy might remain on the primary for months or even years after it's been transferred to the secondary. This approach locks up expensive space on the primary. The secondary Snapshot management provides ability to create some Snapshot copies on the secondary itself instead of creating them on primary and then transferring them over the network. Thus, additional backup Snapshot copies (for example, weekly and monthly) can be created directly on the secondary while those on daily are being transferred from the primary.

This is done by extending the rules in the SnapMirror policy. Before ONTAP 9.0, each rule identified a type of backup Snapshot copy to be transferred from the primary. With this feature each rule specifies if the associated Snapshot copy is to be transferred from the primary or just created locally.

**User Interface**

Existing schema of a policy rule can be extended like the following:

<table>
<thead>
<tr>
<th>Rules:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SnapMirror Label</td>
</tr>
<tr>
<td>daily</td>
</tr>
</tbody>
</table>
SnapMirror Policy UI

```
<cluster>::* snapmirror policy add-rule ?
  -vserver <vserver name> Vserver Name
  [-policy <snapmirror policy>] SnapMirror Policy Name
  [-snapmirror-label <text>] Snapshot Copy Label
  [-keep <text>] Snapshot Copy Retention Count
  [[-preserve [true|false]]] Snapshot Copy Preserve Enabled
  [ -warn <integer> ] Warning Threshold Count
  [ -schedule <text> ] Schedule name (associated with cron)
  [ -prefix <text> ] Snapshot copy name prefix associated with the schedule
```

Usage Example

The user wants to maintain a limited amount of data on the primary but wants to back up all this data for a long amount of time (say, 7 years). Suppose the user maintains 3 dailies and 2 weekly Snapshot copies on the primary, but on the secondary wants to maintain the most recent 14 daily, 4 weekly, 12 monthly, and 7 yearly Snapshot copies.

Create a snapshot policy which will create a daily snapshot and retain it up to 3 days on primary, and create a weekly snapshot and retain it up to 2 weeks. Create a Vault relationship between the primary and secondary and attach a midnight transfer schedule to it.

Create cron jobs: monthly (@1st of every month), yearly (@1st of Jan) Create a SnapMirror policy to be attached to the relationship with the following rules:
A rule with snapmirror-label daily, having keep as 14 and action as none.
A rule with snapmirror-label weekly, having keep as 12 and action as none.
A rule with snapmirror-label monthly, having keep as 12, action=create and schedule as monthly@1. A prefix can be added to have a specific snapshot name format.
A rule with snapmirror-label yearly, having keep as 7, action=create and schedule as yearly. Again, prefix can be added if required.
<table>
<thead>
<tr>
<th>Schedule</th>
<th>Count</th>
<th>Prefix</th>
<th>SnapMirror Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>daily</td>
<td>3</td>
<td>daily</td>
<td>daily</td>
</tr>
<tr>
<td>weekly</td>
<td>2</td>
<td>weekly</td>
<td>weekly</td>
</tr>
</tbody>
</table>

```
arpan4cluster::*> cron show
(job schedule cron show)
Name  Description
--------------------
5min  0:00,0:05,0:10,0:15,0:20,0:25,0:30,0:35,0:40,0:45,0:50,0:55
8hour @2:15,10:15,18:15
daily @0:10
hourly @:05
monthly@1 Jan-Dec 180:00
weekly Sun@0:15
yearly Jan 180:00
7 entries were displayed.
```

```
arpan4cluster::*> snapmirror policy create -vserver arpan4cluster -policy SecSnapMirrorPolicy
arpan4cluster::*> snapmirror policy add-rule -vserver arpan4cluster SecSnapMirrorPolicy -snapmirror-label daily -keep 14
arpan4cluster::*> snapmirror policy add-rule -vserver arpan4cluster SecSnapMirrorPolicy -snapmirror-label weekly -keep 4
arpan4cluster::*> snapmirror policy add-rule -vserver arpan4cluster SecSnapMirrorPolicy -snapmirror-label daily -keep 12 -schedule monthly@1 -prefix monthly
arpan4cluster::*> snapmirror policy add-rule -vserver arpan4cluster SecSnapMirrorPolicy -snapmirror-label daily -keep 7 -schedule yearly -prefix yearly
```

```
arpan4cluster::*> snapmirror policy show SecSnapMirrorPolicy
Vserver Policy       Policy Number Transfer
Name   Name    Type Of Rules Tries Priority Comment
------- ------------------ ---- ------ ------ ---- --------
arpan4cluster SecSnapMirrorPolicy vault 3 8 normal

SnapMirror Label: daily Keep: 14
weekly 4
monthly 12
yearly 7
Total Keep: 37
```

### 8.7 Configuring Cascades: Unified Replication

SnapMirror allows volumes to be in multiple relationships, serving as source for multiple destinations, while receiving updates from some other source. Moreover, SnapMirror allows DR and vault relationships to be used simultaneously. The key capabilities that need to be preserved are:

- **Ability for a volume to work simultaneously on multiple relationships:** that is, not imposing any constraint on when transfers can take place. This capability is implemented through the notion of soft locks on Snapshot copies.
- **Ability to reconfigure relationships if one of the participants becomes unavailable,** without requiring a new baseline transfer. This capability is implemented by making sure that there is a common Snapshot copy across all the volumes in a relationship graph. This approach might require the transmission of extra Snapshot copies above and beyond those specified by the policies for a vault.
- **Ability to troubleshoot this distributed graph of incremental changes and Snapshot copies.**

With ONTAP 9.0, vault-to-vault cascade configurations are also supported. That is, XDP-to-XDP type relationship cascades are supported.

Unified replication to unified replication cascading:
A -- XDP > B -- XDP > C

Both A > B and B > C relationships could be attached to a SnapMirror policy of type async-mirror or mirror-vault or just vault. The A > B relationship always creates a new Snapshot copy on A at the time of an update. Depending on the rules added in the policy, the relationship might replicate just the newly created Snapshot copy or all Snapshot copies on the primary volume that are newer than the latest common Snapshot copy.

The B > C relationship does not create a new Snapshot copy on B. Instead it replicates the latest or all Snapshot copies on B to C, depending on the rules in the associated SnapMirror policy.

If the middle volume B is lost, the A > C relationship can be established using SnapMirror resync. The procedure to restore the original topology after B comes back is to utilize a SnapMirror resync in the opposite direction (C > B) or in the forward direction like earlier (A > B).

Cascade support allows:

1. A backup admin can offload certain Snapshot copies to tertiary and thus retain more than the actual number of Snapshot copies supported by a single volume (255 currently).
2. Tiering multiple backup copies. More frequent Snapshot copies (say, daily and weekly) can be retained on secondary (B), whereas the monthly and yearly Snapshot copies can be retained on tertiary (C).
3. Data can be moved from SSDs to SATA disk drives.
4. Data can be distributed globally across locations without any hindrance to ongoing operations. It can be further moved to the cloud as archival storage.

9 Storage Virtual Machine Disaster Recovery

9.1 Overview

- Simple predefined steps to failover
- Ease of management with automation
- Assured protection for SVM data
• Proven SnapMirror engine inside
• Protect SVM namespace, not just volumes
• Automated setup and provisioning
• Automated change management
• Familiar SnapMirror commands on CLI
• GUI-based management using Workflow Automation

9.2 Options

When setting up SVM DR, there are really only two options: identity-preserve=true and identity-preserve=false. For both identity-preserve=true and identity-preserve=false, all volumes/data is replicated. The differences between the two options are in the configuration data that is replicated.

Figure 18) Identity-preserve=true.
9.3 Requirements

See the appropriate express guide in the “Disaster Recovery and Backup” section.

9.4 Use Cases

With only two options for SVM DR, there are only a handful of use cases. Overall SVM DR is used when the entire SVM needs to be replicated or moved to a destination cluster. When you use identity preserve true or false determines how much of the configuration data is sent to the destination. It determines how the destination SVM operates while the relationship is in place.

When using identity preserve true, we maintain the CIFS server identity as well as the network configuration. Because of this fact, the destination SVM is offline until the SnapMirror relationship is broken and the source SVM is offline. Here are a few use cases for this option:

- Source and destination SVM remain in the same layer 2 network.
- Source and destination SVM are in different layer 2 networks but have access to the same Active Directory structure.
- User is moving an SVM from one cluster to a different cluster and maintaining the CIFS server configuration and possibly network configuration.

The first of these use cases is for customers who have two clusters in the same layer 2 network. This configuration could be in the same data center or an extended layer 2 network across data centers. The cutover from source to destination cluster does not require any additional SVM configuration changes to bring the SVM online.

In the second use case, because we are maintaining the network configuration, but moving the SVM into a different network, a couple of configuration changes need to be made. First, you need to change the IP addresses on the data LIFs on the SVM after the cutover. Second, you have to change the routing table of the SVM itself. Each SVM has a unique routing table that determines the default gateway for the network. In most cases only these two changes are required. If the DNS server that is configured for the SVM is not reachable on the network, then you have to change the DNS settings. This should be the extent of the changes that are required for CIFS environments. For NFS environments, if your NFS clients
also change their IP addresses (think whole site failover), then you need to make sure that your export policies are updated to use the new IP addresses of those hosts.

The third example is more of moving an SVM than using it for SVM DR. For instance, let’s say you have an SVM that is in the cloud and you want to move it back to on your premises. You can use SVM DR to establish a whole SVM relationship between clusters and move the SVM from one cluster to another. After the cutover to the new cluster, you just make the necessary changes to the network/route/dns/exports as needed, delete the SnapMirror relationship, and continue serving data.

There is one primary use case for using identity preserve false. Because we are not maintaining the network configuration, the CIFS server configuration, or any of the export policies, we can have the destination SVM in an active read-only environment.

10 SnapMirror and Data ONTAP Feature Interaction

10.1 SnapMirror and Snapshot Copies
SnapMirror creates a Snapshot copy before it performs a replication update. A SnapMirror Snapshot copy is created on the source volume, and that Snapshot copy is then compared to the previous SnapMirror Snapshot copy that was replicated. All data between the new SnapMirror Snapshot copy and the previous one (including all Snapshot copies on the volume between those and all data in those Snapshot copies) is replicated to the destination volume. After the SnapMirror update is complete, the new SnapMirror Snapshot copy is exported on the destination system.

SnapMirror maintains a history of one SnapMirror Snapshot copy on the source volume and two on the destination volume.

Best Practice

Verify that SnapMirror updates are not scheduled to occur on the source volume at the same time as other Snapshot copies.

Data ONTAP maintains locks on Snapshot copies created by SnapMirror to prevent them from being deleted. These Snapshot copies are required to perform scheduled updates. If the Snapshot copies created by SnapMirror must be deleted, the volumes can still be resynchronized without having to perform a full baseline as long as other common Snapshot copies between the two volumes still exist on the volumes. In this example, a SnapMirror resync is performed on a volume where all Snapshot copies created by SnapMirror were deleted.

Note: The system specifies the name of an hourly Snapshot copy used for the base of the resync.

```
cluster02::> snapmirror resync -source-path cluster01://vs1/vol1 -destination-path cluster02://vs2/vol1
Warning: All data newer than Snapshot copy hourly.2011-12-06_1805 on volume cluster02://vs2/vol1 will be deleted.
Do you want to continue? [y|n]: y
[Job 1364] Job is queued: snapmirror resync to destination cluster02://vs2/vol1.
```

10.2 SnapMirror and Qtrees
Qtrees are special directories that allow the application of file system quotas for NAS. Clustered Data ONTAP allows creation of qtrees, and qtrees can exist in volumes that are replicated with SnapMirror. However, SnapMirror does not allow replication of individual qtrees or qtree-level replication. In addition, nothing otherwise can be done with a qtree in clustered Data ONTAP.
10.3 SnapMirror and FlexClone

A NetApp FlexClone volume is a writable point-in-time clone of a FlexVol volume. A FlexClone volume shares data blocks with the parent volume, storing only new data or changes made to the clone. A FlexClone volume can also be split from its parent to create a new standalone volume.

A SnapMirror relationship can be created using a FlexClone volume as the source. However, a SnapMirror destination volume cannot be a FlexClone volume. Starting in clustered Data ONTAP 8.3, a SnapMirror relationship can be created using a FlexClone volume as the source or destination or both.

FlexClone technology also makes it possible to create a writable volume from a read-only SnapMirror destination without interrupting the SnapMirror replication process or the production operations. Figure illustrates the creation of a FlexClone volume at the SnapMirror destination.

**Best Practice**

SnapMirror replicates Snapshot copy history from source to destination volumes. If a Snapshot copy is removed from the source volume, the next SnapMirror update removes that Snapshot copy from the destination volume. If that Snapshot copy cannot be removed from the destination—for example, if the Snapshot copy is locked because it is the base Snapshot copy of a FlexClone volume—then the SnapMirror update fails. The only way for a SnapMirror update to proceed is to delete the FlexClone volume or split it to remove the Snapshot copy dependency.

To avoid this issue when creating FlexClone volumes on SnapMirror destinations, create the base Snapshot copy required by the FlexClone volume on the source system and then replicate that Snapshot copy to the destination system and use that Snapshot copy as the base for the FlexClone volume, as shown in Figure. Using a Snapshot copy specifically created for the FlexClone volume in this manner prevents the SnapMirror update from failing due to an automatically created Snapshot copy being removed from the source system.

![Figure 15) Creating FlexClone volume at SnapMirror destination.](image)

10.4 SnapMirror and Infinite Volume

SnapMirror works with Infinite Volumes just like with FlexVol volumes except for a few key differences:
• For the namespace (NS) constituent volume, mirroring is restricted to intracluster only. Starting in clustered Data ONTAP 8.2, the NS mirror is automatically created when you create the Infinite Volume. If you use SnapDiff, it automatically creates one NS mirror per node. If you don’t use SnapDiff, then you have only one NS mirror on the Infinite Volume.

• Only intercluster SnapMirror is supported for mirroring the entire Infinite Volume.

The process of creating an Infinite Volume is different compared to a FlexVol volume. Apart from that, the SnapMirror relationship setup is the same as with a FlexVol volume. For more information about Infinite Volumes in clustered Data ONTAP 8.2, refer to TR-4178.

10.5 SnapMirror and NetApp Storage Efficiency

SnapMirror maintains storage efficiency benefits in replicated volumes. If the source volume is deduplicated, the destination volume is in a deduplicated state as well. SnapMirror does not inflate deduplicated data during a transfer. If the source volume is compressed, the destination volume is in a compressed state as well. Replication of compressed volumes does not uncompress the source volume to read data for a transfer. Data is replicated in a compressed state to the destination volume.

It is not possible to have different configurations of storage efficiency enabled between the source and destination volumes. For example, it is not possible to compress or deduplicate the SnapMirror destination volume alone without enabling compression or deduplication on the SnapMirror source volume.

SnapMirror creates a Snapshot copy before performing an update transfer. Any blocks in the Snapshot copy are locked and cannot be deduplicated. Therefore, if maximum space savings from deduplication are required, run the dedupe process before performing SnapMirror updates.

10.6 SnapMirror and Volume Move

The volume move capability allows volumes to be moved nondisruptively between nodes in the cluster. DP mirror source or destination volumes can be moved using the volume move command. The SnapMirror relationship does not have to be reconfigured or modified on the source or destination when a volume move is performed. If a volume that is in an intercluster SnapMirror relationship is moved, the node to which the volume is moved must have an intercluster LIF and be connected to the intercluster network in order to perform future SnapMirror updates.

The effect a volume move has on a SnapMirror relationship depends on whether the source volume or the destination volume is being moved. If a SnapMirror transfer is currently in progress and the SnapMirror source volume is being moved, then both the SnapMirror transfer and the volume move transfer can run simultaneously. However, when the volume move cutover occurs (the moment the clustered Data ONTAP redirects I/O to the new volume), the active SnapMirror transfer is then momentarily interrupted and automatically continues from the source volume’s new location.

Note: In clustered Data ONTAP 8.1, for SnapMirror destination volumes, a SnapMirror transfer and a volume move transfer are mutually exclusive. A SnapMirror destination volume move cannot start while a SnapMirror transfer to that volume is in progress. A SnapMirror update transfer cannot be performed if the SnapMirror destination volume is currently in the process of being migrated with volume move. However, starting in clustered Data ONTAP 8.2, for SnapMirror destination volumes, a SnapMirror transfer and a volume move transfer can run simultaneously, except during volume move cutover, when they are mutually exclusive (brief duration of a few seconds).

For more information about volume move, refer to TR-4075, Data Motion for Volumes for Data ONTAP 8.2 and 8.3.

10.7 SnapMirror for Disk Shelf Failure Protection

If you have decided that you want to use SnapMirror to protect against disk shelf failure, you need to be aware of two things:
• You cannot mirror the volumes to be in the same HA pair.
• The configuration does not automatically fail over.

You can mirror the volumes to different nodes in a different HA pair on the same cluster. Mirroring to a different node makes sure that the other volume is always in a different shelf. If you try to mirror to a different shelf on the same node, then the mirror has to be on a different aggregate. However, there is still the risk that an aggregate might have a disk in any shelf. Even if you try to configure otherwise (keeping aggregates on their own shelves), that can change because drives fail and spares get used. This configuration avoids having a single point of failure and provides protection against disk shelf failure. The caveat here is that the configuration does not fail over automatically. You have to manually break the SnapMirror relationship, unmount the clients, remount the clients on the destination volumes, and change the NFS export policies.

10.8 SnapMirror and Volume Autosize

The destination volume must be the same size as or larger than the source volume. SnapMirror updates fail if the destination volume is smaller than the source volume.

Best Practice

Keep the source and destination volumes the same size. However, the destination volume can be slightly larger. The `-filesys-size-fixed` option makes sure that the file system size of a SnapMirror volume remains the same to allow a SnapMirror relationship to be reversed, even if the destination volume size is larger than the source. If this value is not set ahead of time and a destination grows larger than a source, a reverse resync fails.

If the source volume size is automatically increased by the volume autosize feature, or if it is manually increased, then the destination volume size must be increased to match the size of the source volume. Clustered Data ONTAP 8.1 does not automatically increase the size of the destination volume. Use the CLI or System Manager to increase the destination volume. The next SnapMirror update automatically replicates the value of the file system size to the destination volume to match that of the source.

If the autosize feature increases the size of the source volume, to avoid having to manually resize the destination volume, size the destination volume so that it is at least as large as the source volume’s maximum autosize value. To eliminate the need for the additional capacity required to guarantee the larger destination volume, the space guarantee can be disabled on the destination. However, keep in mind that the capacity of the destination system must be properly managed so that there is room for operations that generate data on the destination system.

Starting in Data ONTAP 8.2, when autosize increases the size of the source volume of a SnapMirror relationship, the destination volume also automatically increases in size. This is applicable to only FlexVol volumes and not Infinite Volumes.

10.9 SnapMirror and Network Data Management Protocol

Network Data Management Protocol (NDMP) backups can be performed from SnapMirror source or destination volumes. There are advantages to performing NDMP backups from SnapMirror destination volumes rather than from source volumes. They include:

• SnapMirror transfers can happen quickly and with less impact on the source system than that of NDMP backups. Use NetApp Snapshot copies and perform SnapMirror replication from a primary system as a first stage of backup to significantly shorten or eliminate backup windows. Then perform NDMP backup to tape from the secondary system.

• SnapMirror source volumes are more likely to be moved using volume move capability for performance or capacity reasons. When a volume is moved to a different node, the NDMP backup job must be reconfigured to back up the volume from the new location. If backups are performed from the
SnapMirror destination volume, these volumes are less likely to require a move. Therefore, it is less likely that the NDMP backup jobs need to be reconfigured.

10.10 SnapMirror and Data ONTAP Version Dependencies

Replication for DP or DR is not possible between systems operating in 7-Mode and clustered Data ONTAP:

- Several new replication capabilities have been implemented in SnapMirror in clustered Data ONTAP 8.1 such as block-level replication; support for NetApp storage efficiency; and the ability to replicate between clusters and break, reverse, and resync relationships. The Data ONTAP 8.1 implementation of SnapMirror is not compatible with the Data ONTAP 8.0 implementation. Replication between systems running clustered Data ONTAP 8.0 and 8.1 is not possible. For information about upgrading systems operating in clustered Data ONTAP 8.0 to 8.1, refer to the NetApp Data ONTAP 8.1 Cluster-Mode Upgrade and Revert/Downgrade Guide on the NetApp Support site.

- Clustered Data ONTAP 8.2 introduces SnapVault, supports SnapMirror cascading and SnapMirror to tape for seeding only, and is multitenancy ready with the ability for SVM administrators to manage replication. Additionally, clustered Data ONTAP 8.2 improves scalability (number of concurrent transfers) and increases data transfer speeds.

- The clustered Data ONTAP 8.2 implementation of SnapMirror is compatible with the clustered Data ONTAP 8.1 implementation. Replication between systems running clustered Data ONTAP 8.1 and 8.2 is possible. On an upgrade from clustered Data ONTAP 8.1 to 8.2, existing SnapMirror relationships continue to remain cluster scope. The SnapMirror relationships do not benefit from the scalability improvements unless SVM peering is established and the SnapMirror relationships are deleted and recreated after the source and destination nodes are both upgraded to clustered Data ONTAP 8.2.

- Starting in clustered Data ONTAP 8.2.1, on an upgrade from 8.1 to 8.2.1, the preceding procedure is automated. That is, the SnapMirror relationships are autoconverted to benefit from the scalability improvements after the destination node is upgraded to 8.2.1, source node is upgraded to 8.2 or later, and source and destination volumes are in the same SVM or SVM peering is established between the two SVMs that host the source and destination volumes.

The following replication is supported between different versions of clustered Data ONTAP 8.1 and later:

- Replication is allowed in either direction between minor versions. Clustered Data ONTAP 8.1.x is a minor version. Therefore, replication is supported from 8.1.x to 8.1.y or from 8.1.y to 8.1.x.

- Replication is allowed only from older to newer major versions. Clustered Data ONTAP 8.1 is a major version. Therefore, replication is allowed from 8.1 to a later major release (for example, 8.2). However, replication is not allowed from a later release (for example, 8.2) to an earlier major release (for example, 8.1).

  **Note:** The clustered Data ONTAP 8.0 operating system release family is excluded.

- Unified replication using logical replication available in clustered Data ONTAP 8.3 and later removes the limitation of the destination controller needing to have a clustered Data ONTAP major version number equal to or higher than the major version of the source controller, allowing customers to have nondisruptive upgrades.

Overall the following support for releases shall be allowed:
11 Performance

11.1 SnapMirror and Network Compression

With increasing network bandwidth costs coupled with data growth, customers have to do more with less. As the amount of data to be protected increases, more network bandwidth is needed to maintain the recovery point objective (RPO) or the replication window. Otherwise, replication times increase as the amount of data sent over the network to the DR site increases. Differently put, if you do not want to or cannot increase the network bandwidth, you need to lower the replication frequency that is causing larger RPO values and thus increasing your exposure to larger data loss.

The SnapMirror native network compression feature can cut down on the amount of data replicated over the network. It also offers you more flexibility and choices, as described in the following section.

Maintaining the Same RPO Level

- **Challenge.** Your data replication needs are growing. You need more bandwidth to maintain the same level of RPO.
- **Solution.** By using network compression, it is possible to maintain the same RPO without purchasing additional network bandwidth.

Improve Your RPO Without Buying Additional Bandwidth

- **Challenge.** You are using all of your network bandwidth. However, your customer wants to reduce its exposure to data loss: in other words, to improve its RPO.
- **Solution.** By using network compression, you can improve your RPO without purchasing more network bandwidth.

Use the Network Bandwidth for Other Purposes

- **Challenge.** Your replication is consuming all of your bandwidth. You want to use the network bandwidth for other purposes such as client access or applications without purchasing additional bandwidth.
• **Solution.** By using network compression, it is possible to reduce the bandwidth consumed by SnapMirror without sacrificing RPO, thereby freeing up network bandwidth for other purposes.

**Speeding Up the Initial Transfers**

• **Challenge.** Initial SnapMirror transfers could be large and therefore could take a long time to complete under bandwidth constraints.
• **Solution.** By using network compression, it is possible to speed up the initial SnapMirror transfers.

**What Is SnapMirror Network Compression?**

SnapMirror network compression enables data compression over the network for SnapMirror transfers. It is a native feature that is built into SnapMirror software. SnapMirror network compression is not the same as volume compression. SnapMirror network compression does not compress data at rest. Figure 20 shows a very high-level flow of SnapMirror network compression.

Figure 20) SnapMirror network compression functional diagram.

On the source system, the data blocks that need to be sent to the destination system are handed off to the compression engine, which compresses the data blocks. The compression engine on the source system creates several threads, depending on the number of CPUs available on the storage system. These compression threads help to compress data in a parallel fashion. The compressed blocks are then sent over the network.

On the destination system, the compressed blocks are received over the network and are then decompressed. The destination compression engine also has several threads to decompress the data in a parallel fashion. The decompressed data is reordered and is saved to the disk on the appropriate volume.

In other words, when SnapMirror network compression is enabled, two additional steps are performed: compression processing on the source system before data is sent over the network and decompression processing on the destination system before the data is written to the disk.

**Prerequisites**

SnapMirror network compression is supported from clustered Data ONTAP 8.3 on both source and destination systems.

All platforms that support clustered Data ONTAP 8.3 also support SnapMirror network compression.

**Enabling and Disabling Network Compression**

SnapMirror network compression can be enabled or disabled by the `--is-network-compression-enabled` option in SnapMirror policy. It cannot be enabled for an active transfer. To enable compression
for an existing transfer, you must first abort the transfer, set the `-is-network-compression-enabled` option to true in SnapMirror policy, and then resume the transfer.

### Reporting the Compression Ratio

The SnapMirror network compression ratio is reported in the `snapmirror show -instance` output.

```
cluster::> snapmirror show -destination-path vs3:dst -instance

Source Path: vs1:src_test
Destination Path: vs3:dst
Relationship Type: DP
Relationship Group Type: none
SnapMirror Schedule: -
SnapMirror Policy Type: async-mirror
SnapMirror Policy: DPrequest
Try Limit: -
Throttle (KB/sec): unlimited
Mirror State: Snapmirrored
Relationship Status: Transferring
File Restore File Count: -
File Restore File List: -
Transfer Snapshot: snapmirror.89659724-bd35-11e4-9f11-000c299bf0b8_2147484674.2015-03-02_134417
Snapshot Progress: 0B
Total Progress: 0B
Network Compression Ratio: 2:1
Snapshot Checkpoint: 0B
Newest Snapshot: snapmirror.89659724-bd35-11e4-9f11-000c299bf0b8_2147484674.2015-02-25_134212
    Newest Snapshot Timestamp: 02/25 13:22:08
    Exported Snapshot: snapmirror.89659724-bd35-11e4-9f11-000c299bf0b8_2147484674.2015-02-25_134212
    Exported Snapshot Timestamp: 02/25 13:22:08
    Healthy: true
    Unhealthy Reason: -
    Constituent Relationship: false
    Destination Volume Node: vsim
    Relationship ID: d8b4cbc8-bd36-11e4-9f11-000c299bf0b8
    Current Operation ID: 46da2fc6-c125-11e4-9f1a-000c299bf0b8
    Transfer Type: update
    Transfer Error: -
    Current Throttle: unlimited
    Current Transfer Priority: normal
    Last Transfer Type: initialize
    Last Transfer Error: -
    Last Transfer Size: 240KB
    Last Transfer Network Compression Ratio: 1:1
    Last Transfer Duration: 0:0:3
    Last Transfer From: vs1:src_test
    Last Transfer End Timestamp: 02/25 13:42:15
    Progress Last Updated: 03/02 13:44:17
    Relationship Capability: 8.2 and above
    Lag Time: 120:22:10
    Number of Successful Updates: 0
    Number of Failed Updates: 0
    Number of Successful Resyncs: 0
    Number of Failed Resyncs: 0
    Number of Successful Breaks: 0
    Number of Failed Breaks: 0
    Total Transfer Bytes: 245760
    Total Transfer Time in Seconds: 3
```

### 11.2 SnapMirror Throttling

With ONTAP 9.0, the node-level SnapMirror throttling feature is available. Under node-level or global throttling, all nodes in a cluster perform SnapMirror transfers at fixed maximum bandwidth each for outgoing and incoming transfers. In addition, each SnapMirror relationship has a per transfer throttle.
associated with it. The per transfer throttle is capped at the node-level throttle if it exceeds the global node throttle value. Otherwise, the transfers take place at its specified value. Each node has global throttling for sender side (outgoing) as well as receiver side (incoming) transfers and an option to enable or disable this throttling.

The minimum throttle bandwidth should be 4KBps, and the maximum can be up to 2TBps. A throttle bandwidth of 0 implies that the transfer is not throttled or that bandwidth is unlimited.

A new cluster-wide option to control throttling as follows:

```
cluster::> options replication*
cluster
   replication.throttle.enable       on
   replication.throttle.incoming.max_kbs 4000
   replication.throttle.outgoing.max_kbs 2000
3 entries were displayed.
```

Each entry can be edited individually. The enable option either enables or disables both the outgoing and incoming throttle.

```
cluster::> options replication.throttle.enable off
1 entry was modified.
```

Changing the outgoing and incoming throttle is reflected in the actual transfer only if the enable option is on. The outgoing and incoming throttle values can be changed irrespective of the enable option value.

```
cluster::> options replication.throttle.outgoing.max_kbs 8000
1 entry was modified.
cluster::> options replication.throttle.incoming.max_kbs 5000
1 entry was modified.
```

### 11.3 SnapMirror Sizing Recommendations

SnapMirror sizing details can be found in the [SnapMirror Sizing Guide (TR-4231i)](https://www.netapp.com/dataONTAP/artifacts/dmmanuals/snapmirror-sizing-guide.pdf).

### 11.4 Concurrent Replication Operations

The number of supported simultaneous SnapMirror operations is limited. This limit is per node and varies depending on the platform and version of Data ONTAP. For information about the number of concurrent SnapMirror operations allowed per node, refer to the [NetApp Clustered Data ONTAP Data Protection Management Guide on the NetApp Support site](https://www.netapp.com/support/documentation/index.html) for the appropriate Data ONTAP release.

**Best Practice**

It is also important to understand which operations in clustered Data ONTAP constitute SnapMirror operations. Regularly scheduled SnapMirror updates of SnapMirror DP or LS relationships are SnapMirror operations. However, volume move and volume copy operations also use SnapMirror as the mechanism to move data from one aggregate to another. Therefore, when planning concurrent operations, it is a best practice to consider the frequency of volume move and volume copy operations in the environment.

Clustered Data ONTAP provides a greater level of scalability by allowing expansion of a NetApp cluster beyond two nodes. Each node in the cluster provides CPU and memory resources that are used for replication of volumes owned by that node.
Best Practice

In order to optimize replication, distribute replicated volumes across different nodes in the clusters rather than placing all volumes requiring replication on a single node. This best practice allows all nodes in the cluster to share replication activity.

11.5 Recommended Replication Intervals

SnapMirror updates require establishing a communication session between the source and destination nodes, creating and deleting Snapshot copies, and determining which blocks of data to send to the destination. Therefore, while the Data ONTAP scheduler supports creating schedules that run every minute, NetApp does not recommend performing a SnapMirror update operation every minute. However, SnapMirror update operations in single digits are possible depending upon your environment. See the System Performance Modeler for proper SnapMirror and system sizing guidelines.

11.6 Network Sizing Requirements

A network with the appropriate bandwidth available to transfer the system’s data ingest rate is required to support the desired replication interval. There are limitations on the network characteristics that are supported for intercluster replication.

Network Sizing Requirements for Intercluster Replication

Figure 21) Factors to consider for optimum performance: packet loss (%), latency (ms), and network bandwidth (Mbps).

The intercluster network must be sized appropriately depending on the data change rate and update interval to meet the recovery point objective (RPO) of the solution and individual node performance characteristics. Intercluster SnapMirror is supported across networks that have a minimum bandwidth of 0.5Mbps, a maximum round-trip network latency of 200ms round-trip time (RTT), and a packet loss of 1% (volume covered by the green triangle in Figure 21).
**Best Practice**

It is important that all paths used for intercluster replication have equal performance characteristics. Configuring multipathing in such a way that a node has one intercluster LIF on a slow path and the same node has another intercluster LIF on a fast path adversely affects performance because data is multiplexed across the slow and fast paths simultaneously.

**Network Sizing Requirements for Intracluster Replication**

All intracluster transfers, including intracluster SnapMirror DP mirrors, LS mirrors, and volume move and volume copy operations, use the private cluster interconnect between nodes in the same cluster. The cluster interconnect bandwidth is not configurable.

**12 Troubleshooting Tips**

**12.1 Troubleshooting Cluster Peer Relationships**

1. **Run the `cluster peer show` command to verify the availability of the cluster peer relationship.** This command displays all existing configured cluster peer relationships.

   ```bash
   cluster01::> cluster peer show
   Peer Cluster Name          Cluster Serial Number Availability
   -------------------------- --------------------- ----------------------------------------
   cluster02                 1-80-000013           Available
   ```

2. **Add `-instance` to the command to view more detailed information about the cluster peers. Include `-cluster <cluster_name>` to view results for a specific cluster. The `-instance` option displays the remote addresses that are used for intercluster communication.

   ```bash
   cluster01::> cluster peer show -cluster cluster02 -instance
   Peer Cluster Name: cluster02
   Availability: Available
   Remote Cluster Name: cluster02
   Active IP Addresses: 10.12.12.3,10.12.12.4
   Cluster Serial Number: 1-80-000013
   ```

3. **Run the `cluster peer ping` command to view information about connectivity between each intercluster address, including RTT response times.** For multiple configured cluster peers, use the `-cluster <cluster_name>` option to perform the ping for one specific peer relationship. The `cluster peer ping` command displays the results of a ping between intercluster interfaces. As mentioned earlier, when performing intercluster SnapMirror mirroring over multiple paths between the local and remote clusters, each path must have the same performance characteristics. In this example, the ping response times (RTTs) are comparatively equal to the pings to nodes where the destination cluster displays as `cluster02`.

   ```bash
   cluster01::> cluster peer ping cluster02
   Node: cluster01-01           Destination Cluster: cluster01
   Destination Node IP Address Count TTL RTT(ms) Status
   -------------------------- -------------------- ------ -------- ------------
   cluster01-01 10.12.12.1     1 255 0.186 interface_reachable
   cluster01-02 10.12.12.2     1 255 1.156 interface_reachable
   Node: cluster01-01           Destination Cluster: cluster02
   Destination Node IP Address Count TTL RTT(ms) Status
   -------------------------- -------------------- ------ -------- ------------
   cluster02-01 10.12.12.3     1 255 7.164 interface_reachable
   cluster02-02 10.12.12.4     1 255 7.065 interface_reachable
   Node: cluster01-02           Destination Cluster: cluster01
   ```
12.2 Troubleshooting SVM Peer Relationships

Here is the list of common issues and how to troubleshoot them:

- SVM peer action failure for intercluster environment:
  a. Check if both the clusters are in same league.
  b. Check if peer cluster is reachable.
  c. Check if both the clusters are running SN and SVM peering capability is enabled.
  d. Check if the peer SVM name is not associated with another cluster from peer SVM names in the SVM peering table.
  e. Check mgwd.log and console logs for error messages.

- SVM peer action failure for intracluster/intercluster environment:
  a. Check if both the clusters are running SN and SVM peering capability is enabled.
  b. Check if local and peer SVM names are not the same.
  c. Check mgwd.log and console logs for error messages.

- Run the `vserver peer show` command to verify the SVM peer relationship. This command displays all existing configured SVM peer relationships.

```
cluster02::> vserver peer show
Peer  Peer  Vserver  Vserver
----------- ----------- ----------- 
vs1_dest vs1_backup peered
vs1_dest vs1_src   peered
```

- Check for any notifications by `vserver peer show all`.

```
cluster02::> vserver peer show all
Peer  Peer  Peer  Peer  Vserver  Vserver  State  Peer Cluster Applications
----------- ----------- ----------- ----------- ----------- ----------- ----------- ----------- 
vs1_dest vs1_backup peered vs1_dest vs1_src   peered vs1_dest vs1_src   peered cluster03 snapmirror
```

12.3 Understanding SnapMirror Relationship Status

The Healthy column indicates the SnapMirror relationship status. This column is shown in the output of the `snapmirror show` command on the CLI, in the cluster element manager web interface, and as the Healthy column in the displayed status of SnapMirror relationships in OnCommand System Manager.

In this example, the `snapmirror show` command displays the Healthy column.

```
cluster02::> snapmirror show
Path  Type  Source Destination Mirror Relationship Total   Progress   Last
------------ -------------- -------------- -------------- -------------- -------------- 
vs1_src:vol1 DP  vs1_dest:vol1 Snapmirrored
```

<table>
<thead>
<tr>
<th>Destination Node IP Address</th>
<th>Count</th>
<th>TTL</th>
<th>RTT (ms)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster01-01</td>
<td>1</td>
<td>255</td>
<td>1.324</td>
<td>interface_reachable</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>1</td>
<td>255</td>
<td>0.809</td>
<td>interface_reachable</td>
</tr>
</tbody>
</table>

Node: cluster01-02  Destination Cluster: cluster02

<table>
<thead>
<tr>
<th>Destination Node IP Address</th>
<th>Count</th>
<th>TTL</th>
<th>RTT (ms)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster02-01</td>
<td>1</td>
<td>255</td>
<td>7.279</td>
<td>interface_reachable</td>
</tr>
<tr>
<td>cluster02-02</td>
<td>1</td>
<td>255</td>
<td>7.282</td>
<td>interface_reachable</td>
</tr>
</tbody>
</table>
The Healthy column displays the health of the SnapMirror relationship. It also indicates whether the RPO is maintained without needing to determine the age of the last update in order to interpret the relationship's health. For example, the Healthy column displays true for a SnapMirror relationship scheduled for regular updates if the last update completed before a following update attempted to start, as shown in the first relationship in the output presented in this example.

If a scheduled update is in progress when the next scheduled update begins, the Healthy column displays false for that relationship. Additionally, if the previously scheduled or manual update fails, then the Healthy column also displays false for that relationship.

If a transfer is currently in progress, the Healthy column displays – and the Total Progress column displays the amount of progress for the currently running transfer.

The Healthy column also displays a – when the relationship is in an uninitialized state, as shown in the third relationship. It also displays a – if the relationship is in a broken state because the snapmirror break command is used.

The Healthy column displays – for the relationship on the source system. To view authoritative information about the health of a SnapMirror relationship, look at that relationship from the destination.

The Mirror State column also displays – if the destination volume is offline or if it cannot be reached.

### 12.4 Troubleshooting SnapMirror Relationships

To determine when the last SnapMirror transfer for a specific relationship completed, refer to the exported Snapshot timestamp for instance information in clustered Data ONTAP 8.2.
Note: The last Snapshot timestamp information also displays at the bottom of the System Manager interface, as shown in Figure 22.

Figure 22: Transfer timestamp information.

For SnapMirror relationship troubleshooting issues, review information about relationships in the event log. Use the `–messagename` option with the `event log show` command to filter the event log for messages related to SnapMirror, as shown in the following example. Specify the `mgmt.snapmir*` message name to filter the output and find only messages related to SnapMirror.

```bash
cluster01::> event log show -messagename mgmt.snapmir*
Time Node Severity Event
-------------------------------
12/6/2011 17:35 cluster02-01 ERROR mgmt.snapmirror.update.fail: Update from source volume 'cluster01://vs1/vol03' to destination volume(s) 'cluster02://vs2/vol03' failed with error 'Failed to setup transfer. (Duplicate transfer specified. (Other error.)).' Job ID 1322.
12/5/2011 05:15:45 cluster02-01 DEBUG mgmt.snapmirror.abnormal.abort: Source Path deleteSnapshot, line 4285, job ID 1215.
```

To find an error message about a specific volume, filter the message list further by specifying the name of the volume, enclosed in asterisks, with the `–event` option, as shown in the following example.

```bash
cluster01::> event log show -messagename mgmt.snapmir* -event *vol01*
Time Node Severity Event
-------------------------------
```
All SnapMirror events are logged to the SnapMirror_audit.log and SnapMirror_error.log files on the node where the destination volume resides. This node might be different from the one where the command was issued. The node running the operation can be determined by running the "snapmirror show -fields destination-volume-node" command. OnCommand System Manager 3.0 allows viewing of the SnapMirror log files.

You can also use System Manager to view the SnapMirror log separately from the rest of the event logs: Cluster > Diagnostics > Logs > SnapMirror Log. From the Select node drop-down list, select the node that owns the volume in which you are interested.

13 Best Practices for DR Configurations

**Best Practices**

- Volumes that belong to one SVM at the source site should be replicated to one SVM at the destination site. An SVM is the root of a NAS namespace for NAS clients and a single storage target in SAN environments. If some NAS volumes are replicated from one SVM into different SVMs at the destination, then all of those volumes cannot be recovered into the same namespace. The same is true of volumes containing LUNs. If the volumes are replicated into different SVMs at the destination, then all of the LUNs are not presented under the same SAN target.

- The destination SVM should be a member of the same Active Directory, LDAP, or NIS domain of which the source SVM is a member. This configuration is required so that access control lists (ACLs) stored in NAS files are not broken if a NAS volume is recovered into an SVM that cannot authenticate those ACLs. The processes of changing file-level ACLs to correct them for access from a different domain can be extremely difficult and time consuming. It is also important so that authentication of tools running in SAN clients such as NetApp SnapDrive® for Windows can be done using the same credentials.

- Because the destination SVM is a different SVM than the source, and because NetApp recommends that it be a member of the same Active Directory domain, the destination SVM must be joined to the domain with a different SVM name. It is common practice to have a DR system with a different name than the source system. In DR failover scenarios, it is typical to change DNS name resolution or use DNS aliases to redirect clients to the name of the recovered storage systems so that the CIFS shares are still accessible using the same UNC path name and NFS clients are also able to access the expected path.

- Using destination volume names that are the same as the source volume names is not required but can make mounting destination volumes into the destination simpler to manage if the junction path where the volume is mounted also has the same name as the volume.

- Construct the destination NAS namespace for an SVM such that it is identical in paths and directory structure as the source SVM.

- Many SAN clients cannot access a LUN that resides in a completely read-only container, such as a SnapMirror destination volume. Generally LUNs should be mapped to igroups and mounted by SAN clients after the SnapMirror break operation is performed.

- Configure the destination SVMs ahead of time as described in the following section. This approach can greatly speed up the storage system DR process, possibly reducing it to a few SnapMirror break operations and the update of some DNS aliases.

- As new volumes are created at the source site, SnapMirror relationships must be created to replicate those volumes. Configuration settings pertaining to those volumes should be made in the DR site after the volumes are created and replicated so they can be ready in the event of a disaster.
14 Configuration and Failover for Disaster Recovery

Configuration and failover for DR are presented here in an overview of the DR process for intracluster SnapMirror DP mirrors. The process is presented in two sections. The first section provides steps that must be completed before a failover is required to prepare the destination for failover. These steps should be completed to prepare the DR site for a DR scenario. The second section provides the steps necessary to perform a failover.

Every environment has its own unique characteristics. Each environment can have an effect on a DR plan. Depending on the type of DR solutions deployed, each organization’s DR situation could be very different. To enable success, proper planning, documentation, and a realistic walkthrough of a DR scenario are required.

14.1 Environment Failover Requirements and Assumptions

To provide a successful DR experience, consider some general requirements and assumptions. The following is not an all-inclusive list. There are many variables for which to plan depending on the configuration of each environment.

- Systems administrators have access to a workstation or server desktop session from which to administer the DR site and perform the failover.
- Systems administrators have all appropriate credentials, accounts, passwords, and so on required to access the systems.
- Connectivity to the DR network is available from wherever operations are performed.
- Certain infrastructure servers already exist in the DR site and are accessible. These systems provide basic services necessary for the administrators to work in the environment and execute the recovery plan.
  - DR site Active Directory services to provide authentication
  - DR site DNS services to provide name resolution
  - DR site license servers to provide licensing services for all applications that require them

Note: It is important that a server performing the necessary Active Directory FSMO roles is available at the DR site. For information regarding transferring roles to a surviving Active Directory server or seizing these roles from a failed server, refer to Microsoft KB 255504. http://support.microsoft.com/kb/255504.

- The DR site has time synchronized to the same source as the primary site or a source in sync with the primary site.
- All required NetApp volumes are being replicated using SnapMirror to the DR site.
- The SnapMirror operations have been monitored and are up to date with respect to the designed RPO.
- The required capacity exists on the DR NetApp controller. This refers to capacity required to support day-to-day operations that have been planned for in the DR environment.
- All DR site application servers have the proper connectivity configured to be able to connect to the DR storage arrays.
- A method exists to isolate or fence the failed primary network from the DR site. This approach is necessary because, if the event causing the disaster is temporary or intermittent in nature, such as an extended power outage, when the primary site systems restart, services might conflict with the recovered operations that are then running at the DR site.
- Plans have been made for providing users and applications access to the data and services at the DR site: for example, updating DNS such that home directory mount requests to the primary site SVM are directed to the DR site SVM instead.
14.2 Preparing the Destination for Failover

Many parts of a DR process for clustered Data ONTAP 8.1 and later versions can be prepared ahead of time, prior to a DR event. For example, mounting volumes into the namespace, creating CIFS shares, assigning NFS export policies, and other things can all be done ahead of time. SnapMirror cannot be used to replicate configuration information that could be independent in the destination SVMs, such as SVM domain membership, CIFS configuration, NFS policies, Snapshot policy schedules, or NetApp storage efficiency policies.

Figure 23 illustrates volume layout for DR.

After volumes have been replicated, complete the following steps to prepare the destination system for failover, as shown in Figure 23.

NAS and SAN Environments
1. Configure the destination SVM membership into the appropriate Active Directory, LDAP, or NIS domain.
2. Determine that the destination SVM is a member of the same domain as the source SVM so that authentication is not broken for tools, such as NetApp SnapDrive for Windows, and so that the same users can be authenticated against file-level ACLs that are replicated by SnapMirror.
3. Create any nondefault Snapshot copy policies needed in the destination cluster.
   **Note:** NetApp recommends configuring Snapshot copy policies in the destination cluster with the same schedules as those in the source. Snapshot copy policies must be applied to DP volumes after failover.
4. Create NetApp storage efficiency policies in the destination SVM.
   **Note:** If NetApp storage efficiency policies are assigned to the volumes in the source SVM, a policy must be created in the destination SVM in order to schedule the dedupe process after failover at the DR site. NetApp storage efficiency policies must be applied to DP volumes after failover.
NAS Environments

1. Verify that all necessary volumes in the source SVM are being replicated to the destination SVM. Volumes can be mounted in subfolders or inside other volumes in the namespace. If this condition exists, it is important to make sure that all the volumes required to properly reconstruct the namespace at the destination are being replicated.

2. Verify security style and permissions on the destination SVM root volume. The security style and permissions of the root of the destination SVM namespace must be set correctly, or the NAS namespace might be inaccessible after failover.

3. Mount the destination NAS volumes into the destination SVM namespace.
SnapMirror does not replicate the SVM namespace junction path information. NAS volumes have no junction path, so they are not accessible after a SnapMirror break occurs unless they are premounted before failover or until they are mounted after failover.

When mounting the volumes, mount them into the namespace using the same junction path into which the source volume was mounted in the source SVM. This configuration is important so that paths in the recovered namespace are not different than paths that existed at the primary site. If the paths are different, then client mount points, links, shortcuts, and aliases might not be able to find the correct paths.

**Note:** Volumes cannot be mounted inside (nested in) other volumes that are still in a DP state. After using the `snapmirror break` command, any volume that has a mount point nested inside a replicated volume must be mounted, and any CIFS shares must be created.

4. Create CIFS shares on the destination SVM using the same share names that were used at the source. Clients are able to access the CIFS shares. However, all data is read-only until the volume is failed over.

5. Apply the proper ACLs to the CIFS shares at the destination.

6. Create appropriate NFS export policies for the destination SVM.

7. Assign the NFS export policies to the destination volumes. Clients are able to access the NFS exports. However, all data is read-only until the volume is failed over.

SAN Environments

1. If the destination SVMs use portsets, they can be configured as required before failover.

2. Configure igroups on the destination SVM.

Typically, there are different application servers that connect to the recovered storage at the DR site. The initiators from these servers can be preconfigured into appropriate igroups in the destination SVM.

Because some hosts do not support connecting to LUNs in read-only containers, which is what a SnapMirror destination volume is, mapping LUNs to igroups is normally done after failover.

SnapMirror Toolkit (for Clustered Data ONTAP 8.2 Operating System)

The main goal of this tool (SnapMirror toolkit) is to improve the user experience of setting up and running SnapMirror (and SnapVault) in clustered Data ONTAP 8.2. The feedback from QA, the usability team, and customers who participated in the clustered Data ONTAP 8.2 early validation program indicates that SnapMirror in clustered Data ONTAP 8.2 is more complicated to set up and manage than 7-Mode SnapMirror. We will improve the usability of SnapMirror in future releases of clustered Data ONTAP, but these scripts provide immediate benefit. They are lightweight and portable and provide a simpler user experience than using the clustered Data ONTAP CLI. Furthermore, customers can build in-house automated tools using these scripts as a foundation.

You can download the SnapMirror toolkit (SMTK) from the SE Utility Toolchest at [http://support.netapp.com/NOW/download/tools/smtk](http://support.netapp.com/NOW/download/tools/smtk).
14.3 Performing a Failover

With most of the configuration necessary for DR performed prior to a failover, the actual steps required to fail over during a DR scenario are greatly reduced. They are as follows.

**NAS Environment**

1. Perform a SnapMirror break operation to fail over each volume. In clustered Data ONTAP, wildcards can be used to perform a SnapMirror operation on multiple volumes with one command. The following example performs failover for all volumes in the destination SVM called vs5. It can be restricted to certain volumes by using part of the volume name in the command.

   ```bash
   cluster02::> snapmirror break -destination-path cluster02://vs5/*
   ```

2. If the volumes have been mounted in the namespace and CIFS shares and NFS export policies created and applied, clients then have read-write access to the NAS data.

3. Redirect clients to the recovered storage.

   It is common practice to have a DR system with a different name than the source system. In DR failover scenarios, it is typical to change DNS name resolution or use DNS aliases to redirect clients to the name of the recovered storage systems. This approach enables CIFS shares to be accessible using the same UNC path name, and NFS clients can also access the expected path. Alternatively, the failed source storage system might be removed from Active Directory and the recovery storage system removed and readded to Active Directory using the same name as the source system. However, it can take time for this change to propagate through a large Active Directory environment.

**SAN Environment**

1. Perform a SnapMirror break operation to fail over each volume. In clustered Data ONTAP 8.1, wildcards can be used to perform a SnapMirror operation on multiple volumes with one command. The following example performs failover for all volumes in the destination SVM called vs5. It can be restricted to certain volumes by using part of the volume name in the command.

   ```bash
   cluster02::> snapmirror break -destination-path cluster02://vs5/*
   ```

2. Make the LUNs in the volume available to the SAN clients at the DR site by mapping the LUN into the appropriate igroup.

3. On the SAN client, perform a storage rescan to detect the connected LUN.

14.4 Postfailover Volume Configuration

Snapshot copy policies and NetApp storage efficiency policies cannot be assigned to volumes in a DP state, so they must be assigned after failover.

1. If using the Data ONTAP Snapshot copy schedule, assign a Snapshot copy policy to the recovered volumes. In SAN environments, Snapshot copies are typically scheduled in the client.

2. If using NetApp storage efficiency technology, assign a storage efficiency policy to the recovered volumes.

15 SnapMirror Transition

There are existing 7-Mode customers who are using QSM and VSM (sync, semi-sync, and async). How would you transition those customers to clustered Data ONTAP? This transition is covered in TR-4052: Clustered Data ONTAP Transition Guide.
Where to Find Additional Information

The following references were used in this TR:

- TR-3975: DataMotion for Volumes Overview in Clustered Data ONTAP 8.2
- TR-4178: Infinite Volume Deployment and Implementation Guide
- TR-4052: Clustered Data ONTAP Transition Guide
- NetApp Support documentation library

Contact Us

Let us know how we can improve this technical report.
Contact us at docfeedback@netapp.com.
Include TECHNICAL REPORT 4015 in the subject line.

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Document Version History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 4.1</td>
<td>November 2019 Arpan Merchant</td>
</tr>
<tr>
<td>Version 4.0</td>
<td>November 2017 Arpan Merchant</td>
</tr>
<tr>
<td>Version 3.1</td>
<td>February 2016 Mike Worthen</td>
</tr>
<tr>
<td>Version 3.0</td>
<td>April 2015 Vincent Goveas</td>
</tr>
<tr>
<td>Version 2.2</td>
<td>November 2013 Amit Prakash Sawant</td>
</tr>
<tr>
<td>Version 2.1</td>
<td>July 2013 Amit Prakash Sawant</td>
</tr>
<tr>
<td>Version 2.0</td>
<td>April 2013 Amit Prakash Sawant</td>
</tr>
<tr>
<td>Version 1.0</td>
<td>February 2012 Larry Touchette</td>
</tr>
</tbody>
</table>
Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2019 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4015-0118