Technical Report

Oracle on MetroCluster

Integrated Data Protection, Disaster Recovery, and High Availability

Jeffrey Steiner, NetApp
December 2018 | TR-4592

Abstract

This document describes best practices, test procedures, and other considerations for operating Oracle databases on a NetApp® MetroCluster™ synchronous replication solution.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Oracle Databases and NetApp MetroCluster

NetApp MetroCluster delivers a highly available, zero data-loss solution for mission-critical Oracle workloads. In addition, integrated solutions such as MetroCluster simplify today's complicated, scale-out Oracle database, application, and virtualization infrastructures. MetroCluster replaces multiple external data protection products and strategies with one simple, central storage array that provides integrated backup, recovery, disaster recovery, and high availability (HA) within a single clustered storage system.

2 MetroCluster Technology

2.1 Data Protection with SyncMirror

At the simplest level, synchronous replication means any change must be made to both sides of mirrored storage before it is acknowledged. For example, an Oracle database is committing a transaction, and data is written to a redo log on synchronously mirrored storage. The storage system must not acknowledge the write until it has been committed to nonvolatile media on both sites. Only then is it safe to proceed without the risk of data loss.

The use of a synchronous replication technology is the first step in designing and managing a synchronous replication solution. The most important consideration is understanding what could happen during various planned and unplanned failure scenarios. Not all synchronous replication solutions offer the same capabilities. When a customer asks for a solution that delivers a recovery point objective (RPO) of zero, meaning zero data loss, all failure scenarios must be considered. In particular, what is the expected result when replication is impossible due to loss of connectivity between sites?

2.2 HA with MetroCluster

MetroCluster replication is based on NetApp SyncMirror® technology, which is designed to efficiently switch into and out of synchronous mode. This capability meets the requirements of customers who demand synchronous replication, but who also need high availability for their data services. For example, if connectivity to a remote site is severed, it is generally preferable to have the storage system continue operating in a nonreplicated state.

Many synchronous replication solutions are only capable of operating in synchronous mode. This type of all-or-nothing replication is sometimes called domino mode. Such storage systems stop serving data rather than allowing the local and remote copies of data to become unsynchronized. If replication is forcibly broken, resynchronization can be extremely time consuming and can leave a customer exposed to complete data loss during the time that mirroring is reestablished.

Not only can SyncMirror seamlessly switch out of synchronous mode if the remote site is unreachable, it can also rapidly resync to an RPO = 0 state when connectivity is restored. The stale copy of data at the remote site can also be preserved in a usable state during resynchronization, which ensures that local and remote copies of data exist at all times.

Where domino mode is required, other options besides MetroCluster exist, such as Oracle DataGuard or extended timeouts for host-side disk mirroring. Consult your NetApp or partner account team for additional information and options.

3 Database Data Protection

A database data protection architecture should be defined by business requirements. These requirements include the speed of recovery, the maximum permissible data loss, and backup retention needs. The data protection plan must also take into consideration various regulatory requirements for data retention and restoration. Finally, different data recovery scenarios must be considered, ranging from the typical and

foreseeable recovery resulting from user or application errors up to disaster recovery scenarios that include the complete loss of a site.

Small changes in data protection and recovery policies can have a significant effect on the overall architecture of storage, backup, and recovery. It is critical to define and document standards before starting design work to avoid complicating a data protection architecture. Unnecessary features or levels of protection lead to unnecessary costs and management overhead, and an initially overlooked requirement can lead a project in the wrong direction or require last-minute design changes.

3.1 Recovery Time Objective

The recovery time objective (RTO) defines the maximum time allowed for the recovery of a service. For example, a human resources database might have an RTO of 24 hours because the business can still operate, even though it would be very inconvenient to lose access to this data during the workday. In contrast, a database supporting the general ledger of a bank would have an RTO measured in minutes or even seconds. An RTO of zero is not possible, because there must be a way to differentiate between an actual service outage and a routine event such as a lost network packet. However, a near-zero RTO is a typical requirement.

3.2 Recovery Point Objective

The RPO defines the maximum tolerable data loss. In a database context, the RPO is usually a question of how much log data can be lost in a specific situation. In a typical recovery scenario in which a database is damaged due to a product bug or user error, the RPO should be zero, meaning there should be no data loss. The recovery procedure involves restoring an earlier copy of the database files and then replaying the log files to bring the database state up to the desired point in time. The log files required for this operation should already be in place in the original location.

In unusual scenarios, log data might be lost. For example, an accidental or malicious `rm –rf *` of database files could result in the deletion of all data. The only option would be to restore from backup, including log files, and some data would inevitably be lost. The only option to improve the RPO in a traditional backup environment would be to perform repeated backups of the log data. This approach has limitations, however, because of the constant data movement and the difficulty maintaining a backup system as a constantly running service. One of the benefits of advanced storage systems is the ability to protect data from accidental or malicious damage to files and thus deliver a better RPO without data movement.

3.3 Disaster Recovery

Disaster recovery includes the IT architecture, policies, and procedures required to recover a service in the event of a physical disaster. Such disasters can include floods; fire; or a human-caused disaster caused by malicious intent, negligence, or a simple error.

Disaster recovery is more than just a set of recovery procedures. It is the complete process of identifying the various risks, defining the data recovery and service continuity requirements, and delivering the right architecture with associated procedures.

When establishing data protection requirements, it is critical to differentiate between typical RPO and RTO requirements and the RPO and RTO requirements needed for disaster recovery. Some database environments require an RPO of zero and a near-zero RTO for data loss situations ranging from a relatively normal user error right up to a fire that destroys a data center. However, there are cost and administrative consequences for these high levels of protection.

In general, nondisaster data recovery requirements should be strict for two reasons. First, application bugs and user errors that damage a database are foreseeable to the point they are almost inevitable. Second, it is not difficult to design a backup strategy that can deliver an RPO of zero and a low RTO as

long as the storage system is not destroyed. There is no reason not to address a significant risk that is easily remedied, which is why the RPO and RTO targets for local recovery should be aggressive.

Disaster recovery RTO and RPO requirements vary more widely based on the likelihood of a disaster and the consequences of the associated data loss or disruption to a business. RPO and RTO requirements should be based on the actual business needs and not on general principles. They must account for multiple logical and physical disaster scenarios.

Logical Disasters

Logical disasters include data corruption caused by users, application or OS bugs, and software malfunctions. Logical disasters can also include malicious attacks by outside parties with viruses or worms or by exploiting application vulnerabilities. In these cases, the physical infrastructure is undamaged but the underlying data is no longer valid.

An increasingly common type of logical disaster is known as ransomware, in which an attack vector is used to encrypt data. Encryption does not damage the data, but it makes it unavailable until payment is made to a third party. An increasing number of enterprises are being specifically targeted with ransomware hacks.

Physical Disasters

Physical disasters include the failure of components of an infrastructure that exceeds its redundancy capabilities and results in a loss of data or an extended loss of service. For example, RAID protection provides drive-level redundancy, and the use of HBAs provides FC port and FC cable redundancy. Hardware failures of such components are foreseeable and do not affect availability.

In a database environment, it is generally possible to protect the infrastructure of an entire site with redundant components to the point where the only foreseeable physical disaster scenario is complete loss of the site. Disaster recovery planning then depends on site-to-site replication.

Synchronous and Asynchronous Data Protection

In an ideal world, all data would be synchronously replicated across geographically dispersed sites. This approach is not always feasible or even possible for several reasons:

- Synchronous replication unavoidably increases write latency because all changes must be replicated to both locations before the application or database can proceed. The resulting performance effect is frequently unacceptable, ruling out the use of synchronous mirroring.
- The increased adoption of 100% SSD storage means that additional write latency is more likely to be noticed because performance expectations include hundreds of thousands of IOPS and submillisecond latency. Gaining the full benefits of using 100% SSDs can require revisiting the disaster recovery strategy.
- Datasets continue to grow in terms of bytes, creating challenges with making sure of sufficient bandwidth to sustain synchronous replication.
- Datasets also grow in terms of complexity, creating challenges with the management of large-scale synchronous replication.
- Cloud-based strategies frequently involve greater replication distances and latency, further precluding the use of synchronous mirroring.

NetApp offers solutions that include both synchronous replication for the most exacting data recovery demands and asynchronous solutions that allow for better database performance and flexibility. In addition, NetApp technology integrates seamlessly with many third-party replication solutions, such as Oracle DataGuard and SQL Server AlwaysOn.

3.4 Retention Time

The final aspect of a data protection strategy is the data retention time, which can vary dramatically:

- A typical requirement is 14 days of nightly backups on the primary site and 90 days of backups stored on a secondary site.
- Many customers create standalone quarterly archives stored on different media.
- A constantly updated database might have no need for historical data, and backups need only be retained for a few days.

Regulatory requirements might require recoverability to the point of any arbitrary transaction in a 365-day window.

4 NetApp ONTAP Data Protection Fundamentals

4.1 Data Protection with NetApp Snapshot Copies

The foundation of NetApp ONTAP® data protection software is NetApp Snapshot® technology. The key values are as follows:

- **Simplicity.** A Snapshot copy is a read-only copy of the contents of a container of data at a specific point in time.
- **Efficiency.** Snapshot copies require no space at the moment of creation. Space is only consumed when data is changed.
- **Manageability.** A backup strategy based on Snapshot copies is easy to configure and manage because Snapshot copies are a native part of the storage OS. If the storage system is powered on, it is ready to create backups.
- **Scalability.** Up to 255 backups of a single container of files and LUNs can be preserved. For complex datasets, multiple containers of data can be protected by a single, consistent set of Snapshot copies.
- Performance is unaffected, whether a volume contains 250 Snapshot copies or none.

As a result, protecting a database running on ONTAP is simple and highly scalable. Database backups do not require movement of data. Therefore, a backup strategy can be tailored to the needs of the business rather than the limitations of network transfer rates, large number of tape drives, or disk staging areas.

4.2 Data Restoration with ONTAP SnapRestore

Rapid data restoration in ONTAP from a Snapshot copy is delivered by NetApp SnapRestore® technology. The key values are as follows:

- Individual files or LUNs can be restored in seconds, whether it is a 2TB LUN or a 4KB file.
- An entire container (a NetApp FlexVol® volume) of LUNs and/or files can be restored in seconds, whether it is 10GB or 100TB of data.

When a critical database is down, critical business operations are down. Tapes can break, and even restores from disk-based backups can be slow to transfer across the network. SnapRestore avoids these problems by delivering near instantaneous restoration of databases. Even petabyte-scale databases can be completely restored with just a few minutes of effort.

4.3 Data Replication and Disaster Recovery

Nearly every database requires data replication. At the most basic level, a replica can be a copy on tape stored off site or database-level replication to a standby database. Disaster recovery refers to the use of those replica copies to bring a service online in the event of catastrophic loss of service.

ONTAP offers multiple replication options to address a variety of requirements natively within the storage array, covering a complete spectrum of needs. These options can include simple replication of backups to a remote site up to a synchronous, fully automated solution that delivers both disaster recovery and HA in the same platform.

The primary ONTAP replication technologies applicable to databases are the NetApp SnapMirror® and NetApp SyncMirror technologies. These are not add-on products; rather they are fully integrated into ONTAP and are activated by the simple addition of a license key. Storage-level replication is not the only option either. Database-level replication, such as with Oracle DataGuard or Microsoft SQL Server AlwaysOn, can also integrate into a data protection strategy based on ONTAP.

The right choice depends on the specific replication, recovery, and retention requirements.

ONTAP SnapMirror

SnapMirror is the NetApp asynchronous replication solution, ideally suited for protecting large, complicated, and dynamic datasets such as databases and their associated applications. Its key values are as follows:

- **Manageability.** SnapMirror is easy to configure and manage because it is a native part of the storage software. No add-on products are required. Replication relationships can be established in minutes and can be managed directly on the storage system.
- **Simplicity.** Replication is based on FlexVol volumes, which are containers of LUNs or files that are replicated as a single consistent group.
- **Efficiency.** After the initial replication relationship is established, only changes are replicated. Furthermore, efficiency features such as deduplication and compression are preserved, further reducing the amount of data that must be transferred to a remote site.
- **Flexibility.** Mirrors can be temporarily broken to allow testing of disaster recovery procedures, and then the mirroring can be easily reestablished with no need for a complete remirroring. Only the changed data must be applied to bring the mirrors back into sync. Mirroring can also be reversed to allow a rapid resync after the disaster concludes and the original site is back in service. Finally, read-write clones of replicated data are available for testing and development.

MetroCluster and SyncMirror

Synchronous replication in ONTAP is delivered by SyncMirror. At the simplest layer, SyncMirror creates two complete sets of RAID-protected data in two different locations. They could be in adjoining rooms within a data center, or they could be located many kilometers apart.

SyncMirror is fully integrated with ONTAP and operates just above the RAID level. Therefore, all the usual ONTAP features, such as Snapshot copies, SnapRestore, and NetApp FlexClone®, work seamlessly. It is still ONTAP. It just includes an additional layer of synchronous data mirroring.

A collection of ONTAP controllers managing SyncMirror data is called NetApp MetroCluster. Many configurations are available, and the primary purpose of MetroCluster is to provide high-availability access to synchronously mirrored data in a variety of typical and disaster recovery failure scenarios.

The key values of data protection with MetroCluster and SyncMirror are as follows:

- In normal operations, SyncMirror delivers guaranteed synchronous mirroring across locations. A write operation is not acknowledged until it is present on nonvolatile media on both sites.

- If connectivity between sites fails, SyncMirror automatically switches into asynchronous mode to keep the primary site serving data until connectivity is restored. When restored, it delivers rapid resynchronization by efficiently updating the changes that have accumulated on the primary site. Full reinitialization is not required.

SnapMirror is also fully compatible with systems based on SyncMirror. For example, a primary database might be running on a MetroCluster cluster spread across two geographic sites. This database can also replicate backups to a third site as long-term archives or for the creation of clones in a DevOps environment.

5 MetroCluster Platforms

ONTAP software is the foundation for advanced data protection and management. However, ONTAP only refers to software. There are two MetroCluster ONTAP hardware platforms from which to choose:

- ONTAP on All Flash FAS (AFF) and FAS MetroCluster
- NetApp Private Storage (NPS) for Cloud

The key concept is that ONTAP is ONTAP. Some hardware options offer better performance, others offer lower costs, and some run within hyperscaler clouds. The core functions of ONTAP are unchanged, with multiple replication options available to bind different ONTAP platforms into a single solution. As a result, data protection and disaster recovery strategies can be built on real-world needs, such as performance requirements, capex/opex considerations, and overall cloud strategy. The underlying storage technology runs anywhere in any environment.

5.1 ONTAP with AFF and FAS Controllers

For maximum performance and control of data, ONTAP on a physical AFF or FAS controller remains the leading solution. This option is the standard on which thousands of customers have relied for more than 20 years. ONTAP delivers solutions for any environment, ranging from three mission-critical databases to 60,000-database service provider deployments, instant restores of petabyte-scale databases, and DBaaS involving hundreds of clones of a single database.

5.2 NetApp Private Storage for Cloud

NetApp introduced the NPS option to address the needs of data-intensive workloads in the public cloud. Although many public cloud storage options exist, most of them have limitations in terms of performance, control, or scale. With respect to database workloads, the primary limitations are as follows:

- Many public cloud storage options do not scale to the IOPS levels required by modern database workloads in terms of cost, efficiency, or manageability.
- Even when the raw IOPS capabilities of a public cloud provider meet requirements, the I/O latencies are frequently unacceptable for database workloads. This limitation has become even more clear as databases have migrated to all-flash storage arrays, and businesses have begun to measure latency in terms of microseconds, not milliseconds.
- Although public cloud storage availability is good overall, it does not yet meet the demands of most mission-critical environments.
- Backup and recovery capabilities exist within public cloud storage services, but they generally cannot meet the zero RPO and near-zero RTO requirements of most databases. Data protection requires true instant snapshot-based backup and recovery, not streaming backup and recovery to and from elsewhere in a cloud.
- Hybrid cloud environments must move data between on-premises and cloud storage systems, mandating a common foundation for storage management.
- Many governments have strict data sovereignty laws that prohibit relocating data outside national borders.

NPS systems deliver maximum storage performance, control, and flexibility to public cloud providers, including Amazon AWS, Microsoft Azure, and IBM SoftLayer. This capability is delivered by AFF and FAS systems, including MetroCluster options, in data centers connected directly to public clouds. The full power of the hyperscaler compute layer can be used without the limitations of hyperscaler storage. Furthermore, NPS enables cloud-independent and multicloud architectures because the data, such as application binaries, databases, database backups, and archives, all remains wholly within the NPS system. There is no need to expend time, bandwidth, or money moving data between cloud providers.

Notably, some NetApp customers have used the NPS model on their own initiative. In many locations, high-speed access to one of the hyperscaler providers is readily available to customer data center facilities. In other cases, customers use a colocation facility that is already capable of providing high-speed access to hyperscaler cloud providers. This approach has led to the use of Amazon AWS, Azure, and SoftLayer as essentially on-demand, consumption-based sources of virtualized servers. In some cases, nothing has changed about the customers' day-to-day operations. They simply use the hyperscaler services as a more powerful, flexible, and cost-efficient replacement for their traditional virtualization infrastructure.

Options are also available for NPS as a service (NPSaaS). In many cases, the demands of database environments are substantial enough to warrant purchasing an NPS system at a colocation facility. However, in some cases, customers prefer to utilize both cloud servers and cloud storage as an operational expense rather than a capital expense. In these cases, they want to use storage resources purely as an as-needed, on-demand service. Several providers now offer NPS as a service for such customers.

6 MetroCluster Physical Architecture

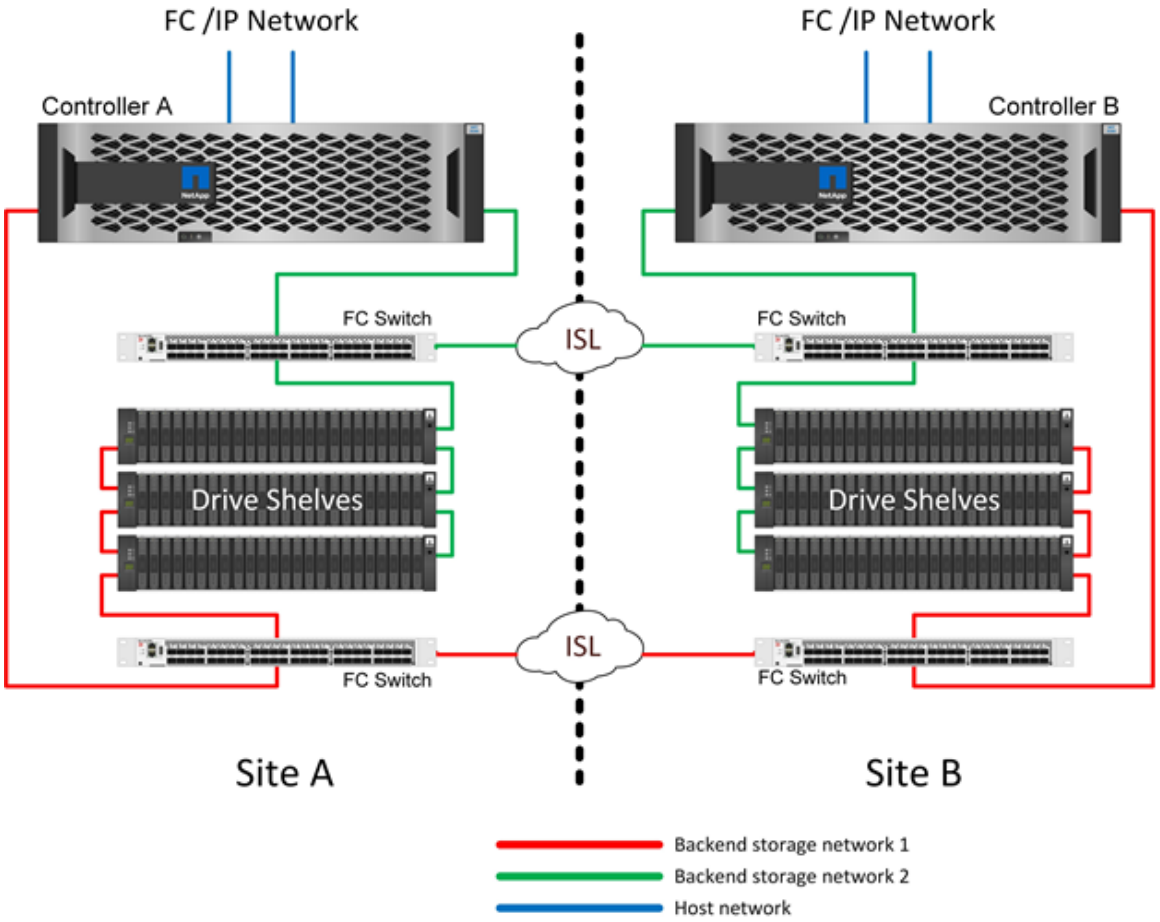
MetroCluster clusters are available in two basic configurations. The two-node version is deployed as a single node per site, and the HA-pair version uses clustered HA nodes on each site.

6.1 Two-Node MetroCluster

The two-node MetroCluster configuration uses only one node per site. This design is simpler than the HA-pair option because there are fewer components to configure and maintain. It also has reduced infrastructure demands in terms of cabling and FC switching. Finally, it reduces costs.

The basic design is depicted in Figure 1.

Figure 1) Two-node MetroCluster basic architecture.



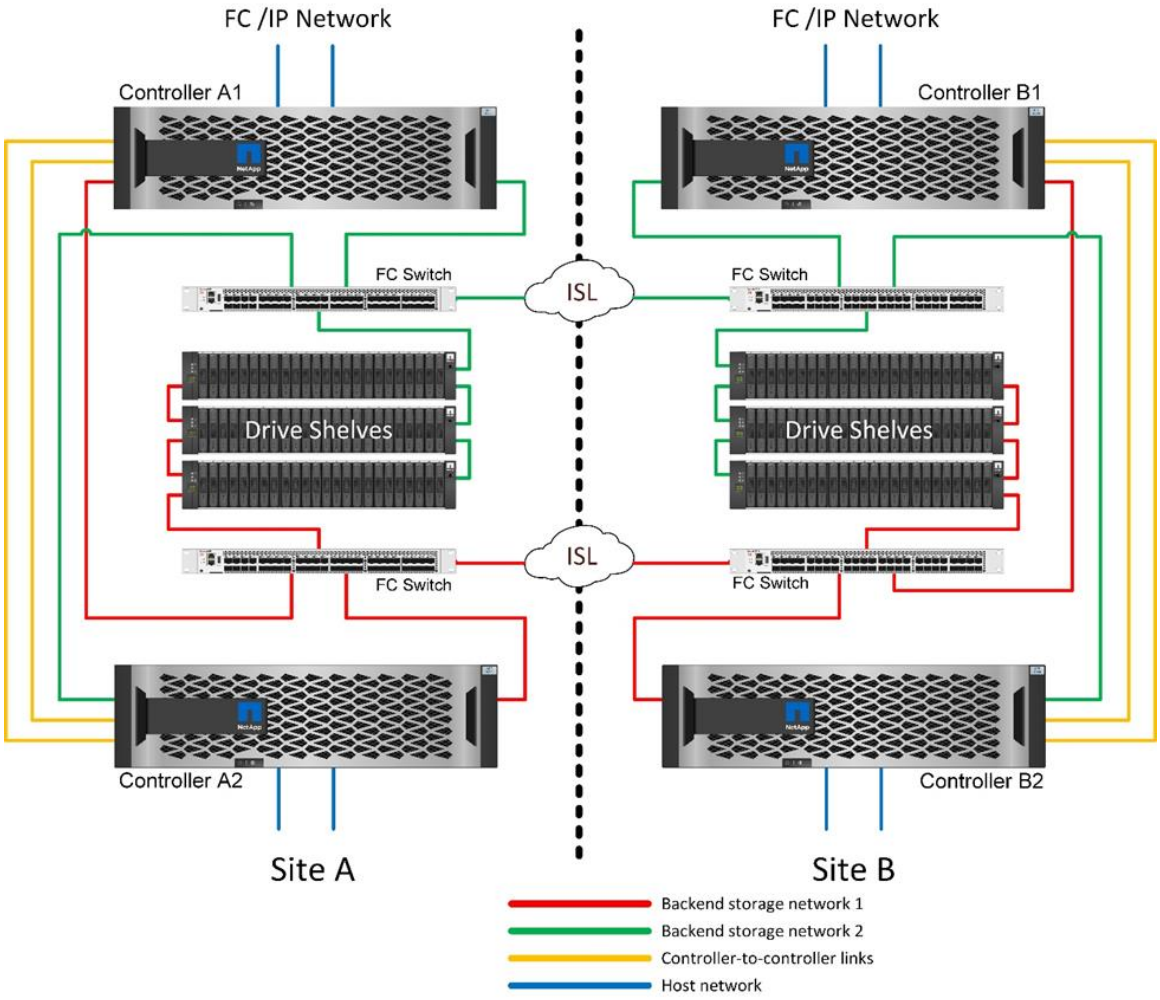
The obvious impact of this design is that controller failure on a single site means that data is available from the opposite site. This restriction is not necessarily a problem. Many enterprises have multisite data center operations with stretched, high-speed, low-latency networks that function essentially as a single infrastructure. In these cases, the two-node version of MetroCluster is the preferred configuration. Two-node systems are currently used at petabyte scale by several service providers.

6.2 HA-Pair MetroCluster

The HA-pair MetroCluster configuration uses two nodes per site. This configuration option increases the complexity and costs relative to the two-node option, but it delivers an important benefit: intrasite redundancy. A simple controller failure does not require data access across the WAN. Data access remains local through the alternate local controller.

The basic design is depicted in Figure 2.

Figure 2) HA-pair MetroCluster basic architecture.



Some multisite infrastructures are not designed for active-active operations, but rather are used more as a primary site and disaster recovery site. In this situation, the HA-pair MetroCluster option is generally preferable for the following reasons:

- Although a two-node MetroCluster cluster is an HA system, unexpected failure of a controller or planned maintenance requires that data services must come online on the opposite site. If the network connectivity between sites cannot support the required bandwidth, performance is affected. The only option would be to also fail over the various host OSs and associated services to the alternate site. The HA-pair MetroCluster cluster eliminates this problem because loss of a controller results in simple failover within the same site.
- Some network topologies are not designed for cross-site access, but instead use different subnets or isolated FC SANs. In these cases, the two-node MetroCluster cluster no longer functions as an HA system because the alternate controller cannot serve data to the servers on the opposite site. The HA-pair MetroCluster option is required to deliver complete redundancy.
- If a two-site infrastructure is viewed as a single highly available infrastructure, the two-node MetroCluster configuration is suitable. However, if the system must function for an extended period of time after site failure, then an HA pair is preferred because it continues to provide HA within a single site.

6.3 MetroCluster Resiliency Features

As shown in Figures 1 and 2, there are no single points of failure in a MetroCluster solution:

- Each controller has two independent paths to the drive shelves on the local site.
- Each controller has two independent paths to the drive shelves on the remote site.
- Each controller has two independent paths to the controllers on the opposite site.
- In the HA-pair configuration, each controller has two paths to its local partner.

In summary, any one component in the configuration can be removed without compromising the ability of MetroCluster to serve data. The only difference in terms of resiliency between the two options is that the HA-pair version is still an overall HA storage system after a site failure.

7 MetroCluster Logical Architecture

There are two fundamental requirements for any storage system: make sure that data is protected and make sure that data is available. A complete explanation of ONTAP data protection technologies is beyond the scope of this document, but a review of the layers is required to fully understand what happens with various fault scenarios.

7.1 Data Protection

Logical data protection within MetroCluster consists of the following key requirements:

- Data transmission on the network must be protected against data corruption.
- Data written to disk must be protected against data corruption.
- Data written to disk must be protected against drive failure.
- Changes to data must be protected against loss.
- The two independent copies of data, one at each site, must be kept in sync

Network Corruption: Checksums

The most basic level of data protection is the checksum, which is a special error-detecting code stored alongside data. Corruption of data during network transmission is detected with the use of a checksum and, in some instances, multiple checksums.

For example, an FC frame includes a form of checksum called a cyclic redundancy check (CRC) to make sure that the payload is not corrupted in transit. The transmitter sends both the data and the CRC of the data. The receiver of an FC frame recalculates the CRC of the received data to make sure that it matches the transmitted CRC. If the newly computed CRC does not match the CRC attached to the frame, the data is corrupt, and the FC frame is discarded or rejected. An iSCSI I/O operation includes checksums at the TCP/IP and Ethernet layers, and, for extra protection, it can also include optional CRC protection at the SCSI layer. Any bit corruption on the wire is detected by the TCP layer or IP layer, which results in retransmission of the packet. As with FC, errors in the SCSI CRC result in a discard or rejection of the operation.

Drive Corruption: Checksums

Checksums are also used to verify the integrity of data stored on drives. Data blocks written to drives are stored with a checksum function that yields an unpredictable number that is tied to the original data. When data is read from the drive, the checksum is recomputed and compared to the stored checksum. If it does not match, then the data has become corrupt and must be recovered by the RAID layer.

Data Corruption: Lost Writes

One of the most difficult types of corruption to detect is a lost or a misplaced write. When a write is acknowledged, it must be written to the media in the correct location. In-place data corruption is relatively easy to detect by using a simple checksum stored with the data. However, if the write is simply lost, then the prior version of data might still exist, and the checksum would be correct. If the write is placed at the wrong physical location, the associated checksum would once again be valid for the stored data, even though the write has destroyed other data.

The solution to this challenge is as follows:

- A write operation must include metadata that indicates the location where the write is expected to be found.
- A write operation must include some sort of version identifier.

When ONTAP writes a block, it includes data on where the block belongs. If a subsequent read identifies a block, but the metadata indicates that it belongs at location 123 when it was found at location 456, then the write has been misplaced.

Detecting a wholly lost write is more difficult. The explanation is very complicated, but essentially ONTAP is storing metadata in a way that a write operation results in updates to two different locations on the drives. If a write is lost, a subsequent read of the data and associated metadata shows two different version identities. The difference indicates that the write was not completed by the drive.

Lost and misplaced write corruption is exceedingly rare, but, as drives continue to grow and datasets push into exabyte scale, the risk increases. Lost write detection should be included in any storage system supporting database workloads.

Drive Failures: RAID, RAID DP, and RAID-TEC

If a block of data on a drive is discovered to be corrupt, or the entire drive fails and is wholly unavailable, the data must be reconstituted. This reconstitution is done in ONTAP by using parity drives. Data is striped across multiple data drives, and then parity data is generated. The parity data is stored separately from the "real" data.

ONTAP originally used RAID 4, which uses a single parity drive for each group of data drives. The result was that any one drive in the group could fail without resulting in data loss. If the parity drive failed, no data was damaged, and a new parity drive could be constructed. If a single data drive failed, the remaining drives could be used with the parity drive to regenerate the missing data.

When drives were small, the statistical chance of two drives failing simultaneously was negligible. As drive capacities have grown, so has the time required to reconstruct data after a drive failure. This additional time has increased the window in which a second drive failure would result in data loss. In addition, the rebuild process creates a lot of additional I/O on the surviving drives. As drives age, the risk of the additional load leading to a second drive failure also increases. Finally, even if the risk of data loss did not increase with the continued use of RAID 4, the consequences of data loss would become more severe. The more data that would be lost in the event of a RAID group failure, the longer it would take to recover the data, extending business disruption.

These issues led NetApp to develop the NetApp RAID DP® technology, a variant of RAID 6. This solution includes two parity drives, meaning that any two drives in a RAID group can fail without creating data loss. Drives have continued to grow in size, which eventually led NetApp to develop the NetApp RAID-TEC™ technology, which introduces a third parity drive.

Some historical database best practices recommend the use of RAID 10, also known as striped mirroring. This type of mirroring offers less data protection than even RAID DP because there are multiple two-disk failure scenarios, whereas in RAID DP there are none.

There are also some historical database best practices that indicate RAID 10 is preferred to RAID 4/5/6 options due to performance concerns. These recommendations sometimes refer to a RAID penalty. Although these recommendations are generally correct, they are inapplicable to the implementations of RAID within ONTAP. The performance concern is related to parity regeneration. With traditional RAID implementations, processing the routine random writes performed by a database requires multiple disk reads to regenerate the parity data and complete the write. The penalty is defined as the additional read IOPS required to perform write operations.

ONTAP does not incur a RAID penalty because writes are staged in memory where parity is generated and then written to disk as a single RAID stripe. No reads are required to complete the write operation.

In summary, when compared to RAID 10, RAID DP and RAID-TEC deliver much more usable capacity, better protection against drive failure, and no performance sacrifice.

Hardware Failure Protection: NVRAM

Any storage array servicing a database workload must service write operations as quickly as possible. Furthermore, a write operation must be protected from loss caused by an unexpected event such as a power or device failure.

AFF and FAS systems rely on NVRAM to meet these requirements. The write process works as follows:

1. The inbound write data is stored in RAM.
2. The changes that must be made to data on disk are journaled into NVRAM on both the local and partner nodes. NVRAM is not a write cache; rather it is a journal similar to a database redo log. Under normal conditions, it is not read. It is only used for recovery, such as after a power failure during I/O processing.
3. The write is then acknowledged to the host

The write process at this stage is complete from the application point of view, and the data is protected against loss because it is stored in two different locations. Eventually, the changes are written to disk, but this process is out of band from the application point of view because it occurs after the write is acknowledged and therefore does not affect latency. This process is once again similar to database logging. A change to the database is recorded in the redo logs as quickly as possible, and the change is then acknowledged as committed. The updates to the data files occur much later and do not directly affect the speed of processing.

In the event of a controller failure, the partner controller takes ownership of the required disks and replays the logged data in NVRAM to recover any I/O operations that were in flight when the failure occurred.

Site Failure Protection: NVRAM and MetroCluster

MetroCluster extends NVRAM data protection in the following ways:

- In a two-node configuration, NVRAM data is replicated using the interswitch links (ISLs) to the remote partner.
- In an HA-pair configuration, NVRAM data is replicated to both the local partner and a remote partner.
- A write is not acknowledged until it is replicated to all partners

This architecture protects in-flight I/O from site failure by replicating NVRAM data to a remote partner. This process is not involved with drive-level data replication. The controller that owns the aggregates is responsible for data replication by writing to both plexes in the aggregate, but there still must be protection against in-flight I/O loss in the event of site loss. Replicated NVRAM data is only used if a partner controller must take over for a failed controller.

Site and Shelf Failure Protection: SyncMirror and Plexes

SyncMirror is a mirroring technology that enhances, but does not replace, RAID DP or RAID-TEC. It mirrors the contents of two independent RAID groups. The logical configuration is as follows:

1. Drives are configured into two pools based on location. One pool is composed of all drives on site A, and the second pool is composed of all drives on site B.
2. A common pool of storage, known as an aggregate, is then created based on mirrored sets of RAID groups. An equal number of drives is drawn from each site. For example, a 20-drive SyncMirror aggregate would be composed of 10 drives from site A and 10 drives from site B.
3. Each set of drives on a given site is automatically configured as one or more fully redundant RAID DP or RAID-TEC groups, independent of the use of mirroring. This use of RAID underneath mirroring provides data protection even after the loss of a site.

Figure 3) SyncMirror.

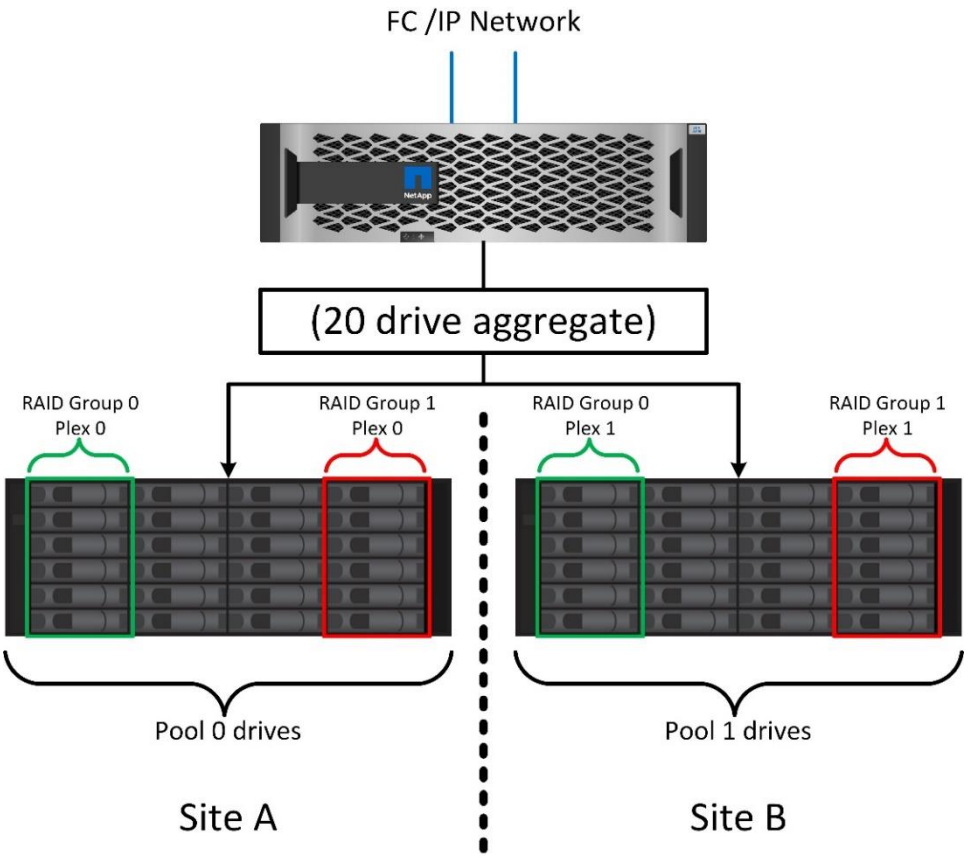


Figure 3 illustrates a sample SyncMirror configuration. A 24-drive aggregate was created on the controller with 12 drives from a shelf allocated on site A and 12 drives from a shelf allocated on site B. The drives were grouped into two mirrored RAID groups. RAID group 0 includes a 6-drive plex on site A mirrored to a 6-drive plex on site B. Likewise, RAID group 1 includes a 6-drive plex on site A mirrored to a 6-drive plex on site B.

SyncMirror is normally used to provide remote mirroring with MetroCluster systems, with one copy of the data at each site. On occasion, it has been used to provide an extra level of redundancy in a single system. In particular, it provides shelf-level redundancy. A drive shelf already contains dual power supplies and controllers and is overall little more than sheet metal, but in some cases the extra protection might be warranted. For example, one NetApp customer has deployed SyncMirror for a mobile real-time

analytics platform used during automotive testing. The system was separated into two physical racks supplied with independent power feeds and independent UPS systems.

Redundancy Failure: NVFAIL

As discussed earlier, a write is not acknowledged until it has been logged into local NVRAM and NVRAM on at least one other controller. This approach makes sure that a hardware failure or power outage does not result in the loss of in-flight I/O. If the local NVRAM fails or the connectivity to other nodes fails, then data would no longer be mirrored.

If the local NVRAM reports an error, the node shuts down. This shutdown results in failover to a partner controller when HA pairs are used. With MetroCluster, the behavior depends on the overall configuration chosen, but it can result in automatic failover to the remote note. In any case, no data is lost because the controller experiencing the failure has not acknowledged the write operation.

A site-to-site connectivity failure that blocks NVRAM replication to remote nodes is a more complicated situation. Writes are no longer replicated to the remote nodes, creating a possibility of data loss if a catastrophic error occurs on a controller. More importantly, attempting to fail over to a different node during these conditions results in data loss.

The controlling factor is whether NVRAM is synchronized. If NVRAM is synchronized, node-to-node failover is safe to proceed without risk of data loss. In a MetroCluster configuration, if NVRAM and the underlying aggregate plexes are in sync, it is safe to proceed with switchover without risk of data loss.

ONTAP does not permit a failover or switchover when the data is out of sync unless the failover or switchover is forced. Forcing a change in conditions in this manner acknowledges that data might be left behind in the original controller and that data loss is acceptable.

Databases are especially vulnerable to corruption if a failover or switchover is forced because databases maintain larger internal caches of data on disk. If a forced failover or switchover occurs, previously acknowledged changes are effectively discarded. The contents of the storage array effectively jump backward in time, and the state of the database cache no longer reflects the state of the data on disk.

To protect databases from this situation, ONTAP allows volumes to be configured for special protection against NVRAM failure. When triggered, this protection mechanism results in a volume entering a state called NVFAIL. This state results in I/O errors that cause a database crash. This crash causes the databases to shut down so that they do not use stale data. Data should not be lost because any committed transaction data should be present in the logs. The usual next steps are for an administrator to fully shut down the hosts before manually placing the LUNs and volumes back online again. Although these steps can involve some work, this approach is the safest way to make sure of data integrity. Not all data requires this protection, which is why NVFAIL behavior can be configured on a volume-by-volume basis.

7.2 High Availability

A complete description of ONTAP HA features is beyond the scope of this document. However, as with data protection, a basic understanding of this functionality is important when designing a database infrastructure.

HA Pairs

The basic unit of HA is the HA pair. Each pair contains redundant links to support replication of NVRAM data. NVRAM is not write cache. The RAM inside a controller serves as the write cache. The purpose of NVRAM is to temporarily journal data as a safeguard against unexpected system failure. In this respect, it is similar to a database redo log.

Both NVRAM and a database redo log are used to store data quickly, allowing changes to data to be committed as quickly as possible. The update to the persistent data on drives (or datafiles) does not take

place until later during a process called a checkpoint on both ONTAP and most database platforms. Neither NVRAM data nor database redo logs are read during normal operations.

If a controller fails abruptly, there are likely to be pending changes stored in NVRAM that have not yet been written to the drives. The partner controller detects the failure, takes control of the drives, and applies the required changes that have been stored in NVRAM.

HA Pairs and MetroCluster

MetroCluster is available in two configurations: two-node and HA pair. The two-node configuration behaves the same as an HA pair with respect to NVRAM. In the event of sudden failure, the partner node can replay NVRAM data to make the drives consistent and make sure that no acknowledged writes have been lost.

The HA-pair configuration replicates NVRAM to the local partner node as well. A simple controller failure results in an NVRAM replay on the partner node, as is the case with a standalone HA-pair without MetroCluster. In the event of sudden complete site loss, the remote site also has the NVRAM required to make the drives consistent and start serving data.

One important aspect of MetroCluster is that the remote nodes have no access to partner data under normal operational conditions. Each site functions essentially as an independent system that can assume the personality of the opposite site. This process is known as a switchover and includes a planned switchover in which site operations are migrated nondisruptively to the opposite site. It also includes unplanned situations in which a site is lost and a manual or automatic switchover is required as part of disaster recovery.

Takeover and Giveback

Takeover and giveback refers to the process of transferring responsibility for storage resources between nodes in an HA pair. There are two aspects to takeover and giveback:

- Management of the network connectivity that allows access to the drives
- Management of the drives themselves

Network interfaces supporting CIFS and NFS traffic are configured with both a home and failover location. A takeover includes moving the network interfaces to their temporary home on a physical interface located on the same subnets as the original location. A giveback includes moving the network interfaces back to their original locations. The exact behavior can be tuned as required.

Network interfaces supporting SAN block protocols such as iSCSI and FC are not relocated during takeover and giveback. Instead, LUNs should be provisioned with paths that include a complete HA pair, which results in a primary path and a secondary path.

Note: Additional paths to additional controllers can also be configured to support relocating data between nodes in a larger cluster, but this relocation is not part of the HA process.

The second aspect of takeover and giveback is the transfer of disk ownership. The exact process depends on multiple factors, including the reason for the takeover/giveback and the command line options issued. The goal is to perform the operation as efficiently as possible. Although the overall process might appear to require several minutes, the actual moment in which ownership of the drive is transitioned from node to node can generally be measured in seconds.

Takeover Time

Host I/O experiences a short pause in I/O during takeover and giveback operations, but there should not be application disruption in a correctly configured environment. The actual transition process in which I/O is delayed is generally measured in seconds, but the host might require additional time to recognize the change in data paths and resubmit I/O operations.

The nature of the disruption depends on the protocol:

- A network interface supporting NFS and CIFS traffic issues an Address Resolution Protocol (ARP) request to the network after the transition to a new physical location. This request causes the network switches to update their MAC address tables and resume processing I/O. Disruption in the case of planned takeover and giveback is usually measured in seconds and, in many cases, is not detectable. Some networks might be slower to fully recognize the change in network path, and some OSs might queue up a lot of I/O in a very short time that must be retried. This queuing can extend the time required to resume I/O.
- A network interface supporting SAN protocols does not transition to a new location. A host OS must change the path or paths in use. The pause in I/O observed by the host depends on multiple factors. From a storage system point of view, the period when I/O cannot be served is just a few seconds. However, different host OSs might require additional time to allow an I/O to time out before retry. Newer OSs are better able to recognize a path change much more quickly, but older OSs typically require up to 30 seconds to recognize a change.

The expected takeover times during which the storage system cannot serve data to a database environment are shown in Table 1.

Table 1) Expected takeover times.

	NAS	SAN Optimized OS	SAN
Planned takeover	15 seconds	2–10 seconds	2–10 seconds
Unplanned takeover	30 seconds	2–15 seconds	30 seconds

Takeover Triggers

Takeovers can occur under the following conditions:

- Manual initiation of a takeover with the `storage failover takeover` command.
- A software or system failure occurs that leads to a controller panic. After the panic completes and the system reboots, the storage resources are given back, returning the system to normal. This behavior is a default that can be changed if desired.
- A controller has a complete system failure, such as loss of power, and cannot reboot.
- A partner controller fails to receive a heartbeat message. This situation could happen if the partner experiences a hardware or software failure that does not result in a panic but still prevents it from functioning correctly.
- Manually halting a node can trigger a takeover, unless the command is executed with an `-inhibit_takeover` parameter of `true` that prevents a takeover.
- Manually rebooting a node can trigger a takeover, unless the command is executed with a `-inhibit_takeover` parameter of `true` that prevents a takeover.
- If hardware-assisted takeover is enabled, it can trigger a takeover when the service processor detects failure of the partner node.

Hardware-Assisted Takeover

The service processor is an out-of-band management device embedded in AFF and FAS systems. It is accessed by its own IP address and is used for direct console access and other management functions irrespective of whether the controller is operational.

ONTAP by itself can trigger a takeover of a failed node after it no longer detects the heartbeat from the partner node, but there are timeouts involved. Hardware-assisted takeover uses the service process to

speed up the takeover process by more quickly detecting failures and immediately initiating the takeover. It does not wait for ONTAP to recognize that the partner's heartbeat has stopped.

Switchover and Switchback

The terms switchover and switchback refer to the process of transitioning volumes between remote controllers in a MetroCluster configuration. This process only applies to the remote nodes. When MetroCluster is used in a four-volume configuration, local node failover is the same takeover and giveback process described previously.

Planned Switchover and Switchback

A planned switchover or switchback is similar to a takeover or giveback between nodes. The process has multiple steps and might appear to require several minutes, but what is actually happening is a multiphase graceful transition of storage and network resources. The moment when control transfers occurs much more quickly than the time required for the complete command to execute.

The primary difference between takeover/giveback and switchover/switchback is with the effect on FC SAN connectivity. With local takeover/giveback, a host experiences the loss of all FC paths to the local node and relies on its native MPIO to change over to available alternate paths. Ports are not relocated. With switchover and switchback, the virtual FC target ports on the controllers transition to the other site. They effectively cease to exist on the SAN for a moment and then reappear on an alternate controller.

SyncMirror Timeouts

SyncMirror is a ONTAP mirroring technology that provides protection against shelf failures. When shelves are separated across a distance, the result is remote data protection.

SyncMirror does not deliver universal synchronous mirroring. The result is better availability. Some storage systems use constant all-or-nothing mirroring, sometimes called domino mode. This form of mirroring is limited in application because all write activity must cease if the connection to the remote site is lost. Otherwise, a write would exist at one site but not at the other. Typically, such environments are configured to take LUNs offline if site-to-site connectivity is lost for more than a short period (such as 30 seconds).

This behavior is desirable for a small subset of database environments. However, most databases demand a solution that delivers guaranteed synchronous replication under normal operating conditions, but with the ability to suspend replication. A complete loss of site-to-site connectivity is frequently considered a near-disaster situation. Typically, such database environments are kept online and serving data until connectivity is repaired or a formal decision is made to shut down the database to protect data. A requirement for automatic shutdown of the database purely because of remote replication failure is unusual.

SyncMirror supports synchronous mirroring requirements with the flexibility of a timeout. If connectivity to the remote controller and/or plex is lost, a 30-second timer begins counting down. When the counter reaches 0, write I/O processing resumes using the local data. The remote copy of the data is usable, but it is frozen in time until connectivity is restored. Resynchronization leverages aggregate-level snapshots to return the system to synchronous mode as quickly as possible.

Notably, in many cases, this sort of universal all-or-nothing domino mode replication is better implemented at the application layer. For example, Oracle DataGuard includes maximum protection mode, which guarantees long-instance replication under all circumstances. If the replication link fails for a period exceeding a configurable timeout, the databases shut down.

Automatic Unattended Switchover

Automatic unattended switchover (AUSO) is a MetroCluster feature that delivers a form of cross-site HA. As discussed previously, MetroCluster is available in two types: a single controller on each site or an HA pair on each site. The principal advantage of the HA option is that planned or unplanned controller shutdown still allows all I/O to be local. The advantage of the single-node option is reduced costs, complexity, and infrastructure.

The primary value of AUSO is to improve the HA capabilities of MetroCluster systems. Each site monitors the health of the opposite site, and, if no nodes remain to serve data, AUSO results in rapid switchover. This approach is especially useful in MetroCluster configurations with just a single node per site because it brings the configuration closer to an HA pair in terms of availability.

AUSO cannot offer comprehensive monitoring at the level of an HA pair. An HA pair can deliver extremely high availability because it includes two redundant physical cables for direct node-to-node communication. Furthermore, both nodes in an HA pair have access to the same set of disks on redundant loops, delivering another route for one node to monitor the health of another.

MetroCluster clusters exist across sites for which both node-to-node communication and disk access rely on the site-to-site network connectivity. The ability to monitor the heartbeat of the rest of the cluster is limited. AUSO has to discriminate between a situation where the other site is actually down rather than unavailable due to a network problem.

As a result, a controller in an HA pair can prompt a takeover if it detects a controller failure that occurred for a specific reason, such as a system panic. It can also prompt a takeover if there is a complete loss of connectivity, sometimes known as a lost heartbeat.

A MetroCluster system can only safely perform an automatic switchover when a specific fault is detected on the original site. Also, the controller taking ownership of the storage system must be able to guarantee that disk and NVRAM data is in sync. The controller cannot guarantee the safety of a switchover just because it lost contact with the source site, which could still be operational. For additional options for automating a switchover, see the information on the MetroCluster tiebreaker (MCTB) solution in the next section.

Tiebreakers

The MCTB solution is a standard design third-site tiebreaker (also called a witness server). It operates similar to a tiebreaker for any clustered solution, including application clusters and operating system clusters. Its principal purpose is to trigger site failover while preventing a split-brain situation in which two sites are serving the same data.

The basic summary of a tiebreaker's operation is as follows:

1. The tiebreaker software runs at a third site with connectivity to the two sites where the MetroCluster nodes are running.
2. If the tiebreaker loses all contact with one of the sites, it performs a health check on the surviving site.
3. If the surviving site also confirms a complete loss of contact with the opposite side, the tiebreaker server breaks the tie and issues an alert or, optionally, a switchover command.

This logic covers the following scenarios:

- If the tiebreaker loses contact with one site and yet the other site confirms connectivity, then the problem is an interruption between the tiebreaker and the site. There is no loss of service and no need for a switchover.
- If the tiebreaker loses contact with both sites, then it cannot trigger any actions, and nothing happens.
- If the tiebreaker loses contact with one site, and the other site also verifies loss of connectivity, then a disaster is presumed to have occurred, and the tiebreaker issues an alert or a switchover command.

The MCTB solution also includes multiple tunable settings and monitoring options. It cannot address some rolling disaster situations, such as when site-to-site connectivity is completely lost for an extended period of time before a site itself is lost.

For further information, see the official MCTB documentation or contact your NetApp account team.

8 Oracle and NVFAIL

If a failover or switchover is forced, databases are vulnerable to corruption because they maintain large internal caches. If a forced failover or forced MetroCluster switchover occurs, previously acknowledged changes are effectively discarded. The contents of the storage array jump backward in time, and the state of the database cache no longer reflects the state of the data on disk. This inconsistency results in data corruption.

Caching can occur at the application or server layer. For example, an Oracle Real Application Cluster (RAC) configuration with servers active on both a primary and a remote site caches data within the Oracle SGA. A forced switchover operation that resulted in lost data would put the database at risk of corruption because the blocks stored in the SGA might not match the blocks on disk.

A less obvious use of caching is at the OS file system layer. Blocks from a mounted NFS file system might be cached in the OS. Alternatively, a clustered file system based on LUNs located on the primary site could be mounted on servers at the remote site, and once again data could be cached. A failure of NVRAM or a forced takeover or forced switchover in these situations could result in file system corruption.

ONTAP systems protect databases and operating systems from this scenario with NVFAIL and its associated parameters.

8.1 nvfail

Any database volume on ONTAP storage should have the `nvfail` parameter set to `on`.

This setting protects the volume from a catastrophic failure of NVRAM journaling that puts data integrity in question. The `nvfail` parameter takes effect during startup. If NVRAM errors are detected, then there might be uncommitted changes that have been lost, and the drive state might not match the database cache. ONTAP then sets volumes with an `nvfail` parameter of `on` to `in-nvfailed-state`. As a result, any database process attempting to access the data receives an I/O error, which leads to a protective crash or shutdown of the database.

8.2 in-nvfailed-state

A volume with an `in-nvfailed-state` of `true` returns an error on I/O operations. Specifically, NFS access returns a stale file handle (`ESTALE`) to the client, and LUN access fails because the LUNs are forced offline. NFS files remain inaccessible until the `in-nvfailed-state` flag is cleared. LUNs remain offline until the `in-nvfailed-state` flag is cleared from the volume and the LUNs are brought online again.

8.3 dr-force-nvfail

The `dr-force-nvfail` parameter protects data against certain unplanned MetroCluster switchover events. As discussed in the section 7.2, SyncMirror has a 30-second timeout upon the loss of remote connectivity. It is possible that a rolling disaster could first interrupt replication and some minutes later destroy the remainder of the site. In addition, it is possible that a disaster could strike during maintenance when the states of the primary and remote copies of data were not in sync. If any applications were accessing data from the remote site, then a switchover could result in an older copy of data that does not match the state of the cache.

The nvfail parameter is primarily intended to address takeover and giveback procedures within a single HA pair. While it does apply to MetroCluster switchover events where NVRAM inconsistencies are detected, it does not protect against loss of data during a forced switchover. An administrator performing a forced switchover is essentially acknowledging that the state of the controller at the disaster site may have unreplicated data, but the surviving copy of the data must be activated anyway. Databases may also need to be protected in this situation with a second parameter called dr-force-nvfail. Volumes with dr-force-nvfail enter in-nvfailed-state during a forced switchover. The reason this NVFAIL is forced is because the data might not actually be out of sync. The remote data could be completely consistent with the original data, but there is no way to guarantee consistency if the primary site is unreachable.

The primary situation in which dr-force-nvfail is needed is a database cluster in which servers on both the primary and remote site are accessing data on the storage system. As a result, the servers on the remote site might have cached data that was committed to the primary site but was not replicated to the remote site due to maintenance or an unusual rolling disaster. A forced switchover in these circumstances would risk data corruption if the remote copy of data did not precisely match the state of the primary site.

If no servers were accessing data from the remote site at the point of the disaster, there would generally be no reason to set dr-force-nvfail on a volume. It is still possible that a rolling disaster could result in a surviving copy of data that does not match the original data, but dr-force-nvfail provides no protection if there is no cached data that must be forcibly purged.

For maximum protection, any volume that might be accessed by a remote server that caches data should have dr-force-nvfail set to on. It is only safe to leave the parameter set to off for data that is not cached by an application or operating system or when the switchover process is guaranteed to occur within the 30-second SyncMirror timeout.

Caution: Prior to ONTAP 9.0, the dr-force-nvfail parameter also placed volumes into the nvfailed state during a graceful, planned switchover.

8.4 force-nvfail-all

The -force-nvfail-all parameter is an optional argument used with a switchover command. It sets the in-nvfailed-state parameter to true for all volumes being switched over, and it sets the -dr-force-nvfail parameter to true for any volumes that do not already have it enabled.

The primary use for this argument is a disaster situation in which the remote site must be forced online and there are questions about whether data on the original site was cached somewhere on the disaster recovery site. If the nvfail and dr-force-nvfail parameters have been set correctly on all volumes, it is unnecessary to use force-nvfail-all. However, it still might be desirable to use -force-nvfail-all for extra certainty. Drawbacks would include a need to clear the in-nvfailed-state flag from all the volumes and the I/O errors that would occur on hosts trying to access those volumes.

9 Oracle Single Instance on MetroCluster

As stated previously, the presence of a MetroCluster system does not necessarily add to or change any best practices for operating a database. The majority of databases currently running on customer MetroCluster systems are single instance and follow the recommendations in TR-3633: Oracle Databases on ONTAP. Standby server resources exist on the remote site, but no special automation or integration efforts have been made.

9.1 Failover with a Preconfigured OS

SyncMirror delivers a synchronous copy of the data at the disaster recovery site, but making that data available requires an operating system and the associated applications. Basic automation can

dramatically improve the failover time of the overall environment. Clusterware products such as Veritas Cluster Server (VCS) are often used to create a cluster across the sites, and in many cases the failover process can be driven with simple scripts.

If the primary nodes are lost, the clusterware (or scripts) is configured to bring the databases online at the alternate site. One option is to create standby servers that are preconfigured for the NFS or SAN resources that make up the database. If the primary site fails, the clusterware or scripted alternative performs a sequence of actions similar to the following:

1. Forcing a MetroCluster switchover
2. Performing discovery of FC LUNs (SAN only)
3. Mounting file systems and/or mounting ASM disk groups
4. Starting the database

The primary requirement of this approach is a running OS in place on the remote site. It must be preconfigured with Oracle binaries, which also means that tasks such as Oracle patching must be performed on the primary and standby site. Alternatively, the Oracle binaries can be mirrored to the remote site and mounted if a disaster is declared.

The actual activation procedure is simple. Commands such as LUN discovery require just a few commands per FC port. File system mounting is nothing more than a `mount` command, and both databases and ASM can be started and stopped at the CLI with a single command. If the volumes and file systems are not in use at the disaster recovery site prior to the switchover, there is no requirement to set `dr-force-nvfail` on volumes.

9.2 Failover with a Virtualized OS

Failover of database environments can be extended to include the operating system itself. In theory, this failover can be done with boot LUNs, but most often it is done with a virtualized OS. The procedure is similar to the following steps:

1. Forcing a MetroCluster switchover
2. Mounting the datastores hosting the database server virtual machines
3. Starting the virtual machines
4. Starting databases manually or configuring the virtual machines to automatically start the databases

For example, an ESX cluster could span sites. In the event of disaster, the virtual machines can be brought online at the disaster recovery site after the switchover. As long as the datastores hosting the virtualized database servers are not in use at the time of the disaster, there is no requirement for setting `dr-force-nvfail` on associated volumes.

10 Extended Oracle RAC on MetroCluster

Many customers optimize their RTO by stretching an Oracle RAC cluster across sites, yielding a fully active-active configuration. The overall design becomes more complicated because it must include quorum management of Oracle RAC. Additionally, data is accessed from both sites, which means a forced switchover might lead to the use of an out-of-date copy of the data.

Although a copy of the data is present on both sites, only the controller that currently owns an aggregate can serve data. Therefore, with stretched RAC, the nodes that are remote must perform I/O across a site-to-site connection. The result is added I/O latency, but this latency is not generally a problem. The RAC interconnect network must also be stretched across sites, which means a high-speed, low-latency network is required anyway. If the added latency does cause a problem, the cluster can be operated in an active-passive manner. I/O-intensive operations would then need to be directed to the RAC nodes that

are local to the controller that owns the aggregates. The remote nodes then perform lighter I/O operations or are used purely as warm standby servers.

If active-active stretched RAC is required, ASM mirroring should be considered in place of MetroCluster. ASM mirroring allows a specific replica of the data to be preferred. Therefore, a stretched RAC cluster can be built in which all reads occur locally. Read I/O never crosses sites, which delivers the lowest possible latency. All write activity must still transit the intersite connection, but such traffic is unavoidable with any synchronous mirroring solution.

Note: If boot LUNs, including virtualized boot disks, are used with Oracle RAC, the `misscount` parameter might need to be changed. For more information about RAC timeout parameters, see TR-3633: Oracle Databases on ONTAP.

10.1 Two-Site Configuration

A two-site stretched RAC configuration can deliver active-active database services that can survive many, but not all, disaster scenarios nondisruptively.

RAC Voting Disks

The first consideration when deploying stretched RAC on MetroCluster should be quorum management. Oracle RAC has two mechanisms to manage quorum: disk heartbeat and network heartbeat. The disk heartbeat monitors storage access using the voting disks. With a single-site RAC configuration, a single voting resource is sufficient as long as the underlying storage system offers HA capabilities.

The first difficulty with a two-site configuration is making sure that each site can always access more than half of the voting disks in a way that guarantees a nondisruptive disaster recovery process. For example:

- One common voting disk guarantees eviction of one site if intersite connectivity is lost.
- One voting disk per site guarantees node eviction on both sites if intersite connectivity is lost because neither site would have a voting disk quorum.
- Two voting disks on one site and a single voting disk on the other site allows for active-active operations when both sites are operational and reachable. However, if the single-disk site is lost or isolated from the network, then that site is evicted.

RAC Network Heartbeat

The Oracle RAC network heartbeat monitors node reachability across the cluster interconnect. To remain in the cluster, a node must be able to contact more than half of the other nodes. In a two-site architecture, this requirement creates the following choices for the RAC node count:

- Placement of an equal number of nodes per site results in eviction at one site in the event network connectivity is lost.
- Placement of *N* nodes on one site and *N*+1 nodes on the opposite site guarantees that loss of intersite connectivity results in the site with the larger number of nodes remaining in network quorum and the site with fewer nodes evicting.

Prior to Oracle 12cR2, it was not feasible to control which side would experience an eviction during site loss. When each site has an equal number of nodes, eviction is controlled by the master node, which in general is the first RAC node to boot.

Oracle 12cR2 introduces node weighting capability. This capability gives an administrator more control over how Oracle resolves split-brain conditions. As a simple example, the following command sets the preference for a particular node in an RAC:

```
[root@jfs3-a ~]# /grid/bin/crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed. Restart
Oracle High Availability Services for new value to take effect.
```

After restarting Oracle High-Availability Services, the configuration looks as follows:

```
[root@jfs3-a lib]# /grid/bin/crsctl status server -f | egrep '^NAME|CSS_CRITICAL='
NAME=jfs3-a
CSS_CRITICAL=yes
NAME=jfs3-b
CSS_CRITICAL=no
```

Node `jfs3-a` is now designated as the critical server. If the two RAC nodes are isolated, jfs3-a survives, and `jfs3-b` is evicted.

Note: For complete details, see the Oracle white paper “Oracle Clusterware 12c Release 2 Technical Overview.”

For versions of Oracle RAC prior to 12cR2, the master node can be identified by checking the CRS logs as follows:

```
[root@jfs3-a ~]# /grid/bin/crsctl status server -f | egrep '^NAME|CSS_CRITICAL='
NAME=jfs3-a
CSS_CRITICAL=yes
NAME=jfs3-b
CSS_CRITICAL=no
 [root@jfs3-a ~]# grep -i 'master node' /grid/diag/crs/jfs3-a/crs/trace/crsd.trc
2017-05-04 04:46:12.261525 :   CRSSE:2130671360: {1:16377:2} Master Change Event; New Master Node
ID:1 This Node's ID:1
2017-05-04 05:01:24.979716 :   CRSSE:2031576832: {1:13237:2} Master Change Event; New Master Node
ID:2 This Node's ID:1
2017-05-04 05:11:22.995707 :   CRSSE:2031576832: {1:13237:221} Master Change Event; New Master
Node ID:1 This Node's ID:1
2017-05-04 05:28:25.797860 :   CRSSE:3336529664: {1:8557:2} Master Change Event; New Master Node
ID:2 This Node's ID:1
```

This log indicates that the master node is `2` and the node `jfs3-a` has an ID of `1`. This fact means that `jfs3-a` is not the master node. The identity of the master node can be confirmed with the command `olsnodes -n`.

```
[root@jfs3-a ~]# /grid/bin/olsnodes -n
jfs3-a  1
jfs3-b  2
```

The node with an ID of `2` is `jfs3-b`, which is the master node. In a configuration with equal numbers of nodes on each site, the site with `jfs3-b` is the site that survives if the two sets lose network connectivity for any reason.

It is possible that the log entry that identifies the master node can age out of the system. In this situation, the timestamps of the Oracle Cluster Registry (OCR) backups can be used.

```
[root@jfs3-a ~]#  /grid/bin/ocrconfig -showbackup

jfs3-b     2017/05/05 05:39:53     /grid/cdata/jfs3-cluster/backup00.ocr     0

jfs3-b     2017/05/05 01:39:53     /grid/cdata/jfs3-cluster/backup01.ocr     0

jfs3-b     2017/05/04 21:39:52     /grid/cdata/jfs3-cluster/backup02.ocr     0

jfs3-a     2017/05/04 02:05:36     /grid/cdata/jfs3-cluster/day.ocr     0

jfs3-a     2017/04/22 02:05:17     /grid/cdata/jfs3-cluster/week.ocr     0
```

This example shows that the master node is `jfs3-b`. It also indicates a change in the master node from `jfs3-a` to `jfs3-b` somewhere between 2:05 and 21:39 on May 4. This method of identifying the master node is only safe to use if the CRS logs have also been checked because it is possible that the master node has changed since the previous OCR backup. If this change has occurred, then it should be visible in the OCR logs.

Most customers choose a single voting disk that services the entire environment and an equal number of RAC nodes on each site. If this voting disk is on ASM, then all the underlying ASM LUNs should be on a single site. The voting disk should be placed on the site that owns the aggregates that contain the database. The result is that loss of connectivity results in eviction on the remote site. The remote site would no longer have quorum, nor would it have access to the database files, but the local site continues running as usual. When connectivity is restored, the remote instance can be brought online again.

In the event of disaster, a switchover is required to bring the database files and voting LUNs online on the surviving site. If the disaster allows AUSO to trigger the switchover, NVFAIL is not triggered because the cluster is known to be in sync, and the storage resources come online normally. AUSO is a very fast operation and should complete before the `disktimeout` period expires.

Because there are only two sites, it is not feasible to use any type of automated external tiebreaking software, which means forced switchover must be a manual operation.

10.2 Three-Site Configuration

A stretched RAC cluster is much easier to architect with three sites. The two sites hosting each half of the MetroCluster system also support the database workloads, while the third site serves as a tiebreaker for both the database and the MetroCluster system.

Oracle Tiebreaker

The database tiebreaker is a voting resource on the third site. This resource does not have to match the type or protocol of the voting resources on the primary site. For example, the primary site might be using FCP exclusively for storage connectivity, while the third-site voting resource is an NFS file system or iSCSI LUN.

One concern with using an NFS file system as a voting resource is the fact that it is hard mounted. This type of file system mount can lead to hangs on system commands such as `df` or `ls` if connectivity to the third site is interrupted. This arrangement is generally not a problem with Oracle database operation, but it can interfere with other software that performs tasks such as monitoring file system resources or automated software installation.

MetroCluster Tiebreaker

The NetApp MetroCluster Tiebreaker software can run on a third site to monitor the health of the MetroCluster environment, send notifications, and optionally force a switchover in a disaster situation. A complete description of the Tiebreaker can be found on the NetApp support site, but the primary purpose of the MetroCluster Tiebreaker is to detect site loss. It must also discriminate between site loss and a loss of connectivity. For example, switchover should not occur because the Tiebreaker was unable to reach the primary site, which is why the Tiebreaker also monitors the remote site's ability to contact the primary site.

Automatic switchover with AUSO is also compatible with the MCTB. AUSO reacts very quickly because it is designed to detect specific failure events and then invoke the switchover only when NVRAM and SyncMirror plexes are in sync.

In contrast, the tiebreaker is located remotely and therefore must wait for a timer to elapse before declaring a site dead. The tiebreaker eventually detects the sort of controller failure covered by AUSO, but in general AUSO has already started the switchover and possibly completed the switchover before the tiebreaker acts. The resulting second switchover command coming from the tiebreaker would be rejected.

Caution: The MCTB software does not verify that NVRAM was and/or plexes are in sync when forcing a switchover. Automatic switchover, if configured, should be disabled during maintenance activities that result in loss of sync for NVRAM or SyncMirror plexes.

Additionally, the MCTB might not address a rolling disaster that leads to the following sequence of events:

1. Connectivity between sites is interrupted for more than 30 seconds.
2. SyncMirror replication times out, and operations continue on the primary site, leaving the remote replica stale.
3. The primary site is lost.

The result is the presence of unreplicated changes on the primary site. A switchover might then be undesirable for a number of reasons, including the following:

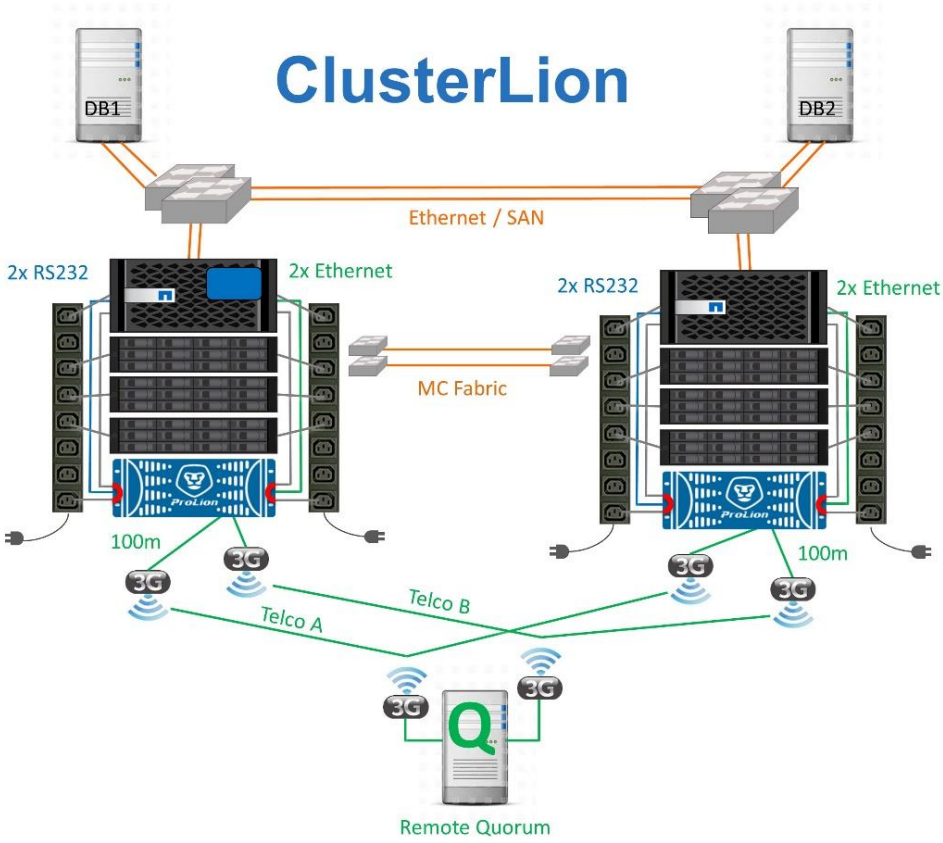
- Critical data might be present on the primary site, and that data might be eventually recoverable. A switchover that allowed the database to continue operating would effectively discard that critical data.
- A database on the surviving site that was using storage resources on the primary site at the time of site loss might have cached data. A switchover would introduce a stale version of the data that does not match the cache.
- An operating system on the surviving site that was using storage resources on the primary site at the time of site loss might have cached data. A switchover would introduce a stale version of the data that does not match the cache.

The safest option is to configure the tiebreaker to send an alert if it detects site failure and then have a person make a decision on whether to force a switchover. Databases and/or operating systems might first need to be shut down to clear any cached data. In addition, the NVFAIL settings can be used to add further protection and help streamline the failover process.

10.3 Virtual Third Site with ClusterLion

ClusterLion is an advanced MetroCluster monitoring appliance that functions as a virtual third site. This approach allows MetroCluster to be safely deployed in a two-site configuration with fully automated switchover capability. Complete documentation is available from ProLion. Figure 4 shows some of the highlights.

Figure 4) ProLion Architecture



- The ClusterLion appliances monitor the health of the controllers with directly connected Ethernet and serial cables.
- The two appliances are connected to each other with redundant 3G wireless connections.
- Power to the ONTAP controller is routed through internal relays. In the event of a site failure, ClusterLion, which contains an internal UPS system, cuts the power connections before invoking a switchover. This process makes sure that no split-brain condition occurs.
- ClusterLion performs a switchover within the 30-second SyncMirror timeout or not at all.
- ClusterLion does not perform a switchover unless the states of NVRAM and SyncMirror plexes are in sync.
- Because ClusterLion only performs a switchover if MetroCluster is fully in sync, NVFAIL is not required. This configuration permits site-spanning environments such as an extended Oracle RAC to remain online, even during an unplanned switchover.

11 Extended RAC and NVFAIL

11.1 Extended RAC with Manually Forced NVFAIL

The safest option to force a switchover with an extended RAC cluster is by specifying `-force-nvfail-all` at the command line. This option is available as an emergency measure to make sure that all cached data is flushed. If a host is using storage resources originally located on the disaster-stricken site, it receives either I/O errors or a stale file handle (`ESTALE`) error. Oracle databases crash, and file systems either go offline entirely or switch to read-only mode.

After the switchover completes, the `in-nvfailed-state` flag needs to be cleared, and the LUNs need to be placed online. After this activity is complete, the database can be restarted. These tasks can be automated to reduce the RTO.

11.2 Extended RAC with dr-force-nvfail

The safest configuration is to set the `dr-force-nvfail` flag on all volumes that might be accessed from a remote site. As a result, the volumes become unavailable when they enter `in-nvfailed-state` during a switchover. After the switchover completes, the `in-nvfailed-state` flag must be cleared, and the LUNs must be placed online. After these activities are complete, the database can be restarted. These tasks can be automated to reduce the RTO.

The result is similar to using the `-force-nvfail-all` flag. However, the number of volumes affected can be limited to just those volumes that must be protected from applications or operating systems with stale caches.

11.3 Extended RAC Without dr-force-nvfail

There are two critical requirements for an environment that does not use `dr-force-nvfail` on database volumes:

- A forced switchover must occur no more than 30 seconds after primary site loss.
- A switchover must not occur during maintenance tasks or any other conditions in which SyncMirror plexes or NVRAM replication are out of sync.

The first requirement can be met by using tiebreaker software that is configured to perform a switchover within 30 seconds of a site failure. This requirement does not mean the switchover must be performed within 30 seconds of the detection of a site failure. It does mean that it is no longer safe to force a switchover if 30 seconds has elapsed since a site was confirmed to be operational.

The second requirement can be partially met by disabling all automated switchover capabilities when the MetroCluster configuration is known to be out of sync. A better option is to have a tiebreaker solution that can monitor the health of NVRAM replication and the SyncMirror plexes. If the cluster is not fully synchronized, the tiebreaker should not trigger a switchover.

The NetApp MCTB software cannot monitor the synchronization status, so it should be disabled when MetroCluster is not in sync for any reason. ClusterLion does include NVRAM-monitoring and plex-monitoring capabilities and can be configured to not trigger the switchover unless the MetroCluster system is confirmed to be fully synchronized.

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.

TR-4592-0717