Oracle Database

Enterprise applications

NetApp
January 12, 2026

Table of Contents

Oracle Database

Oracle databases on ONTAP

ONTAP is designed for Oracle databases. For decades, ONTAP has been optimized for the unique demands of relational database I/O and multiple ONTAP features were created specifically to service the needs of Oracle databases and even at the request of Oracle Inc. itself.

This documentation replaces these previously published technical reports *TR-3633: Oracle databases on ONTAP; TR-4591: Oracle data protection: Backup, recovery, replication; TR-4592: Oracle on MetroCluster; and TR-4534: Migration of Oracle Databases to NetApp Storage Systems*

In addition to the many possible ways ONTAP brings value to your database environment, there is also a wide variety of user requirements, including database size, performance requirements, and data protection needs. Known deployments of NetApp storage include everything from a virtualized environment of approximately 6,000 databases running under VMware ESX to a single-instance data warehouse currently sized at 996TB and growing. As a result, there are few clear best practices for configuring an Oracle database on NetApp storage.

The requirements for operating an Oracle database on NetApp storage are addressed in two ways. First, when a clear best practice exists, it will be called out specifically. At a high level, many design considerations that must be addressed by architects of Oracle storage solutions based on their specific business requirements will be explained.

ONTAP configuration on AFF/FAS systems

RAID

RAID refers to the use of redundancy to protect data against the loss of a drive.

Questions occasionally arise concerning RAID levels in the configuration of NetApp storage used for Oracle databases and other enterprise applications. Many legacy Oracle best practices regarding storage array configuration contain warnings about using RAID mirroring and/or avoiding certain types of RAID. Although they raise valid points, these sources do not apply to RAID 4 and the NetApp RAID DP and RAID-TEC technologies used in ONTAP.

RAID 4, RAID 5, RAID 6, RAID DP, and RAID-TEC all use parity to ensure drive failure does not result in data loss. These RAID options offer much better storage utilization in comparison to mirroring, but most RAID implementations have a drawback that affects write operations. Completion of a write operation on other RAID implementations may require multiple drive reads to regenerate the parity data, a process commonly called the RAID penalty.

ONTAP, however, does not incur this RAID penalty. This is because of the integration of NetApp WAFL (Write Anywhere File Layout) with the RAID layer. Write operations are coalesced in RAM and prepared as a complete RAID stripe, including parity generation. ONTAP does not need to perform a read in order to complete a write, which means that ONTAP and WAFL avoid the RAID penalty. Performance for latency-critical operations, such as redo logging, is unimpeded, and random data-file writes do not incur any RAID penalty resulting from a need to regenerate parity.

With respect to statistical reliability, even RAID DP offers better protection than RAID mirroring. The primary problem is the demand made on drives during a RAID rebuild. With a mirrored RAID set, the risk of data loss from a drive failing while rebuilding to its partner in the RAID set is much greater than the risk of a triple- drive failure in a RAID DP set.

Capacity management

Managing a database or other enterprise application with predictable, manageable, high performance enterprise storage requires some free space on the drives for data and metadata management. The amount of free space required depends on the type of drive used, and business processes.

Free space is defined as any space that is not used for actual data and includes unallocated space on the aggregate itself and unused space within the constituent volumes. Thin provisioning must also be considered. For example, a volume might contain a 1TB LUN of which only 50% is utilized by real data. In a thin provisioned environment, this would correctly appear to be consuming 500GB of space. However, in a fully provisioned environment, the full capacity of 1TB appears to be in use. The 500GB of unallocated space is hidden. This space is unused by actual data and should therefore be included in the calculation of total free space.

NetApp recommendations for storage systems used for enterprise applications are as follows:

SSD aggregates, including AFF systems

NetApp recommends a minimum of 10% free space. This includes all unused space, including free space within the aggregate or a volume and any free space that is allocated due to the use of full provisioning but is not used by actual data. Logical space is unimportant, the question is how much actual free physical space is available for data storage.

The recommendation of 10% free space is very conservative. SSD aggregates can support workloads at even higher levels of utilization without any effect on performance. However, as the utilization of the aggregate increases, the risk of running out of space also increases if utilization is not monitored carefully. Furthermore, while running a system at 99% capacity may not incur a performance penalty, but it would likely incur management effort trying to keep it from filling up completely while additional hardware is ordered, and it may take some time to procure and install additional drives.

HDD aggregates, including Flash Pool aggregates

NetApp recommends a minimum of 15% free space when spinning drives are used. This includes all unused space, including free space within the aggregate or a volume and any free space that is allocated due to the use of full provisioning but is not used by actual data. Performance will be impacted as free space approaches 10%.

Storage Virtual Machines

Oracle database storage management is centralized on a Storage Virtual Machine (SVM)

An SVM, known as a vserver at the ONTAP CLI, is a basic functional unit of storage, and it is useful to compare an SVM to a guest on a VMware ESX server.

When first installed, ESX has no pre-configured capabilities, such as hosting a guest OS or supporting an end-user application. It is an empty container until a virtual machine (VM) is defined. ONTAP is similar. When

ONTAP is first installed, it has no data-serving capabilities until an SVM is created. It is the SVM personality that defines the data services.

As with other aspects of storage architecture, the best options for SVM and logical interface (LIF) design depend heavily on scaling requirements and business needs.

SVMs

There is no official best practice for provisioning SVMs for ONTAP. The right approach depends on management and security requirements.

Most customers operate one primary SVM for most of their day-to-day requirements but then create a small number of SVMs for special needs. For example, you might wish to create:

- An SVM for a critical business database managed by a specialist team
- An SVM for a development group to whom complete administrative control has been given so that they can manage their own storage independently
- An SVM for sensitive business data, such as human resources or financial reporting data, for which the administrative team must be limited

In a multi-tenant environment, each tenant's data can be given a dedicated SVM. The limit for the number of SVMs and LIFs per cluster, HA pair, and node are dependant on the protocol being used, the node model, and the version of ONTAP. Consult the NetApp Hardware Universe for these limits.

Performance management with ONTAP QoS

Safely and efficiently managing multiple Oracle databases requires an effective QoS strategy. The reason is the ever-increasing performance capabilities of a modern storage system.

Specifically, the increased adoption of all-flash storage has enabled the consolidation of workloads. Storage arrays relying on spinning media tended to support only a limited number of I/O-intensive workloads because of the limited IOPS capabilities of older rotational drive technology. One or two highly active databases would saturate the underlying drives long before the storage controllers reached their limits. This has changed. The performance capability of a relatively small number of SSD drives can saturate even the most powerful storage controllers. This means the full capabilities of the controllers can be leveraged without the fear of sudden collapse of performance as spinning media latency spikes.

As a reference example, a simple two-node HA AFF A800 system is capable of servicing up to one million random IOPS before latency climbs above one millisecond. Very few single workloads would be expected to reach such levels. Fully utilizing this AFF A800 system array will involve hosting multiple workloads, and doing this safely while ensuring predictability requires QoS controls.

There are two types of quality of service (QoS) in ONTAP: IOPS and bandwidth. QoS controls can be applied to SVMs, volumes, LUNs, and files.

IOPS QoS

An IOPS QoS control is obviously based on the total IOPS of a given resource, but there are a number of aspects of IOPS QoS that might not be intuitive. A few customers were initially puzzled by the apparent increase in latency when an IOPS threshold is reached. Increasing latency is the natural result of limiting IOPS. Logically, it functions similarly to a token system. For example, if a given volume containing datafiles has a 10K IOPS limit, each I/O that arrives must first receive a token to continue processing. So long as no more than

10K tokens have been consumed in a given second, no delays are present. If IO operations must wait to receive their token, this wait appears as additional latency. The harder a workload pushes up against the QoS limit, the longer each IO must wait in the queue for its turn to be processed, which appears to the user as higher latency.

> Be cautious when applying QoS controls to database transaction/redo log data. While the performance demands of redo logging are normally much, much lower than datafiles, the redo log activity is bursty. The IO happens in brief pulses, and a QoS limit that appears appropriate for average redo IO levels may be too low for the actual requirements. The result can be severe performance limitations as QoS engages with each redo log burst. In general, redo and archive logging should not be limited by QoS.

Bandwidth QoS

Not all I/O sizes are the same. For example, a database might be performing a large number of small block reads which would result in the IOPS threshold being reached, but databases might also be performing a full table scan operation which would consist of a very small number of large block reads, consuming a very large amount of bandwidth but relatively few IOPS.

Likewise, a VMware environment might drive a very high number of random IOPS during boot-up, but would perform fewer but larger IOs during an external backup.

Sometimes effectively managing performance require either IOPS or bandwidth QoS limits, or even both.

Minimum/guaranteed QoS

Many customers seek a solution that includes guaranteed QoS, which is more difficult to achieve than it might seem and is potentially quite wasteful. For example, placing 10 databases with a 10K IOPS guarantee requires sizing a system for a scenario in which all 10 databases are simultaneously running at 10K IOPS, for a total of 100K.

The best use for minimum QoS controls is to protect critical workloads. For example, consider an ONTAP controller with a maximum possible IOPS of 500K and a mix of production and development workloads. You should apply maximum QoS policies to development workloads to prevent any given database from monopolizing the controller. You would then apply minimum QoS policies to production workloads to make sure that they always have the required IOPS available when needed.

Adaptive QoS

Adaptive QoS refers to the ONTAP feature where the QoS limit is based on the capacity of the storage object. It is rarely used with databases because there is not usually any link between the size of a database and its performance requirements. Large databases can be nearly inert, while smaller databases can be the most IOPS-intensive.

Adaptive QoS can be very useful with virtualization datastores because the IOPS requirements of such datasets do tend to correlate to the total size of the database. A newer datastore containing 1TB of VMDK files is likely to need about half the performance as a 2TB datastore. Adaptive QoS allows you to grow the QoS limits automatically as the datastore becomes populated with data.

Efficiency

ONTAP space efficiency features are optimized for Oracle databases. In almost all cases, the best approach is to leave the defaults in place with all efficiency features enabled.

Space efficiency features, such as compression, compaction, and deduplication are designed to increase the amount of logical data that fits on a given amount of physical storage. The result is lower costs and management overhead.

At a high level, compression is a mathematical process whereby patterns in data are detected and encoded in a way that reduces space requirements. In contrast, deduplication detects actual repeated blocks of data and removes the extraneous copies. Compaction allows multiple logical blocks of data to share the same physical block on media.

See the sections below on thin provisioning for an explanation of the interaction between storage efficiency and fractional reservation.

Compression

Prior to the availability of all-flash storage systems, array-based compression was of limited value because most I/O-intensive workloads required a very large number of spindles to provide acceptable performance. Storage systems invariably contained much more capacity than required as a side effect of the large number of drives. The situation has changed with the rise of solid-state storage. There is no longer a need to vastly overprovision drives purely to obtain good performance. The drive space in a storage system can be matched to actual capacity needs.

The increased IOPS capability of solid-state drives (SSDs) almost always yields cost savings compared to spinning drives, but compression can achieve further savings by increasing the effective capacity of solid-state media.

There are several ways to compress data. Many databases include their own compression capabilities, but this is rarely observed in customer environments. The reason is usually the performance penalty for a **change** to compressed data, plus with some applications there are high licensing costs for database-level compression. Finally, there is the overall performance consequences to database operations. It makes little sense to pay a high per-CPU license cost for a CPU that performs data compression and decompression rather than real database work. A better option is to offload the compression work on to the storage system.

Adaptive compression

Adaptive compression has been thoroughly tested with enterprise workloads with no observed effect on performance, even in an all-flash environment in which latency is measured in microseconds. Some customers have even reported a performance increase with the use of compression because the data remains compressed in cache, effectively increasing the amount of available cache in a controller.

ONTAP manages physical blocks in 4KB units. Adaptive compression uses a default compression block size of 8KB, which means data is compressed in 8KB units. This matches the 8KB block size most often used by relational databases. Compression algorithms become more efficient as more data is compressed as a single unit. A 32KB compression block size would be more space-efficient than an 8KB compression block unit. This does mean that adaptive compression using the default 8KB block size does lead to slightly lower efficiency rates, but there is also a significant benefit to using a smaller compression block size. Database workloads include a large amount of overwrite activity. Overwriting a 8KB of a compressed 32KB block of data requires reading back the entire 32KB of logical data, decompressing it, updating the required 8KB region, recompressing, and then writing the entire 32KB back to the drives. This is a very expensive operation for a storage system and is the reason some competing storage arrays based on larger compression block sizes also incur a significant performance penalty with database workloads.

The block size used by adaptive compression can be increased up to 32KB. This may improve storage efficiency and should be considered for quiescent files such as transaction logs and backup files when a substantial amount of such data is stored on the array. In some situations, active databases that use a 16KB or 32KB block size may also benefit from increasing the block size of adaptive compression to match. Consult a NetApp or partner representative for guidance on whether this is appropriate for your workload.

Compression block sizes larger than 8KB should not be used alongside deduplication on streaming backup destinations. The reason is small changes to the backed-up data affect the 32KB compression window. If the window shifts, the resulting compressed data differs across the entire file. Deduplication occurs after compression, which means the deduplication engine sees each compressed backup differently. If deduplication of streaming backups is required, only 8KB block adaptive compression should be used. Adaptive compression is preferable, because it works at a smaller block size and does not disrupt deduplication efficiency. For similar reasons, host-side compression also interferes with deduplication efficiency.

Compression alignment

Adaptive compression in a database environment requires some consideration of compression block alignment. Doing so is only a concern for data that is subject to random overwrites of very specific blocks. This approach is similar in concept to overall file system alignment, where the start of a filesystem must be aligned to a 4K device boundary and the blocksize of a filesystem must be a multiple of 4K.

For example, an 8KB write to a file is compressed only if it aligns with an 8KB boundary within the file system itself. This point means that it must fall on the first 8KB of the file, the second 8KB of the file, and so forth. The simplest way to ensure correct alignment is to use the correct LUN type, any partition created should have an offset from the start of the device that is a multiple of 8K, and use a filesystem block size that is a multiple of the database block size.

Data such as backups or transaction logs are sequentially written operations that span multiple blocks, all of which are compressed. Therefore, there is no need to consider alignment. The only I/O pattern of concern is the random overwrites of files.

Data compaction

Data compaction is a technology that improves compression efficiency. As stated previously, adaptive compression alone can provide at best 2:1 savings because it is limited to storing an 8KB I/O in a 4KB WAFL block. Compression methods with larger block sizes deliver better efficiency. However, they are not suitable for data that is subject to small block overwrites. Decompressing 32KB units of data, updating an 8KB portion, recompressing, and writing back to the drives creates overhead.

Data compaction works by allowing multiple logical blocks to be stored within physical blocks. For example, a database with highly compressible data such as text or partially full blocks may compress from 8KB to 1KB. Without compaction, that 1KB of data would still occupy an entire 4KB block. Inline data compaction allows that 1KB of compressed data to be stored in just 1KB of physical space alongside other compressed data. It is not a compression technology; it is simply a more efficient way of allocating space on the drives and therefore should not create any detectable performance effect.

The degree of savings obtained vary. Data that is already compressed or encrypted cannot generally be further compressed, and therefore such datasets do not benefit from compaction. In contrast, newly initialized datafiles that contain little more than block metadata and zeros compress up to 80:1.

Temperature sensitive storage efficiency

Temperature sensitive storage efficiency (TSSE) is available in ONTAP 9.8 and later. It relies on block access heat maps to identify infrequently accessed blocks and compress them with greater efficiency.

Deduplication

Deduplication is the removal of duplicate block sizes from a dataset. For example, if the same 4KB block existed in 10 different files, deduplication would redirect that 4KB block within all 10 files to the same 4KB physical block. The result would be a 10:1 improvement in efficiency for that data.

Data such as VMware guest boot LUNs usually deduplicate extremely well because they consist of multiple copies of the same operating system files. Efficiency of 100:1 and greater have been observed.

Some data does not contain duplicate data. For example, an Oracle block contains a header that is globally unique to the database and a trailer that is nearly unique. As a result, deduplication of an Oracle database rarely delivers more than 1% savings. Deduplication with MS SQL databases is slightly better, but unique metadata at the block level is still a limitation.

Space savings of up to 15% in databases with 16KB and large block sizes have been observed in a few cases. The initial 4KB of each block contains the globally unique header, and the final 4KB block contains the nearly unique trailer. The internal blocks are candidates for deduplication, although in practice this is almost entirely attributed to the deduplication of zeroed data.

Many competing arrays claim the ability to deduplicate databases based on the presumption that a database is copied multiple times. In this respect, NetApp deduplication could also be used, but ONTAP offers a better option: NetApp FlexClone technology. The end result is the same; multiple copies of a database that share most of the underlying physical blocks are created. Using FlexClone is much more efficient than taking the time to copy database files and then deduplicating them. It is, in effect, nonduplication rather than deduplication, because a duplicate is never created in the first place.

Efficiency and thin provisioning

Efficiency features are forms of thin provisioning. For example, a 100GB LUN occupying a 100GB volume might compress down to 50GB. There are no actual savings realized yet because the volume is still 100GB. The volume must first be reduced in size so that the space saved can be used elsewhere on the system. If later changes to the 100GB LUN result in the data becoming less compressible, then the LUN grows in size and the volume could fill up.

Thin provisioning is strongly recommended because it can simplify management while delivering a substantial improvement in usable capacity with associated cost savings. The reason is simple - database environments frequently include a lot of empty space, a large number of volumes and LUNs, and compressible data. Thick provisioning results in the reservation of space on storage for volumes and LUNs just in case they someday become 100% full and contain 100% uncompressible data. That is unlikely to ever occur. Thin provisioning allows that space to be reclaimed and used elsewhere and allows capacity management to be based on the storage system itself rather than many smaller volumes and LUNs.

Some customers prefer to use thick provisioning, either for specific workloads or generally based on established operational and procurement practices.

If a volume is thick provisioned, care must be taken to completely disable all efficiency features for that volume, including decompression and the removal of deduplication using the `sis undo` command. The volume should not appear in `volume efficiency show` output. If it does, the volume is still partially configured for efficiency features. As a result, overwrite guarantees work differently, which increases the chance that configuration oversights cause the volume to unexpectedly run out of space, resulting in database I/O errors.

Efficiency best practices

NetApp recommends the following:

AFF defaults

Volumes created on ONTAP running on an all-flash AFF system are thin provisioned with all inline efficiency features enabled. Although databases generally do not benefit from deduplication and may include uncompressible data, the default settings are nevertheless appropriate for almost all workloads. ONTAP is designed to efficiently process all types of data and I/O patterns, whether or not they result in savings. Defaults should only be changed if the reasons are fully understood and there is a benefit to deviating.

General recommendations

- If volumes and/or LUNs are not thin provisioned, you must disable all efficiency settings because using these features provides no savings and the combination of thick provisioning with space efficiency enabled can cause unexpected behavior, including out-of-space errors.
- If data is not subject to overwrites, such as with backups or database transaction logs, you can achieve greater efficiency by enabling TSSE with a low cooling period.
- Some files might contain a significant amount of uncompressible data, for example when compression is already enabled at the application level of files are encrypted. If any of these scenarios are true, consider disabling compression to allow more efficient operation on other volumes containing compressible data.
- Do not use both 32KB compression and deduplication with database backups. See the section Adaptive compression for details.

Thin provisioning

Thin provisioning for an Oracle database requires careful planning because the result is configuring more space on a storage system than is necessarily physically available. It is very much worth the effort because, when done correctly, the result is significant cost savings and improvements in manageability.

Thin provisioning comes in many forms and is integral to many features that ONTAP offers to an enterprise application environment. Thin provisioning is also closely related to efficiency technologies for the same reason: efficiency features allow more logical data to be stored than what technically exists on the storage system.

Almost any use of snapshots involves thin provisioning. For example, a typical 10TB database on NetApp storage includes around 30 days of snapshots. This arrangement results in approximately 10TB of data visible in the active file system and 300TB dedicated to snapshots. The total 310TB of storage usually resides on approximately 12TB to 15TB of space. The active database consumes 10TB, and the remaining 300TB of data only requires 2TB to 5TB of space because only the changes to the original data are stored.

Cloning is also an example of thin provisioning. A major NetApp customer created 40 clones of an 80TB

database for use by development. If all 40 developers using these clones overwrote every block in every datafile, over 3.2PB of storage would be required. In practice, turnover is low, and the collective space requirement is closer to 40TB because only changes are stored on the drives.

Space management

Some care must be taken with thin provisioning an application environment because data change rates can increase unexpectedly. For example, space consumption due to snapshots can grow rapidly if database tables are reindexed, or wide-scale patching is applied to VMware guests. A misplaced backup can write a large amount of data in a very short time. Finally, it can be difficult to recover some applications if a file system runs out of free space unexpectedly.

Fortunately, these risks can be addressed with careful configuration of `volume-autogrow` and `snapshot-autodelete` policies. As their names imply, these options enable a user to create policies that automatically clear space consumed by snapshots or grow a volume to accommodate additional data. Many options are available and needs vary by customer.

See the logical storage management documentation for a complete discussion of these features.

Fractional reservations

Fractional reserve refers to the behavior of a LUN in a volume with respect to space efficiency. When the option `fractional-reserve` is set to 100%, all data in the volume can experience 100% turnover with any data pattern without exhausting space on the volume.

For example, consider a database on a single 250GB LUN in a 1TB volume. Creating a snapshot would immediately result in the reservation of an additional 250GB of space in the volume to guarantee that the volume does not run out of space for any reason. Using fractional reserves is generally wasteful because it is extremely unlikely that every byte in the database volume would need to be overwritten. There is no reason to reserve space for an event that never happens. Still, if a customer cannot monitor space consumption in a storage system and must be certain that space never runs out, 100% fractional reservations would be required to use snapshots.

Compression and deduplication

Compression and deduplication are both forms of thin provisioning. For example, a 50TB data footprint might compress to 30TB, resulting in a savings of 20TB. For compression to yield any benefits, some of that 20TB must be used for other data, or the storage system must be purchased with less than 50TB. The result is storing more data than is technically available on the storage system. From the data point of view, there is 50TB of data, even though it occupies only 30TB on the drives.

There is always a possibility that the compressibility of a dataset changes, which would result in increased consumption of real space. This increase in consumption means that compression must be managed as with other forms of thin provisioning in terms of monitoring and using `volume-autogrow` and `snapshot-autodelete`.

Compression and deduplication are discussed in further detail in the section xref:./oracle/efficiency.html

Compression and fractional reservations

Compression is a form of thin provisioning. Fractional reservations affect the use of compression, with one important note; space is reserved in advance of the snapshot creation. Normally, fractional reserve is only important if a snapshot exists. If there is no snapshot, fractional reserve is not important. This is not the case with compression. If a LUN is created on a volume with compression, ONTAP preserves space to accommodate a snapshot. This behavior can be confusing during configuration, but it is expected.

As an example, consider a 10GB volume with a 5GB LUN that has been compressed down to 2.5GB with no snapshots. Consider these two scenarios:

- Fractional reserve = 100 results in 7.5GB utilization
- Fractional reserve = 0 results in 2.5GB utilization

The first scenario includes 2.5GB of space consumption for current data and 5GB of space to account for 100% turnover of the source in anticipation of snapshot use. The second scenario reserves no extra space.

Although this situation might seem confusing, it is unlikely to be encountered in practice. Compression implies thin provisioning, and thin provisioning in a LUN environment requires fractional reservations. It is always possible for compressed data to be overwritten by something uncompressible, which means a volume must be thin provisioned for compression to result in any savings.

NetApp recommends the following reserve configurations:

- Set `fractional-reserve` to 0 when basic capacity monitoring is in place along with `volume-autogrow` and `snapshot-autodelete`.
- Set `fractional-reserve` to 100 if there is no monitoring ability or if it is impossible to exhaust space under any circumstance.

Free space and LVM space allocation

The efficiency of thin provisioning of active LUNs in a file system environment can be lost over time as data is deleted. Unless that deleted data is either overwritten with zeros (see also ASMRU or the space is released with TRIM/UNMAP space reclamation, the "erased" data occupies more and more unallocated whitespace in the file system. Furthermore, thin provisioning of active LUNs is of limited use in many database environments because datafiles are initialized to their full size at the time of creation.

Careful planning of LVM configuration can improve efficiency and minimize the need for storage provisioning and LUN resizing. When an LVM such as Veritas VxVM or Oracle ASM is used, the underlying LUNs are divided into extents that are only used when needed. For example, if a dataset begins at 2TB in size but could grow to 10TB over time, this dataset could be placed on 10TB of thin-provisioned LUNs organized in an LVM diskgroup. It would occupy only 2TB of space at the time of creation and would only claim additional space as extents are allocated to accommodate data growth. This process is safe as long as space is monitored.

ONTAP failover/switchover

An understanding of storage takeover and switchover functions is required to ensure that Oracle database operations are not disrupted by these operations. In addition, the arguments used by takeover and switchover operations can affect data integrity if used incorrectly.

- Under normal conditions, incoming writes to a given controller are synchronously mirrored to its partner. In a NetApp MetroCluster environment, writes are also mirrored to a remote controller. Until a write is stored in nonvolatile media in all locations, it is not acknowledged to the host application.
- The media storing the write data is called nonvolatile memory or NVMEM. It is also sometimes referred to nonvolatile random- access memory (NVRAM), and it can be thought of as a write cache although it functions as a journal. In a normal operation, the data from NVMEM is not read; it is only used to protect data in the event of a software or hardware failure. When data is written to drives, the data is transferred from the RAM in the system, not from NVMEM.

- During a takeover operation, one node in a high availability (HA) pair takes over the operations from its partner. A switchover is essentially the same, but it applies to MetroCluster configurations in which a remote node takes over the functions of a local node.

During routine maintenance operations, a storage takeover or switchover operation should be transparent, other than for a potential brief pause in operations as the network paths change. Networking can be complicated, however, and it is easy to make errors, so NetApp strongly recommends testing takeover and switchover operations thoroughly before putting a storage system into production. Doing so is the only way to be sure that all network paths are configured correctly. In a SAN environment, carefully check the output of the command `sanlun lun show -p` to make sure that all expected primary and secondary paths are available.

Care must be taken when issuing a forced takeover or switchover. Forcing a change to storage configuration with these options means that the state of the controller that owns the drives is disregarded and the alternative node forcibly takes control of the drives. Incorrect forcing of a takeover can result in data loss or corruption. This is because a forced takeover or switchover can discard the contents of NVMEM. After the takeover or switchover is complete, the loss of that data means that the data stored on the drives might revert to a slightly older state from the point of view of the database.

A forced takeover with a normal HA pair should rarely be required. In almost all failure scenarios, a node shut downs and informs the partner so that an automatic failover takes place. There are some edge cases, such as a rolling failure in which the interconnect between nodes is lost and then one controller is lost, in which a forced takeover is required. In such a situation, the mirroring between nodes is lost before the controller failure, which means that the surviving controller would have no longer has a copy of the writes in progress. The takeover then needs to be forced, which means that data potentially is lost.

The same logic applies to a MetroCluster switchover. In normal conditions, a switchover is nearly transparent. However, a disaster can result in a loss of connectivity between the surviving site and the disaster site. From the point of view of the surviving site, the problem could be nothing more than an interruption in connectivity between sites, and the original site might still be processing data. If a node cannot verify the state of the primary controller, only a forced switchover is possible.

NetApp recommends taking the following precautions:

- Be very careful to not accidentally force a takeover or a switchover. Normally, forcing should not be required, and forcing the change can cause data loss.
- If a forced takeover or switchover is required, make sure that the applications are shut down, all file systems are dismounted and logical volume manager (LVM) volume groups are varyoffed. ASM diskgroups must be unmounted.
- In the event of a forced MetroCluster switchover, fence off the failed node from all surviving storage resources. For more information, see the MetroCluster Management and Disaster Recovery Guide for the relevant version of ONTAP.

MetroCluster and multiple aggregates

MetroCluster is a synchronous replication technology that switches to asynchronous mode if connectivity is interrupted. This is the most common request from customers, because guaranteed synchronous replication means that interruption in site connectivity leads to a complete stall of database I/O, taking the database out of service.

With MetroCluster, aggregates rapidly resynchronize after connectivity is restored. Unlike other storage technologies, MetroCluster should never require a complete remirroring after site failure. Only delta changes must be shipped.

In datasets that span aggregates, there is a small risk that additional data recovery steps would be required in a rolling disaster scenario. Specifically, if (a) connectivity between sites is interrupted, (b) connectivity is restored, (c) the aggregates reach a state in which some are synchronized and some are not, and then (d) the primary site is lost, the result is a surviving site in which the aggregates are not synchronized with one another. If this happens, parts of the dataset are synchronized with one another and it is not possible to bring up applications, databases, or datastores without recovery. If a dataset spans aggregates, NetApp strongly recommends leveraging snapshot-based backups with one of the many available tools to verify rapid recoverability in this unusual scenario.

ONTAP configuration on ASA r2 systems

RAID

RAID refers to the use of parity-based redundancy to protect data against drive failure. ASA r2 uses the same ONTAP RAID technologies as AFF and FAS systems, ensuring robust protection against multiple disk failures.

ONTAP performs RAID configuration automatically for ASA r2 systems. This is a core component of the simplified storage management experience introduced with the ASA r2 personality.

Key details regarding automatic RAID configuration on ASA r2 include:

- Storage Availability Zones (SAZ): Instead of manually managing traditional aggregates and RAID groups, ASA r2 uses Storage Availability Zones (SAZs). These are shared, RAID-protected pools of disks for an HA pair, where both nodes have full access to the same storage.
- Automatic Placement: When a storage unit (LUN or NVMe namespace) is created, ONTAP automatically creates a volume within the SAZ and places it for optimal performance and capacity balance.
- No Manual Aggregate Management: Traditional aggregate and RAID group management commands are not supported on ASA r2. This eliminates the need for administrators to manually plan RAID group sizes, parity disks, or node assignments.
- Simplified Provisioning: Provisioning is handled via System Manager or simplified CLI commands that focus on storage units rather than the underlying physical RAID layout.
- Workload Rebalancing: Beginning with 2025 releases (ONTAP 9.17.1), ONTAP automatically rebalances workloads between nodes in the HA pair to ensure performance and space utilization remain balanced without manual intervention.

ASA r2 automatically uses ONTAP's default RAID technologies: RAID DP for most configurations and RAID-TEC for very large SSD pools. This eliminates the need for manual RAID selection. These parity-based RAID levels provide better storage efficiency and reliability than mirroring, which older Oracle best practices often recommend but is not relevant for ASA r2. ONTAP avoids the traditional RAID write penalty through WAFL integration, ensuring optimal performance for Oracle workloads such as redo logging and random data-file writes. Combined with automated RAID management and Storage Availability Zones, ASA r2 delivers high availability and enterprise-grade protection for Oracle databases.

Capacity management

Managing a database or other enterprise application with predictable, manageable, high performance enterprise storage requires some free space on the drives for data and metadata management. The amount of free space required depends on the type of drive used, and business processes.

ASA r2 uses Storage Availability Zones (SAZ) instead of aggregates, but the principle remains the same: free space includes any physical capacity not consumed by actual data, snapshots, or system overhead. Thin provisioning must also be considered—logical allocations do not reflect true physical usage.

NetApp recommendations for ASA r2 storage systems used for enterprise applications are as follows:

SSD pools in ASA r2 systems

> **NetApp recommends** maintaining a minimum of 10% free physical space in ASA r2 environments. This guideline applies to SSD-only pools used by ASA r2 systems and includes all unused space within the SAZ and storage units. Logical space is unimportant; the focus is on the actual free physical space available for data storage.

While ASA r2 can sustain high utilization without performance degradation, operating near full capacity increases the risk of space exhaustion and administrative overhead when expanding storage. Running at over 90% utilization may not impact performance but can complicate management and delay provisioning of additional drives.

ASA r2 systems support storage units up to 128TB and SAZ sizes up to 2PB per HA pair, with ONTAP automatically balancing capacity across nodes. Monitoring utilization at the cluster, SAZ, and storage unit levels is essential to ensure adequate free space for snapshots, thin-provisioned workloads, and future growth. If capacity approaches critical thresholds (~ 90% utilization), additional SSDs should be added in groups (minimum six drives) to maintain performance and resiliency.

Storage Virtual Machines

Oracle database storage management on ASA r2 systems is also centralized on a Storage Virtual Machine (SVM), known as a vserver at the ONTAP CLI.

An SVM is the fundamental unit of storage provisioning and security in ONTAP, similar to a guest VM on a VMware ESX server. When ONTAP is first installed on ASA r2, it has no data-serving capabilities until an SVM is created. The SVM defines the personality and data services for the SAN environment.

ASA r2 systems use a SAN-only ONTAP personality, which is streamlined to support block protocols (FC, iSCSI, NVMe/FC, NVMe/TCP) and removes NAS-related features. This simplifies management and ensures that all SVM configurations are optimized for SAN workloads. Unlike AFF/FAS systems, ASA r2 does not expose options for NAS services such as home directories or NFS shares.

When a cluster is created, ASA r2 automatically provisions a default data SVM named svm1 with SAN protocols enabled. This SVM is ready for block storage operations without requiring manual configuration of protocol services. By default, IP data LIFs in this SVM support iSCSI and NVMe/TCP protocols and use the default-data-blocks service policy, which simplifies initial setup for SAN workloads. Administrators can later create additional SVMs or customize LIF configurations based on performance, security, or multi-tenant requirements.

> Logical interfaces (LIFs) for SAN protocols should be designed based on performance and availability requirements. ASA r2 supports iSCSI, FC, and NVMe LIFs, but note that automatic iSCSI LIF failover is not enabled by default because ASA r2 uses shared networking for NVMe and SCSI hosts. To enable automatic failover, create iSCSI-only LIFs.

SVMs

As with other ONTAP platforms, there is no official best practice for the number of SVMs to create; the decision depends on management and security requirements.

Most customers operate a single primary SVM for day-to-day operations and create additional SVMs for special needs, such as:

- A dedicated SVM for a critical business database managed by a specialist team
- An SVM for a development group with delegated administrative control
- An SVM for sensitive data requiring restricted administrative access

In multi-tenant environments, each tenant can be assigned a dedicated SVM. The limit for the number of SVMs and LIFs per cluster, HA pair, and node are dependent on the protocol being used, the node model, and the version of ONTAP. Consult the NetApp Hardware Universe for these limits.

ASA r2 supports up to 256 SVMs per cluster and per HA pair starting with ONTAP 9.18.1 (previously 32 in earlier releases).

Performance management with ONTAP QoS on ASA r2 systems

Safely and efficiently managing multiple Oracle databases on ASA r2 requires an effective QoS strategy. This is especially important because ASA r2 systems are all-flash SAN platforms designed for extremely high performance and workload consolidation.

A relatively small number of SSDs can saturate even the most powerful controllers, so QoS controls are essential to ensure predictable performance across multiple workloads.
As a reference, ASA r2 systems such as the ASA A1K or A90 can deliver hundreds of thousands to over a million IOPS with sub-millisecond latency. Very few single workloads would consume this level of performance, so full utilization typically involves hosting multiple databases or applications. Doing this safely requires QoS policies to prevent resource contention.

ONTAP QoS on ASA r2 works the same way as on AFF/FAS systems, with two primary types of controls: IOPS and bandwidth. QoS controls can be applied to SVMs and LUNs.

IOPS QoS

IOPS-based QoS limits the total IOPS for a given resource. In ASA r2, QoS policies can be applied at the SVM level and to individual storage objects such as LUNs. When a workload reaches its IOPS limit, additional I/O requests queue for tokens, which introduces latency. This is expected behavior and prevents any single workload from monopolizing system resources.

Be cautious when applying QoS controls to database transaction/redo log data. These workloads are bursty, and a QoS limit that seems reasonable for average activity may be too low for peak bursts, causing severe performance issues. In general, redo and archive logging should not be limited by QoS.

Bandwidth QoS

Bandwidth-based QoS limits throughput in Mbps. This is useful when workloads perform large block reads or writes, such as full table scans or backup operations, which consume significant bandwidth but relatively few IOPS. Combining IOPS and bandwidth limits can provide more granular control.

Minimum/guaranteed QoS

Minimum QoS policies reserve performance for critical workloads. For example, in a mixed environment with production and development databases, apply maximum QoS to development workloads and minimum QoS to production workloads to ensure predictable performance.

Adaptive QoS

Adaptive QoS adjusts limits based on the size of the storage object. While rarely used for databases (because size does not correlate to performance needs), it can be useful for virtualization workloads where performance requirements scale with capacity.

Efficiency

ONTAP space efficiency features are fully supported and optimized for ASA r2 systems. In almost all cases, the best approach is to leave the defaults in place with all efficiency features enabled.

ASA r2 systems are all-flash SAN platforms, so efficiency technologies such as compression, compaction, and deduplication are critical for maximizing usable capacity and reducing costs.

Compression

Compression reduces space requirements by encoding patterns in data. With SSD-based ASA r2 systems, compression delivers significant savings because flash eliminates the need for overprovisioning for performance. ONTAP adaptive compression is enabled by default and has been thoroughly tested with enterprise workloads, including Oracle databases, with no measurable performance impact—even in environments where latency is measured in microseconds. In some cases, performance improves because compressed data occupies less cache space.

Temperature-sensitive storage efficiency (TSSE) is not applied on ASA r2 systems. On ASA r2 systems, compression is not based on hot (frequently accessed) data or cold (infrequently accessed) data. Compression begins without waiting for data to become cold.

Adaptive compression

Adaptive compression uses an 8KB block size by default, matching the block size commonly used by relational databases. Larger block sizes (16KB or 32KB) can improve efficiency for sequential data such as transaction logs or backups but should be used cautiously for active databases to avoid overhead during overwrites.

Block size can be increased up to 32KB for quiescent files such as logs or backups. Consult NetApp guidance before changing defaults.

Do not use 32KB compression with deduplication for streaming backups. Use 8KB compression to maintain deduplication efficiency.

Compression alignment

Compression alignment matters for random overwrites. Ensure correct LUN type, partition offset (multiple of 8KB), and filesystem block size aligned to database block size. Sequential data such as backups or logs does not require alignment considerations.

Data compaction

Compaction complements compression by allowing multiple compressed blocks to share the same physical block. For example, if an 8KB block compresses to 1KB, compaction ensures that the remaining space is not wasted. This feature is inline and does not introduce performance penalties.

Deduplication

Deduplication removes duplicate blocks across datasets. While Oracle databases typically yield minimal deduplication savings due to unique block headers and trailers, ONTAP deduplication can still reclaim space from zeroed blocks and repeated patterns.

Efficiency and thin provisioning

ASA r2 systems use thin provisioning by default. Efficiency features complement thin provisioning to maximize usable capacity.

Storage units are always thinly provisioned on ASA r2 storage systems. Thick provisioning is not supported.

QuickAssist Technology (QAT)

In NetApp ASA r2 platforms, Intel QuickAssist Technology (QAT) provides hardware-accelerated efficiency that differs significantly from software-based Temperature-Sensitive Storage Efficiency (TSSE) without QAT.

QAT with hardware acceleration:

- Offloads compression and encryption tasks from CPU cores.
- Enables immediate, inline efficiency for both hot (frequently accessed) and cold (infrequently accessed) data.
- Significantly reduces CPU overhead.
- Delivers higher throughput and lower latency.
- Improves scalability for performance-sensitive operations such as TLS and VPN encryption.

TSSE without QAT:

- Relies on CPU-driven processes for efficiency operations.
- Applies efficiency only to cold data after a delay.
- Consumes more CPU resources.
- Limits overall performance compared to QAT-accelerated systems.

Modern ASA r2 systems therefore deliver faster, hardware-accelerated efficiency and better system utilization than older TSSE-only platforms.

Efficiency best practices for ASA r2

NetApp recommends the following:

ASA r2 defaults

Storage units created on ONTAP running on ASA r2 systems are thin provisioned with all inline efficiency

features enabled by default, including compression, compaction, and deduplication. Although Oracle databases generally do not benefit significantly from deduplication and may include uncompressible data, these defaults are appropriate for almost all workloads. ONTAP is designed to efficiently process all types of data and I/O patterns, whether or not they result in savings. Defaults should only be changed if the reasons are fully understood and there is a clear benefit to deviating.

General recommendations

- Disable Compression for Encrypted or App-Compressed Data: If files are already compressed at the application level or encrypted, disable compression to optimize performance and allow more efficient operation on other storage units.
- Avoid Combining Large Compression Blocks with Deduplication: Do not use both 32KB compression and deduplication for database backups. For streaming backups, use 8KB compression to maintain deduplication efficiency.
- Monitor Efficiency Savings: Use ONTAP tools (System Manager, Active IQ) to track actual space savings and adjust policies if needed.

Thin provisioning

Thin provisioning for an Oracle database on ASA r2 requires careful planning because it involves configuring more logical space than is physically available. When implemented correctly, thin provisioning delivers significant cost savings and improved manageability.

Thin provisioning is integral to ASA r2 and closely related to ONTAP efficiency technologies because both allow more logical data to be stored than the physical capacity on the system. ASA r2 systems are SAN-only, and thin provisioning applies to storage units and LUNs within Storage Availability Zones (SAZ).

ASA r2 storage units are thin provisioned by default.

Almost any use of snapshots involves thin provisioning. For example, a typical 10 TiB database with 30 days of snapshots might appear as 310 TiB of logical data, but only 12 TiB to 15 TiB of physical space is consumed because snapshots store only changed blocks.

Similarly, cloning is another form of thin provisioning. A development environment with 40 clones of an 80 TiB database would require 3.2 PiB if fully written, but in practice consumes far less because only changes are stored.

Space management

Some care must be taken with thin provisioning in an application environment because data change rates can increase unexpectedly. For example, space consumption due to snapshots can grow rapidly if database tables are reindexed, or wide-scale patching is applied to VMware guests. A misplaced backup can write a large amount of data in a very short time. Finally, it can be difficult to recover some applications if a LUN runs out of free space unexpectedly.

In ASA r2, these risks are mitigated through **thin provisioning**, **proactive monitoring**, and **LUN resize policies**, rather than ONTAP features like volume-autogrow or snapshot-autodelete. Administrators should:

- Enable thin provisioning on LUNs (`space-reserve disabled`) - this is the default setting in ASA r2
- Monitor capacity using System Manager alerts or API-based automation
- Use scheduled or scripted LUN resize to accommodate growth

- Configure snapshot reserve and automatic snapshot deletion via System Manager (GUI)

> Careful planning of space thresholds and automation scripts is essential because ASA r2 does not support automatic volume growth or CLI-driven snapshot deletion.

ASA r2 does not use fractional reserve settings because it is a SAN-only architecture that abstracts WAFL-based volume options. Instead, space efficiency and overwrite protection are managed at the LUN level. For example, if you have a 250 GiB LUN provisioned from a storage unit, snapshots consume space based on actual block changes rather than reserving an equal amount of space upfront. This eliminates the need for large static reservations, which were common in traditional ONTAP environments using fractional reserve.

> If guaranteed overwrite protection is required and monitoring is not feasible, administrators should provision sufficient capacity in the storage unit and set snapshot reserve appropriately. However, ASA r2's design makes fractional reserve unnecessary for most workloads.

Compression and deduplication

Compression and deduplication in ASA r2 are space efficiency technologies, not traditional thin provisioning mechanisms. These features reduce the physical storage footprint by eliminating redundant data and compressing blocks, allowing more logical data to be stored than the raw capacity would otherwise permit.

For example, a 50 TiB dataset might compress to 30 TiB, saving 20 TiB of physical space. From the application perspective, there is still 50 TiB of data, even though it occupies only 30 TiB on disk.

> The compressibility of a dataset can change over time, which may increase physical space consumption. Therefore, compression and deduplication must be managed proactively through monitoring and capacity planning.

Free space and LVM space allocation

Thin provisioning in ASA r2 environments can lose efficiency over time if deleted blocks are not reclaimed. Unless space is released using TRIM/UNMAP or overwritten with zeros (via ASMRU - Automatic Space Management and Reclamation Utility), deleted data continues to consume physical capacity. In many Oracle database environments, thin provisioning offers limited benefit because datafiles are typically pre-allocated to their full size during creation.

Careful planning of LVM configuration can improve efficiency and minimize the need for storage provisioning and LUN resizing. When an LVM such as Veritas VxVM or Oracle ASM is used, the underlying LUNs are divided into extents that are only used when needed. For example, if a dataset begins at 2 TiB in size but could grow to 10 TiB over time, this dataset could be placed on 10 TiB of thin-provisioned LUNs organized in an LVM diskgroup. It would occupy only 2 TiB of space at the time of creation and would only claim additional space as extents are allocated to accommodate data growth. This process is safe as long as space is monitored.

ONTAP failover

An understanding of storage takeover functions is required to ensure that Oracle database operations are not disrupted during these operations. In addition, the arguments used by takeover operations can affect data integrity if used incorrectly.

Under normal conditions, incoming writes to a given controller are synchronously mirrored to its HA partner. In an ASA r2 environment with SnapMirror Active Sync (SM-as), writes are also mirrored to a remote controller at the secondary site. Until a write is stored in non-volatile media in all locations, it is not acknowledged to the

host application.

The media storing the write data is called non-volatile memory (NVMEM). It is sometimes referred to as non-volatile random-access memory (NVRAM) and can be thought of as a write journal rather than a cache. During normal operation, data from NVMEM is not read; it is only used to protect data in the event of a software or hardware failure. When data is written to drives, the data is transferred from system RAM, not from NVMEM.

During a takeover operation, one node in an HA pair takes over the operations from its partner. In ASA r2, switchover is not applicable because MetroCluster is not supported; instead, SnapMirror Active Sync provides site-level redundancy. Storage takeover operations during routine maintenance should be transparent, other than a brief pause in operations as network paths change. Networking can be complex, and errors are easy to make, so NetApp strongly recommends testing takeover operations thoroughly before putting a storage system into production. Doing so is the only way to ensure that all network paths are configured correctly.
In a SAN environment, verify path status using the command `sanlun lun show -p` or the operating system's native multipathing tools to ensure all expected paths are available. ASA r2 systems provide all active optimized paths for LUNs, and customers using NVMe namespaces should rely on OS-native tools, as NVMe paths are not covered by sanlun.

Care must be taken when issuing a forced takeover. Forcing a change to storage configuration means that the state of the controller that owns the drives is disregarded and the alternative node forcibly takes control of the drives. Incorrect forcing of a takeover can result in data loss or corruption because a forced takeover can discard the contents of NVMEM. After the takeover is complete, the loss of that data means that the data stored on the drives might revert to a slightly older state from the point of view of the database.

A forced takeover with a normal HA pair should rarely be required. In almost all failure scenarios, a node shuts down and informs the partner so that an automatic failover takes place. There are some edge cases, such as a rolling failure in which the interconnect between nodes is lost and then one controller fails, in which a forced takeover is required. In such a situation, the mirroring between nodes is lost before the controller failure, which means that the surviving controller no longer has a copy of the writes in progress. The takeover then needs to be forced, which means that data potentially is lost.

NetApp recommends taking the following precautions:

- Be very careful to not accidentally force a takeover. Normally, forcing should not be required, and forcing the change can cause data loss.
- If a forced takeover is required, make sure that the applications are shut down, all file systems are dismounted, and logical volume manager (LVM) volume groups are varyoffed. ASM diskgroups must be unmounted.
- In the event of a site-level failure when using SM-as, the ONTAP Mediator assisted automatic unplanned failover will be initiated on the surviving cluster, resulting in a brief I/O pause and then database transitions will continue from the surviving cluster. For more information, see the SnapMirror active sync on ASA r2 systems for detailed configuration steps.

Database configuration with AFF/FAS systems

Block sizes

ONTAP internally uses a variable block size, which means Oracle databases can be configured with any block size desired. However, filesystem block sizes can affect performance and in some cases a larger redo block size can improve performance.

Datafile block sizes

Some OSs offer a choice of file system block sizes. For file systems supporting Oracle datafiles, the block size should be 8KB when compression is used. When compression is not required, a block size of either 8KB or 4KB can be used.

If a datafile is placed on a file system with a 512-byte block, misaligned files are possible. The LUN and the file system might be properly aligned based on NetApp recommendations, but the file I/O would be misaligned. Such a misalignment would cause severe performance problems.

File systems supporting redo logs must use a block size that is a multiple of the redo block size. This generally requires that both the redo log file system and the redo log itself use a block size of 512 bytes.

Redo block sizes

At very high redo rates, it is possible that 4KB block sizes would perform better because high redo rates allow I/O to be performed in fewer and more efficient operations. If redo rates are greater than 50MBps, consider testing a 4KB block size.

A few customer problems have been identified with databases using redo logs with a 512-byte block size on a file system with a 4KB block size and many very small transactions. The overhead involved in applying multiple 512-byte changes to a single 4KB file system block led to performance problems that were resolved by changing the file system to use a block size of 512 bytes.

NetApp recommends that you do not change the redo block size unless advised by a relevant customer support or professional services organization or the change is based on official product documentation.

db_file_multiblock_read_count

The `db_file_multiblock_read_count` parameter controls the maximum number of Oracle database blocks that Oracle reads as a single operation during sequential I/O.

This parameter does not, however, affect the number of blocks that Oracle reads during any and all read operations, nor does it affect random I/O. Only the block size of sequential I/O is affected.

Oracle recommends that the user leave this parameter unset. Doing so allows the database software to automatically set the optimum value. This generally means that this parameter is set to a value that yields an I/O size of 1MB. For example, a 1MB read of 8KB blocks would require 128 blocks to be read, and the default value for this parameter would therefore be 128.

Most database performance problems observed by NetApp at customer sites involve an incorrect setting for this parameter. There were valid reasons to change this value with Oracle versions 8 and 9. As a result, the parameter might be unknowingly present in `init.ora` files because the database was upgraded in place to Oracle 10 and later. A legacy setting of 8 or 16, compared to a default value of 128, significantly damages sequential I/O performance.

NetApp recommends setting the `db_file_multiblock_read_count` parameter should not be present in the `init.ora` file. NetApp has never encountered a situation in which changing this parameter improved performance, but there are many cases in which it caused clear damage to sequential I/O throughput.

filesystemio_options

The Oracle initialization parameter `filesystemio_options` controls the use of asynchronous and direct I/O.

Contrary to common belief, asynchronous and direct I/O are not mutually exclusive. NetApp has observed that this parameter is frequently misconfigured in customer environments, and this misconfiguration is directly responsible for many performance problems.

Asynchronous I/O means that Oracle I/O operations can be parallelized. Before the availability of asynchronous I/O on various OSs, users configured numerous dbwriter processes and changed the server process configuration. With asynchronous I/O, the OS itself performs I/O on behalf of the database software in a highly efficient and parallel manner. This process does not place data at risk, and critical operations, such as Oracle redo logging, are still performed synchronously.

Direct I/O bypasses the OS buffer cache. I/O on a UNIX system ordinarily flows through the OS buffer cache. This is useful for applications that do not maintain an internal cache, but Oracle has its own buffer cache within the SGA. In almost all cases, it is better to enable direct I/O and allocate server RAM to the SGA rather than to rely on the OS buffer cache. The Oracle SGA uses the memory more efficiently. In addition, when I/O flows through the OS buffer, it is subject to additional processing, which increases latencies. The increased latencies are especially noticeable with heavy write I/O when low latency is a critical requirement.

The options for `filesystemio_options` are:

- **async.** Oracle submits I/O requests to the OS for processing. This process allows Oracle to perform other work rather than waiting for I/O completion and thus increases I/O parallelization.
- **directio.** Oracle performs I/O directly against physical files rather than routing I/O through the host OS cache.
- **none.** Oracle uses synchronous and buffered I/O. In this configuration, the choice between shared and dedicated server processes and the number of dbwriters are more important.
- **setall.** Oracle uses both asynchronous and direct I/O. In almost all cases, the use of `setall` is optimal.

The `filesystemio_options` parameter has no effect in DNFS and ASM environments. The use of DNFS or ASM automatically results in the use of both asynchronous and direct I/O.

Some customers have encountered asynchronous I/O problems in the past, especially with previous Red Hat Enterprise Linux 4 (RHEL4) releases. Some out-of-date advice on the internet still suggests avoiding asynchronous IO because of out-of-date information. Asynchronous I/O is stable on all current OSs. There is no reason to disable it, absent a known bug with the OS.

If a database has been using buffered I/O, a switch to direct I/O might also warrant a change in the SGA size. Disabling buffered I/O eliminates the performance benefit that the host OS cache provides for the database. Adding RAM back to the SGA repairs this problem. The net result should be an improvement in I/O performance.

Although it is almost always better to use RAM for the Oracle SGA than for OS buffer caching, it might be impossible to determine the best value. For example, it might be preferable to use buffered I/O with very small SGA sizes on a database server with many intermittently active Oracle instances. This arrangement allows the flexible use of the remaining free RAM on the OS by all running database instances. This is a highly unusual situation, but it has been observed at some customer sites.

NetApp recommends setting `filesystemio_options` to `setall`, but be aware that under some circumstances the loss of the host buffer cache might require an increase in the Oracle SGA.

RAC timeouts

Oracle RAC is a clusterware product with several types of internal heartbeat processes that monitor the health of the cluster.

The information in the misscount section includes critical information for Oracle RAC environments using networked storage, and in many cases the default Oracle RAC settings will need to be changed to ensure the RAC cluster survives network path changes and storage failover/switchover operations.

disktimeout

The primary storage-related RAC parameter is `disktimeout`. This parameter controls the threshold within which voting file I/O must complete. If the `disktimeout` parameter is exceeded, then the RAC node is evicted from the cluster. The default for this parameter is 200. This value should be sufficient for standard storage takeover and giveback procedures.

NetApp strongly recommends testing RAC configurations thoroughly before placing them into production because many factors affect a takeover or giveback. In addition to the time required for storage failover to complete, additional time is also required for Link Aggregation Control Protocol (LACP) changes to propagate. Also, SAN multipathing software must detect an I/O timeout and retry on an alternate path. If a database is extremely active, a large amount of I/O must be queued and retried before voting disk I/O is processed.

If an actual storage takeover or giveback cannot be performed, the effect can be simulated with cable pull tests on the database server.

NetApp recommends the following:

- Leaving the `disktimeout` parameter at the default value of 200.
- Always test a RAC configuration thoroughly.

misscount

The `misscount` parameter normally affects only the network heartbeat between RAC nodes. The default is 30 seconds. If the grid binaries are on a storage array or the OS boot drive is not local, this parameter might become important. This includes hosts with boot drives located on an FC SAN, NFS-booted OSs, and boot drives located on virtualization datastores such as a VMDK file.

If access to a boot drive is interrupted by a storage takeover or giveback, it is possible that the grid binary location or the entire OS temporarily hangs. The time required for ONTAP to complete the storage operation and for the OS to change paths and resume I/O might exceed the `misscount` threshold. As a result, a node immediately evicts after connectivity to the boot LUN or grid binaries is restored. In most cases, the eviction and subsequent reboot occur with no logging messages to indicate the reason for the reboot. Not all configurations are affected, so test any SAN-booting, NFS-booting, or datastore-based host in a RAC environment so that RAC remains stable if communication to the boot drive is interrupted.

In the case of nonlocal boot drives or a nonlocal file system hosting `grid` binaries, the `misscount` will need to

be changed to match `disktimeout`. If this parameter is changed, conduct further testing to also identify any effects on RAC behavior, such as node failover time.

NetApp recommends the following:

- Leave the `misscount` parameter at the default value of 30 unless one of the following conditions applies:
  - `grid` binaries are located on a network-attached drive, including NFS, iSCSI, FC, and datastore-based drives.
  - The OS is SAN booted.
- In such cases, evaluate the effect of network interruptions that affect access to OS or `GRID_HOME` file systems. In some cases, such interruptions cause the Oracle RAC daemons to stall, which can lead to a `misscount`-based timeout and eviction. The timeout defaults to 27 seconds, which is the value of `misscount` minus `reboottime`. In such cases, increase `misscount` to 200 to match `disktimeout`.

Database configuration with ASA r2 systems

Block sizes

ONTAP internally uses a variable block size, which means Oracle databases can be configured with any block size desired. However, filesystem block sizes can affect performance, and in some cases, a larger redo block size can improve performance.

ASA r2 does not introduce any changes to Oracle block size recommendations compared to AFF/FAS systems. ONTAP behavior remains consistent across all platforms.

Datafile block sizes

Some OSs offer a choice of file system block sizes. For file systems supporting Oracle datafiles, the block size should be 8KB when compression is used. When compression is not required, a block size of either 8KB or 4KB can be used.

If a datafile is placed on a file system with a 512-byte block, misaligned files are possible. The LUN and the file system might be properly aligned based on NetApp recommendations, but the file I/O would be misaligned. Such a misalignment would cause severe performance problems.

Redo block sizes

File systems supporting redo logs must use a block size that is a multiple of the redo block size. This generally requires that both the redo log file system and the redo log itself use a block size of 512 bytes.

At very high redo rates, it is possible that 4KB block sizes would perform better because high redo rates allow I/O to be performed in fewer and more efficient operations. If redo rates are greater than 50MBps, consider testing a 4KB block size.

A few customer problems have been identified with databases using redo logs with a 512-byte block size on a file system with a 4KB block size and many very small transactions. The overhead involved in applying multiple 512-byte changes to a single 4KB file system block led to performance problems that were resolved by changing the file system to use a block size of 512 bytes.

NetApp recommends that you do not change the redo block size unless advised by a relevant customer support or professional services organization or the change is based on official product documentation.

db_file_multiblock_read_count

The `db_file_multiblock_read_count` parameter controls the maximum number of Oracle database blocks that Oracle reads as a single operation during sequential I/O.

There are no changes in recommendations compared to AFF/FAS systems. ONTAP behavior and Oracle best practices remain identical across ASA r2, AFF, and FAS platforms.

This parameter does not, however, affect the number of blocks that Oracle reads during any and all read operations, nor does it affect random I/O. Only the block size of sequential I/O is affected.

Oracle recommends that the user leave this parameter unset. Doing so allows the database software to automatically set the optimum value. This generally means that this parameter is set to a value that yields an I/O size of 1MB. For example, a 1MB read of 8KB blocks would require 128 blocks to be read, and the default value for this parameter would therefore be 128.

Most database performance problems observed by NetApp at customer sites involve an incorrect setting for this parameter. There were valid reasons to change this value with Oracle versions 8 and 9. As a result, the parameter might be unknowingly present in `init.ora` files because the database was upgraded in place to Oracle 10 and later. A legacy setting of 8 or 16, compared to a default value of 128, significantly damages sequential I/O performance.

NetApp recommends setting the `db_file_multiblock_read_count` parameter should not be present in the `init.ora` file. NetApp has never encountered a situation in which changing this parameter improved performance, but there are many cases in which it caused clear damage to sequential I/O throughput.

filesystemio_options

The Oracle initialization parameter `filesystemio_options` controls the use of asynchronous and direct I/O.

The behavior and recommendations for filesystemio_options on ASA r2 are identical to AFF/FAS systems because the parameter is Oracle-specific and not dependent on the storage platform. ASA r2 uses ONTAP like AFF/FAS, so the same best practices apply.

Contrary to common belief, asynchronous and direct I/O are not mutually exclusive. NetApp has observed that this parameter is frequently misconfigured in customer environments, and this misconfiguration is directly responsible for many performance problems.

Asynchronous I/O means that Oracle I/O operations can be parallelized. Before the availability of asynchronous I/O on various OSs, users configured numerous dbwriter processes and changed the server process configuration. With asynchronous I/O, the OS itself performs I/O on behalf of the database software in a highly efficient and parallel manner. This process does not place data at risk, and critical operations, such as Oracle redo logging, are still performed synchronously.

Direct I/O bypasses the OS buffer cache. I/O on a UNIX system ordinarily flows through the OS buffer cache. This is useful for applications that do not maintain an internal cache, but Oracle has its own buffer cache within

the SGA. In almost all cases, it is better to enable direct I/O and allocate server RAM to the SGA rather than to rely on the OS buffer cache. The Oracle SGA uses the memory more efficiently. In addition, when I/O flows through the OS buffer, it is subject to additional processing, which increases latencies. The increased latencies are especially noticeable with heavy write I/O when low latency is a critical requirement.

The options for `filesystemio_options` are:

- **async.** Oracle submits I/O requests to the OS for processing. This process allows Oracle to perform other work rather than waiting for I/O completion and thus increases I/O parallelization.
- **directio.** Oracle performs I/O directly against physical files rather than routing I/O through the host OS cache.
- **none.** Oracle uses synchronous and buffered I/O. In this configuration, the choice between shared and dedicated server processes and the number of dbwriters are more important.
- **setall.** Oracle uses both asynchronous and direct I/O. In almost all cases, the use of `setall` is optimal.

In ASM environments, Oracle automatically uses direct I/O and asynchronous I/O for ASM-managed disks, so `filesystemio_options` has no effect on ASM disk groups. For non-ASM deployments (e.g., file systems on SAN LUNs), set: `filesystemio_options = setall`. This enables both asynchronous and direct I/O for optimal performance.

Some older operating systems had issues with asynchronous I/O, which led to outdated advice suggesting it should be avoided. However, asynchronous I/O is stable and fully supported on all current operating systems. There is no reason to disable it unless a specific OS bug is identified.

If a database has been using buffered I/O, a switch to direct I/O might also warrant a change in the SGA size. Disabling buffered I/O eliminates the performance benefit that the host OS cache provides for the database. Adding RAM back to the SGA repairs this problem. The net result should be an improvement in I/O performance.

Although it is almost always better to use RAM for the Oracle SGA than for OS buffer caching, it might be impossible to determine the best value. For example, it might be preferable to use buffered I/O with very small SGA sizes on a database server with many intermittently active Oracle instances. This arrangement allows the flexible use of the remaining free RAM on the OS by all running database instances. This is a highly unusual situation, but it has been observed at some customer sites.

NetApp recommends setting `filesystemio_options` to `setall`, but be aware that under some circumstances the loss of the host buffer cache might require an increase in the Oracle SGA. ASA r2 systems are optimized for SAN workloads with low latency, so using setall aligns perfectly with ASA's design for high-performance Oracle deployments.

RAC timeouts

Oracle RAC is a clusterware product with several types of internal heartbeat processes that monitor the health of the cluster.

ASA r2 systems use ONTAP just like AFF/FAS, so the same principles apply for Oracle RAC timeout parameters. There are no ASA-specific changes to disktimeout or misscount recommendations. However, ASA r2 is optimized for SAN workloads and low-latency failover, which makes these best practices even more critical.

The information in the misscount section includes critical information for Oracle RAC environments using networked storage, and in many cases the default Oracle RAC settings will need to be changed to ensure the RAC cluster survives network path changes and storage failover operations.

disktimeout

The primary storage-related RAC parameter is `disktimeout`. This parameter controls the threshold within which voting file I/O must complete. If the `disktimeout` parameter is exceeded, then the RAC node is evicted from the cluster. The default for this parameter is 200. This value should be sufficient for standard storage takeover and giveback procedures.

NetApp strongly recommends testing RAC configurations thoroughly before placing them into production because many factors affect a takeover or giveback. In addition to the time required for storage failover to complete, additional time is also required for Link Aggregation Control Protocol (LACP) changes to propagate. Also, SAN multipathing software must detect an I/O timeout and retry on an alternate path. If a database is extremely active, a large amount of I/O must be queued and retried before voting disk I/O is processed.

If an actual storage takeover or giveback cannot be performed, the effect can be simulated with cable pull tests on the database server.

NetApp recommends the following:

- Leaving the `disktimeout` parameter at the default value of 200.
- Always test a RAC configuration thoroughly.

misscount

The `misscount` parameter normally affects only the network heartbeat between RAC nodes. The default is 30 seconds. If the grid binaries are on a storage array or the OS boot drive is not local, this parameter might become important. This includes hosts with boot drives located on an FC SAN, NFS-booted OSs, and boot drives located on virtualization datastores such as a VMDK file.

If access to a boot drive is interrupted by a storage takeover or giveback, it is possible that the grid binary location or the entire OS temporarily hangs. The time required for ONTAP to complete the storage operation and for the OS to change paths and resume I/O might exceed the `misscount` threshold. As a result, a node immediately evicts after connectivity to the boot LUN or grid binaries is restored. In most cases, the eviction and subsequent reboot occur with no logging messages to indicate the reason for the reboot. Not all configurations are affected, so test any SAN-booting, NFS-booting, or datastore-based host in a RAC environment so that RAC remains stable if communication to the boot drive is interrupted.

In the case of nonlocal boot drives or a nonlocal file system hosting `grid` binaries, the `misscount` will need to be changed to match `disktimeout`. If this parameter is changed, conduct further testing to also identify any effects on RAC behavior, such as node failover time.

NetApp recommends the following:

- Leave the `misscount` parameter at the default value of 30 unless one of the following conditions applies:
  - `grid` binaries are located on a network-attached drive, including iSCSI, FC, and datastore-based drives.
  - The OS is SAN booted.
- In such cases, evaluate the effect of network interruptions that affect access to OS or `GRID_HOME` file systems. In some cases, such interruptions cause the Oracle RAC daemons to stall, which can lead to a `misscount`-based timeout and eviction. The timeout defaults to 27 seconds, which is the value of `misscount` minus `reboottime`. In such cases, increase `misscount` to 200 to match `disktimeout`.

- ASA r2's SAN-optimized design reduces failover latency, but timeouts must still be tuned for networked boot or grid binaries.
- For extended RAC or active-active setups (e.g., SnapMirror active sync), timeout tuning remains essential for zero-RPO architectures.

Host configuration with AFF/FAS systems

AIX

Configuration topics for Oracle database on IBM AIX with ONTAP.

Concurrent I/O

Achieving optimum performance on IBM AIX requires the use of concurrent I/O. Without concurrent I/O, performance limitations are likely because AIX performs serialized, atomic I/O, which incurs significant overhead.

Originally, NetApp recommended using the `cio` mount option to force the use of concurrent I/O on the file system, but this process had drawbacks and is no longer required. Since the introduction of AIX 5.2 and Oracle 10gR1, Oracle on AIX can open individual files for concurrent IO, as opposed to forcing concurrent I/O on the entire file system.

The best method for enabling concurrent I/O is to set the `init.ora` parameter `filesystemio_options` to `setall`. Doing so allows Oracle to open specific files for use with concurrent I/O.

Using `cio` as a mount option forces the use of concurrent I/O, which can have negative consequences. For example, forcing concurrent I/O disables readahead on file systems, which can damage performance for I/O occurring outside the Oracle database software, such as copying files and performing tape backups. Furthermore, products such as Oracle GoldenGate and SAP BR*Tools are not compatible with using the `cio` mount option with certain versions of Oracle.

NetApp recommends the following:

- Do not use the `cio` mount option at the file system level. Rather, enable concurrent I/O through the use of `filesystemio_options=setall`.
- Only use the `cio` mount option should if it is not possible to set `filesystemio_options=setall`.

AIX NFS mount options

The following table lists the AIX NFS mount options for Oracle single instance databases.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
Controlfiles Datafiles Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, intr`

The following table lists the AIX NFS mount options for RAC.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
Controlfiles Datafiles Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac`
`CRS/Voting`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac`
Dedicated `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
Shared `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`

The primary difference between single-instance and RAC mount options is the addition of `noac` to the mount options. This addition has the effect of disabling the host OS caching that enables all instances in the RAC cluster to have a consistent view of the state of the data.

Although using the `cio` mount option and the `init.ora` parameter `filesystemio_options=setall` has the same effect of disabling host caching, it is still necessary to use `noac`. `noac` is required for shared `ORACLE_HOME` deployments to facilitate the consistency of files such as Oracle password files and `spfile` parameter files. If each instance in a RAC cluster has a dedicated `ORACLE_HOME`, then this parameter is not

required.

AIX jfs/jfs2 Mount Options

The following table lists the AIX jfs/jfs2 mount options.

File type	Mount options
ADR Home	Defaults
Controlfiles Datafiles Redo logs	Defaults
ORACLE_HOME	Defaults

Before using AIX `hdisk` devices in any environment, including databases, check the parameter `queue_depth`. This parameter is not the HBA queue depth; rather it relates to the SCSI queue depth of the individual `hdisk device. Depending on how the LUNs are configured, the value for` `queue_depth` might be too low for good performance. Testing has shown the optimum value to be 64.

HP-UX

Configuration topics for Oracle database on HP-UX with ONTAP.

HP-UX NFS Mount Options

The following table lists the HP-UX NFS mount options for a single instance.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
Control files Datafiles Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144,forc edirectio, nointr,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

The following table lists the HP-UX NFS mount options for RAC.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, noac,suid`
Control files Datafiles Redo logs	`rw, bg,hard, [vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio,suid`

File type	Mount options
CRS/Voting	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac, forcedirectio,suid`
Dedicated `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
Shared `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

The primary difference between single-instance and RAC mount options is the addition of `noac` and `forcedirectio` to the mount options. This addition has the effect of disabling host OS caching, which enables all instances in the RAC cluster to have a consistent view of the state of the data. Although using the `init.ora` parameter `filesystemio_options=setall` has the same effect of disabling host caching, it is still necessary to use `noac` and `forcedirectio`.

The reason `noac` is required for shared `ORACLE_HOME` deployments is to facilitate consistency of files such as Oracle password files and spfiles. If each instance in a RAC cluster has a dedicated `ORACLE_HOME`, this parameter is not required.

HP-UX VxFS mount options

Use the following mount options for file systems hosting Oracle binaries:

```
delaylog,nodatainlog
```

Use the following mount options for file systems containing datafiles, redo logs, archive logs, and control files in which the version of HP-UX does not support concurrent I/O:

```
nodatainlog,mincache=direct,convosync=direct
```

When concurrent I/O is supported (VxFS 5.0.1 and later, or with the ServiceGuard Storage Management Suite), use these mount options for file systems containing datafiles, redo logs, archive logs, and control files:

```
delaylog,cio
```

The parameter `db_file_multiblock_read_count` is especially critical in VxFS environments. Oracle recommends that this parameter remain unset in Oracle 10g R1 and later unless specifically directed otherwise. The default with an Oracle 8KB block size is 128. If the value of this parameter is forced to 16 or less, remove the `convosync=direct` mount option because it can damage sequential I/O performance. This step damages other aspects of performance and should only be taken if the value of `db_file_multiblock_read_count` must be changed from the default value.

Linux

Configuration topics specific to the Linux OS.

Linux NFSv3 TCP slot tables

TCP slot tables are the NFSv3 equivalent of host bus adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables.

Run the `sysctl -a | grep tcp.*.slot_table` command, and observe the following parameters:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

All Linux systems should include `sunrpc.tcp_slot_table_entries`, but only some include `sunrpc.tcp_max_slot_table_entries`. They should both be set to 128.

Failure to set these parameters may have significant effects on performance. In some cases, performance is limited because the linux OS is not issuing sufficient I/O. In other cases, I/O latencies increases as the linux OS attempts to issue more I/O than can be serviced.

Linux NFS mount options

The following table lists the Linux NFS mount options for a single instance.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
Control files Datafiles Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`
ORACLE_HOME	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`

The following table lists the Linux NFS mount options for RAC.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, actimeo=0`

File type	Mount options
Control files Data files Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,actimeo=0`
CRS/voting	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,actimeo=0`
Dedicated `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
Shared `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,actimeo=0`

The primary difference between single-instance and RAC mount options is the addition of `actimeo=0` to the mount options. This addition has the effect of disabling the host OS caching, which enables all instances in the RAC cluster to have a consistent view of the state of the data. Although using the `init.ora` parameter `filesystemio_options=setall` has the same effect of disabling host caching, it is still necessary to use `actimeo=0`.

The reason `actimeo=0` is required for shared `ORACLE_HOME` deployments is to facilitate consistency of files such as the Oracle password files and spfiles. If each instance in a RAC cluster has a dedicated `ORACLE_HOME`, then this parameter is not required.

Generally, nondatabase files should be mounted with the same options used for single-instance datafiles, although specific applications might have different requirements. Avoid the mount options `noac` and `actimeo=0` if possible because these options disable file system-level readahead and buffering. This can cause severe performance problems for processes such as extraction, translation, and loading.

ACCESS and GETATTR

Some customers have noted that an extremely high level of other IOPS such as ACCESS and GETATTR can dominate their workloads. In extreme cases, operations such as reads and writes can be as low as 10% of the total. This is normal behavior with any database that includes using `actimeo=0` and/or `noac` on Linux because these options cause the Linux OS to constantly reload file metadata from the storage system. Operations such as ACCESS and GETATTR are low-impact operations that are serviced from the ONTAP cache in a database environment. They should not be considered genuine IOPS, such as reads and writes, that create true demand on storage systems. These other IOPS do create some load, however, especially in RAC environments. To address this situation, enable DNFS, which bypasses the OS buffer cache and avoids these unnecessary metadata operations.

Linux Direct NFS

One additional mount option, called `nosharecache`, is required when (a) DNFS is enabled and (b) a source volume is mounted more than once on a single server (c) with a nested NFS mount. This configuration is seen primarily in environments supporting SAP applications. For example, a single volume on a NetApp system could have a directory located at `/vol/oracle/base` and a second at `/vol/oracle/home`. If `/vol/oracle/base` is mounted at `/oracle` and `/vol/oracle/home` is mounted at `/oracle/home`, the result is nested NFS mounts that originate on the same source.

The OS can detect the fact that `/oracle` and `/oracle/home` reside on the same volume, which is the same source file system. The OS then uses the same device handle for accessing the data. Doing so improves the

use of OS caching and certain other operations, but it interferes with DNFS. If DNFS must access a file, such as the `spfile`, on `/oracle/home`, it might erroneously attempt to use the wrong path to the data. The result is a failed I/O operation. In these configurations, add the `nosharecache` mount option to any NFS file system that shares a source volume with another NFS file system on that host. Doing so forces the Linux OS to allocate an independent device handle for that file system.

Linux Direct NFS and Oracle RAC

The use of DNFS has special performance benefits for Oracle RAC on the Linux OS because Linux does not have a method to force direct I/O, which is required with RAC for coherency across the nodes. As a workaround, Linux requires the use of the `actimeo=0` mount option, which causes file data to expire immediately from the OS cache. This option in turn forces the Linux NFS client to constantly reread attribute data, which damages latency and increases load on the storage controller.

Enabling DNFS bypasses the host NFS client and avoids this damage. Multiple customers have reported significant performance improvements on RAC clusters and significant decreases in ONTAP load (especially with respect to other IOPS) when enabling DNFS.

Linux Direct NFS and oranfstab file

When using DNFS on Linux with the multipathing option, multiple subnets must be used. On other OSs, multiple DNFS channels can be established by using the `LOCAL` and `DONTROUTE` options to configure multiple DNFS channels on a single subnet. However, this does not work properly on Linux and unexpected performance problems can result. With Linux, each NIC used for DNFS traffic must be on a different subnet.

I/O scheduler

The Linux kernel allows low-level control over the way that I/O to block devices is scheduled. The defaults on various distribution of Linux vary considerably. Testing shows that Deadline usually offers the best results, but on occasion NOOP has been slightly better. The difference in performance is minimal, but test both options if it is necessary to extract the maximum possible performance from a database configuration. CFQ is the default in many configurations, and it has demonstrated significant performance problems with database workloads.

See the relevant Linux vendor documentation for instructions on configuring the I/O scheduler.

Multipathing

Some customers have encountered crashes during network disruption because the multipath daemon was not running on their system. On recent versions of Linux, the installation process of the OS and the multipathing daemon might leave these OSs vulnerable to this problem. The packages are installed correctly, but they are not configured for automatic startup after a reboot.

For example, the default for the multipath daemon on RHEL5.5 might appear as follows:

```
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

This can be corrected with the following commands:

```
[root@host1 iscsi]# chkconfig multipathd on
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ASM mirroring

ASM mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate fail group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM does not mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites.

The Linux settings shown in the NetApp Host Utilities documentation include multipath parameters that result in indefinite queuing of I/O. This means an I/O on a LUN device with no active paths waits as long as required for the I/O to complete. This is usually desirable because Linux hosts wait as long as needed for SAN path changes to complete, for FC switches to reboot, or for a storage system to complete a failover.

This unlimited queuing behavior causes a problem with ASM mirroring because ASM must receive an I/O failure for it to retry I/O on an alternate LUN.

Set the following parameters in the Linux `multipath.conf` file for ASM LUNs used with ASM mirroring:

```
polling_interval 5
no_path_retry 24
```

These settings create a 120-second timeout for ASM devices. The timeout is calculated as the `polling_interval` * `no_path_retry` as seconds. The exact value might need to be adjusted in some circumstances, but a 120 second timeout should be sufficient for most uses. Specifically, 120 seconds should allow a controller takeover or giveback to occur without producing an I/O error that would result in the fail group being taken offline.

A lower `no_path_retry` value can shorten the time required for ASM to switch to an alternate fail group, but this also increases the risk of an unwanted failover during maintenance activities such as a controller takeover. The risk can be mitigated by careful monitoring of the ASM mirroring state. If an unwanted failover occurs, the mirrors can be rapidly resynced if the resync is performed relatively quickly. For additional information, see the Oracle documentation on ASM Fast Mirror Resync for the version of Oracle software in use.

Linux xfs, ext3, and ext4 mount options

NetApp recommends using the default mount options.

ASMLib/AFD (ASM Filter Driver)

Configuration topics specific to the Linux OS using AFD and ASMlib

ASMlib block sizes

ASMlib is an optional ASM management library and associated utilities. Its primary value is the capability to stamp a LUN or an NFS-based file as an ASM resource with a human-readable label.

Recent versions of ASMlib detect a LUN parameter called Logical Blocks Per Physical Block Exponent (LBPPBE). This value was not reported by the ONTAP SCSI target until recently. It now returns a value that indicates that a 4KB block size is preferred. This is not a definition of block size, but it is a hint to any application that uses LBPPBE that I/Os of a certain size might be handled more efficiently. ASMlib does, however, interpret LBPPBE as a block size and persistently stamps the ASM header when the ASM device is created.

This process can cause problems with upgrades and migrations in a number of ways, all based on the inability to mix ASMlib devices with different block sizes in the same ASM diskgroup.

For example, older arrays generally reported an LBPPBE value of 0 or did not report this value at all. ASMlib interprets this as a 512-byte block size. Newer arrays would be interpreted as having a 4KB block size. It is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The preferred solution is to patch ASMlib. The Oracle bug ID is 13999609, and the patch is present in oracleasm-support-2.1.8-1 and higher. This patch allows a user to set the parameter `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` to `true` in the `/etc/sysconfig/oracleasm` configuration file. Doing so blocks ASMlib from using the LBPPBE parameter, which means that LUNs on the new array are now recognized as 512-byte block devices.

> The option does not change the block size on LUNs that were previously stamped by ASMlib. For example, if an ASM diskgroup with 512-byte blocks must be migrated to a new storage system that reports a 4KB block, the option `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` must be set before the new LUNs are stamped with ASMlib. If devices have already been stamped by oracleasm, they must be reformatted before being restamped with a new block size. First, deconfigure the device with `oracleasm deletedisk`, and then clear the first 1GB of the device with `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`. Finally, if the device had been previously partitioned, use the `kpartx` command to remove stale partitions or simply reboot the OS.

If ASMlib cannot be patched, ASMlib can be removed from the configuration. This change is disruptive and requires the unstamping of ASM disks and making sure that the `asm_diskstring` parameter is set correctly. This change does not, however, require the migration of data.

ASM Filter Drive (AFD) block sizes

AFD is an optional ASM management library which is becoming the replacement for ASMlib. From a storage point of view, it is very similar to ASMlib, but it includes additional features such as the ability to block non-Oracle I/O to reduce the chances of user or application errors that could corrupt data.

Device block sizes

Like ASMlib, AFD also reads the LUN parameter Logical Blocks Per Physical Block Exponent (LBPPBE) and by default uses the physical block size, not the logical block size.

This could create a problem if AFD is added to an existing configuration where the ASM devices are already formatted as 512 byte block devices. The AFD driver would recognize the LUN as a 4K device and the mismatch between the ASM label and the physical device would prevent access. Likewise, migrations would be affected because it is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an

ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The solution is simple - AFD includes a parameter to control whether it uses the logical or physical block sizes. This is a global parameter affecting all devices on the system. To force AFD to use the logical block size, set `options oracleafd oracleafd_use_logical_block_size=1` in the `/etc/modprobe.d/oracleafd.conf` file.

Multipath transfer sizes

Recent linux kernel changes enforce I/O size restrictions sent to multipath devices, and AFD does not honor these restrictions. The I/Os are then rejected, which causes the LUN path to go offline. The result is an inability to install Oracle Grid, configure ASM, or create a database.

The solution is to manually specify the maximum transfer length in the multipath.conf file for ONTAP LUNs:

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

Even if no problems currently exist, this parameter should be set if AFD is used to ensure that a future linux upgrade does not unexpectedly cause problems.

Microsoft Windows

Configuration topics for Oracle database on Microsoft Windows with ONTAP..

NFS

Oracle supports the use of Microsoft Windows with the direct NFS client. This capability offers a path to the management benefits of NFS, including the ability to view files across environments, dynamically resize volumes, and leverage a less expensive IP protocol. See the official Oracle documentation for information on installing and configuring a database on Microsoft Windows using DNFS. No special best practices exist.

SAN

For optimal compression efficiency, ensure the NTFS file system uses an 8K or larger allocation unit. Use of a 4K allocation unit, which is generally the default, negatively impacts compression efficiency.

Solaris

Configuration topics specific to the Solaris OS.

Solaris NFS mount options

The following table lists the Solaris NFS mount options for a single instance.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1], roto=tcp, timeo=600,rsize=262144,wsize=262144`
Controlfiles Datafiles Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,llock,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

The use of `llock` has been proven to dramatically improve performance in customer environments by removing the latency associated with acquiring and releasing locks on the storage system. Use this option with care in environments in which numerous servers are configured to mount the same file systems and Oracle is configured to mount these databases. Although this is a highly unusual configuration, it is used by a small number of customers. If an instance is accidentally started a second time, data corruption can occur because Oracle is unable to detect the lock files on the foreign server. NFS locks do not otherwise offer protection; as in NFS version 3, they are advisory only.

Because the `llock` and `forcedirectio` parameters are mutually exclusive, it is important that `filesystemio_options=setall` is present in the `init.ora` file so that `directio` is used. Without this parameter, host OS buffer caching is used and performance can be adversely affected.

The following table lists the Solaris NFS RAC mount options.

File type	Mount options
ADR Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, noac`
Control files Data files Redo logs	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio`
CRS/Voting	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio`
Dedicated `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
Shared `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

The primary difference between single-instance and RAC mount options is the addition of `noac` and `forcedirectio` to the mount options. This addition has the effect of disabling the host OS caching, which enables all instances in the RAC cluster to have a consistent view of the state of the data. Although using the `init.ora` parameter `filesystemio_options=setall` has the same effect of disabling host caching, it is still necessary to use `noac` and `forcedirectio`.

The reason `actimeo=0` is required for shared `ORACLE_HOME` deployments is to facilitate consistency of files such as Oracle password files and spfiles. If each instance in a RAC cluster has a dedicated `ORACLE_HOME`, this parameter is not required.

Solaris UFS mount options

NetApp strongly recommends using the logging mount option so that data integrity is preserved in the case of a Solaris host crash or the interruption of FC connectivity. The logging mount option also preserves the usability of Snapshot backups.

Solaris ZFS

Solaris ZFS must be installed and configured carefully to deliver optimum performance.

mvector

Solaris 11 included a change in how it processes large I/O operations which can result in severe performance problems on SAN storage arrays. The problem is documented NetApp tracking bug report 630173, "Solaris 11 ZFS Performance Regression."

This is not an ONTAP bug. It is a Solaris defect that is tracked under Solaris defects 7199305 and 7082975.

You can consult Oracle Support to find out if your version of Solaris 11 is affected, or you can test the workaround by changing `zfs_mvector_max_size` to a smaller value.

You can do this by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

If any unexpected problems arise from this change, it can be easily reversed by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

Kernel

Reliable ZFS performance requires a Solaris kernel patched against LUN alignment problems. The fix was introduced with patch 147440-19 in Solaris 10 and with SRU 10.5 for Solaris 11. Only use Solaris 10 and later with ZFS.

LUN configuration

To configure a LUN, complete the following steps:

1. Create a LUN of type `solaris`.
2. Install the appropriate Host Utility Kit (HUK) specified by the NetApp Interoperability Matrix Tool (IMT).
3. Follow the instructions in the HUK exactly as described. The basic steps are outlined below, but refer to the latest documentation for the proper procedure.
    a. Run the `host_config` utility to update the `sd.conf/sdd.conf` file. Doing so allows the SCSI drives

to correctly discover ONTAP LUNs.

b. Follow the instructions given by the `host_config` utility to enable multipath input/output (MPIO).

c. Reboot. This step is required so that any changes are recognized across the system.

4. Partition the LUNs and verify that they are properly aligned. See "Appendix B: WAFL Alignment Verification" for instructions on how to directly test and confirm alignment.

zpools

A zpool should only be created after the steps in the LUN Configuration are performed. If the procedure is not done correctly, it can result in serious performance degradation due to the I/O alignment. Optimum performance on ONTAP requires I/O to be aligned to a 4K boundary on a drive. The file systems created on a zpool use an effective block size that is controlled through a parameter called `ashift`, which can be viewed by running the command `zdb -C`.

The value of `ashift` defaults to 9, which means 2^9, or 512 bytes. For optimum performance, the `ashift` value must be 12 (2^12=4K). This value is set at the time the zpool is created and cannot be changed, which means that data in zpools with `ashift` other than 12 should be migrated by copying data to a newly created zpool.

After creating a zpool, verify the value of `ashift` before proceeding. If the value is not 12, the LUNs were not discovered correctly. Destroy the zpool, verify that all steps shown in the relevant Host Utilities documentation were performed correctly, and recreate the zpool.

zpools and Solaris LDOMs

Solaris LDOMs create an additional requirement for making sure that I/O alignment is correct. Although a LUN might be properly discovered as a 4K device, a virtual vdsk device on an LDOM does not inherit the configuration from the I/O domain. The vdsk based on that LUN defaults back to a 512-byte block.

An additional configuration file is required. First, the individual LDOM's must be patched for Oracle bug 15824910 to enable the additional configuration options. This patch has been ported into all currently used versions of Solaris. Once the LDOM is patched, it is ready for configuration of the new properly aligned LUNs as follows:

1. Identify the LUN or LUNs to be used in the new zpool. In this example, it is the c2d1 device.

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. Retrieve the vdc instance of the devices to be used for a ZFS pool:

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. Edit `/platform/sun4v/kernel/drv/vdc.conf`:

```
block-size-list="1:4096";
```

This means that device instance 1 is assigned a block size of 4096.

As an additional example, assume vdsk instances 1 through 6 need to be configured for a 4K block size and `/etc/path_to_inst` reads as follows:

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. The final `vdc.conf` file should contain the following:

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

Caution
The LDOM must be rebooted after vdc.conf is configured and the vdsk is created. This step cannot be avoided. The block size change only takes effect after a reboot. Proceed with zpool configuration and ensure that ashift is properly set to 12 as described previously.

ZFS Intent Log (ZIL)

Generally, there is no reason to locate the ZFS Intent Log (ZIL) on a different device. The log can share space with the main pool. The primary use of a separate ZIL is when using physical drives that lack the write caching features in modern storage arrays.

logbias

Set the `logbias` parameter on ZFS file systems hosting Oracle data.

```
zfs set logbias=throughput <filesystem>
```

Using this parameter reduces overall write levels. Under the defaults, written data is committed first to the ZIL and then to the main storage pool. This approach is appropriate for a configuration using a plain drive configuration, which includes an SSD-based ZIL device and spinning media for the main storage pool. This is because it allows a commit to occur in a single I/O transaction on the lowest latency media available.

When using a modern storage array that includes its own caching capability, this approach is not generally necessary. Under rare circumstances, it might be desirable to commit a write with a single transaction to the log, such as a workload that consists of highly concentrated, latency-sensitive random writes. There are consequences in the form of write amplification because the logged data is eventually written to the main storage pool, resulting in a doubling of the write activity.

Direct I/O

Many applications, including Oracle products, can bypass the host buffer cache by enabling direct I/O. This strategy does not work as expected with ZFS file systems. Although the host buffer cache is bypassed, ZFS itself continues to cache data. This action can result in misleading results when using tools such as fio or sio to perform performance tests because it is difficult to predict whether I/O is reaching the storage system or whether it is being cached locally within the OS. This action also makes it very difficult to use such synthetic tests to compare ZFS performance to other file systems. As a practical matter, there is little to no difference in file system performance under real user workloads.

Multiple zpools

Snapshot-based backups, restores, clones, and archiving of ZFS-based data must be performed at the level of the zpool and typically requires multiple zpools. A zpool is analogous to an LVM disk group and should be configured using the same rules. For example, a database is probably best laid out with the datafiles residing on `zpool1` and the archive logs, control files, and redo logs residing on `zpool2`. This approach permits a standard hot backup in which the database is placed in hot backup mode, followed by a snapshot of `zpool1`. The database is then removed from hot backup mode, the log archive is forced, and a snapshot of `zpool2` is created. A restore operation requires unmounting the zfs file systems and offlining the zpool in its entirety, following by a SnapRestore restore operation. The zpool can then be brought online again and the database recovered.

filesystemio_options

The Oracle parameter `filesystemio_options` works differently with ZFS. If `setall` or `directio` is used, write operations are synchronous and bypass the OS buffer cache, but reads are buffered by ZFS. This action causes difficulties in performance analysis because I/O is sometimes intercepted and serviced by the ZFS cache, making storage latency and total I/O less than it might appear to be.

Host configuration with ASA r2 systems

AIX

Configuration topics for Oracle database on IBM AIX with ASA r2 ONTAP.

AIX is supported with NetApp ASA r2 for hosting Oracle databases, provided:

- You configure Oracle for concurrent I/O properly.
- You use supported SAN protocols (FC/iSCSI/NVMe).
- You run ONTAP 9.16.x or later on ASA r2.

Concurrent I/O

Achieving optimum performance on IBM AIX with ASA r2 requires the use of concurrent I/O. Without concurrent I/O, performance limitations are likely because AIX performs serialized, atomic I/O, which incurs significant overhead.

Originally, NetApp recommended using the `cio` mount option to force concurrent I/O on the file system, but this process had drawbacks and is no longer required. Since the introduction of AIX 5.2 and Oracle 10gR1, Oracle on AIX can open individual files for concurrent I/O, as opposed to forcing concurrent I/O on the entire file system.

The best method for enabling concurrent I/O is to set the `init.ora` parameter `filesystemio_options` to `setall`. Doing so allows Oracle to open specific files for use with concurrent I/O.

Using cio as a mount option forces the use of concurrent I/O, which can have negative consequences. For example, forcing concurrent I/O disables readahead on file systems, which can damage performance for I/O occurring outside the Oracle database software, such as copying files and performing tape backups. Furthermore, products such as Oracle GoldenGate and SAP BR*Tools are not compatible with using the cio mount option with certain versions of Oracle.

NetApp recommends the following:

- Do not use the `cio` mount option at the file system level. Rather, enable concurrent I/O through the use of `filesystemio_options=setall`.
- Only use the `cio` mount option if it is not possible to set `filesystemio_options=setall`.

Since ASA r2 does not support NAS, all Oracle deployments on AIX must use block protocols.

AIX jfs/jfs2 Mount Options

The following table lists the AIX jfs/jfs2 mount options.

File type	Mount options
ADR Home	Defaults
Controlfiles	Defaults
Datafiles	Defaults
Redo logs	Defaults
ORACLE_HOME	Defaults

Before using AIX `hdisk` devices in any environment, including databases, check the parameter `queue_depth`. This parameter is not the HBA queue depth; rather it relates to the SCSI queue depth of the individual `hdisk device`. Depending on how the ASA r2 LUNs are configured, the value for `queue_depth` might be too low for good performance. Testing has shown the optimum value to be 64.

HP-UX

Configuration topics for Oracle database on HP-UX with ASA r2 ONTAP.

HP-UX is supported with NetApp ASA r2 for hosting Oracle databases, provided:

- ONTAP version is 9.16.x or later.
- Use SAN protocols (FC/iSCSI/NVMe). NAS is not supported on ASA r2.
- Apply HP-UX-specific mount and I/O tuning best practices.

HP-UX VxFS mount options

Use the following mount options for file systems hosting Oracle binaries:

```
delaylog,nodatainlog
```

Use the following mount options for file systems containing datafiles, redo logs, archive logs, and control files in which the version of HP-UX does not support concurrent I/O:

```
nodatainlog,mincache=direct,convosync=direct
```

When concurrent I/O is supported (VxFS 5.0.1 and later, or with the ServiceGuard Storage Management Suite), use these mount options for file systems containing datafiles, redo logs, archive logs, and control files:

```
delaylog,cio
```

The parameter `db_file_multiblock_read_count` is especially critical in VxFS environments. Oracle recommends that this parameter remain unset in Oracle 10g R1 and later unless specifically directed otherwise. The default with an Oracle 8KB block size is 128. If the value of this parameter is forced to 16 or less, remove the `convosync=direct` mount option because it can damage sequential I/O performance. This step damages other aspects of performance and should only be taken if the value of `db_file_multiblock_read_count` must be changed from the default value.

Linux

Configuration topics specific to the Linux OS with ASA r2 ONTAP.

Linux (Oracle Linux, RHEL, SUSE) is supported with ASA r2 for Oracle databases. Use SAN protocols, configure multipathing correctly, and apply Oracle best practices for ASM and I/O tuning.

I/O scheduler

The Linux kernel allows low-level control over the way that I/O to block devices is scheduled. The defaults on various distribution of Linux vary considerably. Testing shows that Deadline usually offers the best results, but on occasion NOOP has been slightly better. The difference in performance is minimal, but test both options if it is necessary to extract the maximum possible performance from a database configuration. CFQ is the default in many configurations, and it has demonstrated significant performance problems with database workloads.

See the relevant Linux vendor documentation for instructions on configuring the I/O scheduler.

Multipathing

Some customers have encountered crashes during network disruption because the multipath daemon was not running on their system. On recent versions of Linux, the installation process of the OS and the multipathing daemon might leave these OSs vulnerable to this problem. The packages are installed correctly, but they are not configured for automatic startup after a reboot.

For example, the default for the multipath daemon on RHEL 9.7 might appear as follows:

```
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          disabled
```

This can be corrected with the following commands:

```
[root@host1 ~]# systemctl enable multipathd.service
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          enabled
```

Queue depth

Set appropriate queue depth for SAN devices to avoid I/O bottlenecks. The default queue depth on Linux is often set to 128, which can lead to performance problems with Oracle databases. Setting the queue depth too high can cause excessive I/O queuing, leading to increased latency and reduced throughput. Setting it too low

can limit the number of outstanding I/O requests, reducing overall performance. A queue depth of 64 is often a good starting point for Oracle database workloads on ASA r2, but it may need to be adjusted based on specific workload characteristics and performance testing.

ASM mirroring

ASM mirroring might require changes to the Linux multipath settings to allow ASM to recognize a problem and switch over to an alternate fail group. Most ASM configurations on ONTAP use external redundancy, which means that data protection is provided by the external array and ASM does not mirror data. Some sites use ASM with normal redundancy to provide two-way mirroring, normally across different sites.

For ASA r2 systems that support active-active multipathing, these multipath settings should be adjusted. Since all paths are active and load-balanced, indefinite queuing is not required. Instead, multipath parameters should prioritize performance and quick failback. This behavior is important for ASM mirroring because ASM must receive an I/O failure for it to retry I/O on an alternate LUN. If I/O is queued indefinitely, ASM cannot trigger a failover.

Set the following parameters in the Linux `multipath.conf` file for ASM LUNs used with ASM mirroring:

```
polling_interval 5
no_path_retry 24
failback immediate
path_grouping_policy multibus
path_selector "service-time 0"
```

These settings create a 120-second timeout for ASM devices. The timeout is calculated as the `polling_interval` * `no_path_retry` as seconds. The exact value might need to be adjusted in some circumstances, but a 120 second timeout should be sufficient for most uses. Specifically, 120 seconds should allow a controller takeover or giveback to occur without producing an I/O error that would result in the fail group being taken offline.

A lower `no_path_retry` value can shorten the time required for ASM to switch to an alternate fail group, but this also increases the risk of an unwanted failover during maintenance activities such as a controller takeover. The risk can be mitigated by careful monitoring of the ASM mirroring state. If an unwanted failover occurs, the mirrors can be rapidly resynced if the resync is performed relatively quickly. For additional information, see the Oracle documentation on ASM Fast Mirror Resync for the version of Oracle software in use.

Linux xfs, ext3, and ext4 mount options

NetApp recommends using the default mount options. Ensure proper alignment when creating file systems on LUNs.

ASMLib/AFD (ASM Filter Driver)

Configuration topics specific to the Linux OS using AFD and ASMlib with ASA r2 ONTAP.

ASMlib block sizes

ASMlib is an optional ASM management library and associated utilities. Its primary value is the capability to stamp a LUN as an ASM resource with a human-readable label.

Recent versions of ASMlib detect a LUN parameter called Logical Blocks Per Physical Block Exponent (LBPPBE). This value was not reported by the ONTAP SCSI target until recently. It now returns a value that indicates that a 4KB block size is preferred. This is not a definition of block size, but it is a hint to any application that uses LBPPBE that I/Os of a certain size might be handled more efficiently. ASMlib does, however, interpret LBPPBE as a block size and persistently stamps the ASM header when the ASM device is created.

This process can cause problems with upgrades and migrations in a number of ways, all based on the inability to mix ASMlib devices with different block sizes in the same ASM diskgroup.

For example, older arrays generally reported an LBPPBE value of 0 or did not report this value at all. ASMlib interprets this as a 512-byte block size. Newer arrays would be interpreted as having a 4KB block size. It is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The preferred solution is to patch ASMlib. The Oracle bug ID is 13999609, and the patch is present in oracleasm-support-2.1.8-1 and higher. This patch allows a user to set the parameter `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` to `true` in the `/etc/sysconfig/oracleasm` configuration file. Doing so blocks ASMlib from using the LBPPBE parameter, which means that LUNs on the new array are now recognized as 512-byte block devices.

> The option does not change the block size on LUNs that were previously stamped by ASMlib. For example, if an ASM diskgroup with 512-byte blocks must be migrated to a new storage system that reports a 4KB block, the option `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` must be set before the new LUNs are stamped with ASMlib. If devices have already been stamped by oracleasm, they must be reformatted before being restamped with a new block size. First, deconfigure the device with `oracleasm deletedisk`, and then clear the first 1GB of the device with `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`. Finally, if the device had been previously partitioned, use the `kpartx` command to remove stale partitions or simply reboot the OS.

If ASMlib cannot be patched, ASMlib can be removed from the configuration. This change is disruptive and requires the unstamping of ASM disks and making sure that the `asm_diskstring` parameter is set correctly. This change does not, however, require the migration of data.

ASM Filter Drive (AFD) block sizes

AFD is an optional ASM management library which is becoming the replacement for ASMlib. From a storage point of view, it is very similar to ASMlib, but it includes additional features such as the ability to block non-Oracle I/O to reduce the chances of user or application errors that could corrupt data.

Device block sizes

Like ASMlib, AFD also reads the LUN parameter Logical Blocks Per Physical Block Exponent (LBPPBE) and by default uses the physical block size, not the logical block size.

This could create a problem if AFD is added to an existing configuration where the ASM devices are already formatted as 512 byte block devices. The AFD driver would recognize the LUN as a 4K device and the mismatch between the ASM label and the physical device would prevent access. Likewise, migrations would be affected because it is not possible to mix both 512-byte and 4KB devices in the same ASM diskgroup. Doing so would block a user from increasing the size of the ASM diskgroup using LUNs from two arrays or leveraging ASM as a migration tool. In other cases, RMAN might not permit the copying of files between an

ASM diskgroup with a 512-byte block size and an ASM diskgroup with a 4KB block size.

The solution is simple - AFD includes a parameter to control whether it uses the logical or physical block sizes. This is a global parameter affecting all devices on the system. To force AFD to use the logical block size, set `options oracleafd oracleafd_use_logical_block_size=1` in the `/etc/modprobe.d/oracleafd.conf` file.

Multipath transfer sizes

Recent linux kernel changes enforce I/O size restrictions sent to multipath devices, and AFD does not honor these restrictions. The I/Os are then rejected, which causes the LUN path to go offline. The result is an inability to install Oracle Grid, configure ASM, or create a database.

The solution is to manually specify the maximum transfer length in the multipath.conf file for ONTAP LUNs:

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

Even if no problems currently exist, this parameter should be set if AFD is used to ensure that a future linux upgrade does not unexpectedly cause problems.

Microsoft Windows

Configuration topics for Oracle database on Microsoft Windows with ASA r2 ONTAP.

SAN

For optimal compression efficiency, ensure the NTFS file system uses an 8K or larger allocation unit. Use of a 4K allocation unit, which is generally the default, negatively impacts compression efficiency.

Solaris

Configuration topics specific to the Solaris OS with ASA r2 ONTAP.

Solaris UFS mount options

NetApp strongly recommends using the logging mount option so that data integrity is preserved in the case of a Solaris host crash or the interruption of FC connectivity. The logging mount option also preserves the usability of Snapshot backups.

Solaris ZFS

Solaris ZFS must be installed and configured carefully to deliver optimum performance.

mvector

Solaris 11 included a change in how it processes large I/O operations which can result in severe performance problems on SAN storage arrays. The problem is documented NetApp tracking bug report 630173, "Solaris 11 ZFS Performance Regression."

This is not an ONTAP bug. It is a Solaris defect that is tracked under Solaris defects 7199305 and 7082975.

You can consult Oracle Support to find out if your version of Solaris 11 is affected, or you can test the workaround by changing `zfs_mvector_max_size` to a smaller value.

You can do this by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

If any unexpected problems arise from this change, it can be easily reversed by running the following command as root:

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

Kernel

Reliable ZFS performance requires a Solaris kernel patched against LUN alignment problems. The fix was introduced with patch 147440-19 in Solaris 10 and with SRU 10.5 for Solaris 11. Only use Solaris 10 and later with ZFS.

LUN configuration

To configure a LUN, complete the following steps:

1. Create a LUN of type `solaris`.
2. Install the appropriate Host Utility Kit (HUK) specified by the NetApp Interoperability Matrix Tool (IMT).
3. Follow the instructions in the HUK exactly as described. The basic steps are outlined below, but refer to the latest documentation for the proper procedure.
    a. Run the `host_config` utility to update the `sd.conf/sdd.conf` file. Doing so allows the SCSI drives to correctly discover ONTAP LUNs.
    b. Follow the instructions given by the `host_config` utility to enable multipath input/output (MPIO).
    c. Reboot. This step is required so that any changes are recognized across the system.
4. Partition the LUNs and verify that they are properly aligned. See "Appendix B: WAFL Alignment Verification" for instructions on how to directly test and confirm alignment.

zpools

A zpool should only be created after the steps in the LUN Configuration are performed. If the procedure is not done correctly, it can result in serious performance degradation due to the I/O alignment. Optimum performance on ONTAP requires I/O to be aligned to a 4K boundary on a drive. The file systems created on a zpool use an effective block size that is controlled through a parameter called `ashift`, which can be viewed by running the command `zdb -C`.

The value of `ashift` defaults to 9, which means 2^9, or 512 bytes. For optimum performance, the `ashift` value must be 12 (2^12=4K). This value is set at the time the zpool is created and cannot be changed, which means that data in zpools with `ashift` other than 12 should be migrated by copying data to a newly created zpool.

After creating a zpool, verify the value of `ashift` before proceeding. If the value is not 12, the LUNs were not discovered correctly. Destroy the zpool, verify that all steps shown in the relevant Host Utilities documentation were performed correctly, and recreate the zpool.

zpools and Solaris LDOMs

Solaris LDOMs create an additional requirement for making sure that I/O alignment is correct. Although a LUN might be properly discovered as a 4K device, a virtual vdsk device on an LDOM does not inherit the configuration from the I/O domain. The vdsk based on that LUN defaults back to a 512-byte block.

An additional configuration file is required. First, the individual LDOM's must be patched for Oracle bug 15824910 to enable the additional configuration options. This patch has been ported into all currently used versions of Solaris. Once the LDOM is patched, it is ready for configuration of the new properly aligned LUNs as follows:

1. Identify the LUN or LUNs to be used in the new zpool. In this example, it is the c2d1 device.

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. Retrieve the vdc instance of the devices to be used for a ZFS pool:

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. Edit `/platform/sun4v/kernel/drv/vdc.conf`:

```
block-size-list="1:4096";
```

This means that device instance 1 is assigned a block size of 4096.

As an additional example, assume vdsk instances 1 through 6 need to be configured for a 4K block size and `/etc/path_to_inst` reads as follows:

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. The final `vdc.conf` file should contain the following:

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

The LDOM must be rebooted after vdc.conf is configured and the vdsk is created. This step cannot be avoided. The block size change only takes effect after a reboot. Proceed with zpool configuration and ensure that ashift is properly set to 12 as described previously.

ZFS Intent Log (ZIL)

Generally, there is no reason to locate the ZFS Intent Log (ZIL) on a different device. The log can share space with the main pool. The primary use of a separate ZIL is when using physical drives that lack the write caching features in modern storage arrays.

logbias

Set the `logbias` parameter on ZFS file systems hosting Oracle data.

```
zfs set logbias=throughput <filesystem>
```

Using this parameter reduces overall write levels. Under the defaults, written data is committed first to the ZIL and then to the main storage pool. This approach is appropriate for a configuration using a plain drive configuration, which includes an SSD-based ZIL device and spinning media for the main storage pool. This is because it allows a commit to occur in a single I/O transaction on the lowest latency media available.

When using a modern storage array that includes its own caching capability, this approach is not generally necessary. Under rare circumstances, it might be desirable to commit a write with a single transaction to the log, such as a workload that consists of highly concentrated, latency-sensitive random writes. There are consequences in the form of write amplification because the logged data is eventually written to the main storage pool, resulting in a doubling of the write activity.

Direct I/O

Many applications, including Oracle products, can bypass the host buffer cache by enabling direct I/O. This strategy does not work as expected with ZFS file systems. Although the host buffer cache is bypassed, ZFS itself continues to cache data. This action can result in misleading results when using tools such as fio or sio to perform performance tests because it is difficult to predict whether I/O is reaching the storage system or whether it is being cached locally within the OS. This action also makes it very difficult to use such synthetic tests to compare ZFS performance to other file systems. As a practical matter, there is little to no difference in file system performance under real user workloads.

Multiple zpools

Snapshot-based backups, restores, clones, and archiving of ZFS-based data must be performed at the level of the zpool and typically requires multiple zpools. A zpool is analogous to an LVM disk group and should be configured using the same rules. For example, a database is probably best laid out with the datafiles residing on `zpool1` and the archive logs, control files, and redo logs residing on `zpool2`. This approach permits a standard hot backup in which the database is placed in hot backup mode, followed by a snapshot of `zpool1`. The database is then removed from hot backup mode, the log archive is forced, and a snapshot of `zpool2` is created. A restore operation requires unmounting the zfs file systems and offlining the zpool in its entirety, following by a SnapRestore restore operation. The zpool can then be brought online again and the database recovered.

filesystemio_options

The Oracle parameter `filesystemio_options` works differently with ZFS. If `setall` or `directio` is used, write operations are synchronous and bypass the OS buffer cache, but reads are buffered by ZFS. This action causes difficulties in performance analysis because I/O is sometimes intercepted and serviced by the ZFS cache, making storage latency and total I/O less than it might appear to be.

Network configuration on AFF/FAS systems

Logical interfaces

Oracle databases need access to storage. Logical interfaces (LIFs) are the network plumbing that connects a storage virtual machine (SVM) to the network and in turn to the database. Proper LIF design is required to ensure sufficient bandwidth exists for each database workload, and failover does not result in a loss of storage services.

This section provides an overview of key LIF design principles. For more comprehensive documentation, see the ONTAP Network Management documentation. As with other aspects of database architecture, the best options for storage virtual machine (SVM, known as a vserver at the CLI) and logical interface (LIF) design depend heavily on scaling requirements and business needs.

Consider the following primary topics when building a LIF strategy:

- **Performance.** Is the network bandwidth sufficient?
- **Resiliency.** Are there any single points of failure in the design?
- **Manageability.** Can the network be scaled nondisruptively?

These topics apply to the end-to-end solution, from the host through the switches to the storage system.

LIF types

There are multiple LIF types. ONTAP documentation on LIF types provide more complete information on this topic, but from a functional perspective LIFs can be divided into the following groups:

- **Cluster and node management LIFs.** LIFs used to manage the storage cluster.
- **SVM management LIFs.** Interfaces that permit access to an SVM through the REST API or ONTAPI (also known as ZAPI) for functions such as snapshot creation or volume resizing. Products such as SnapManager for Oracle (SMO) must have access to an SVM management LIF.
- **Data LIFs.** Interfaces for FC, iSCSI, NVMe/FC, NVMe/TCP, NFS, or SMB/CIFS data.

> A data LIF used for NFS traffic can also be used for management by changing the firewall policy from `data` to `mgmt` or another policy that allows HTTP, HTTPS, or SSH. This change can simplify network configuration by avoiding the configuration of each host for access to both the NFS data LIF and a separate management LIF. It is not possible to configure an interface for both iSCSI and management traffic, despite the fact that both use an IP protocol. A separate management LIF is required in iSCSI environments.

SAN LIF design

LIF design in a SAN environment is relatively simple for one reason: multipathing. All modern SAN implementations allow a client to access data over multiple, independent, network paths and select the best

path or paths for access. As a result, performance with respect to LIF design is simpler to address because SAN clients automatically load-balance I/O across the best available paths.

If a path becomes unavailable, the client automatically selects a different path. The resulting simplicity of design makes SAN LIFs generally more manageable. This does not mean that a SAN environment is always more easily managed, because there are many other aspects of SAN storage that are much more complicated than NFS. It simply means that SAN LIF design is easier.

Performance

The most important consideration with LIF performance in a SAN environment is bandwidth. For example, a two-node ONTAP AFF cluster with two 16Gb FC ports per node allows up to 32Gb of bandwidth to/from each node.

Resiliency

SAN LIFs do not fail over on an AFF storage system. If a SAN LIF fails because of controller failover, then the client's multipathing software detects the loss of a path and redirects I/O to a different LIF. With ASA storage systems, LIFs will be failed over after a short delay, but this does not interrupt IO because there are already active paths on the other controller. The failover process occurs in order to restore host access on all defined ports.

Manageability

LIF migration is a much more common task in an NFS environment because LIF migration is often associated with relocating volumes around the cluster. There is no need to migrate a LIF in a SAN environment when volumes are relocated within the HA pair. That is because, after the volume move has completed, ONTAP sends a notification to the SAN about a change in paths, and the SAN clients automatically reoptimize. LIF migration with SAN is primarily associated with major physical hardware changes. For example, if a nondisruptive upgrade of the controllers is required, a SAN LIF is migrated to the new hardware. If an FC port is found to be faulty, a LIF can be migrated to an unused port.

Design recommendations

NetApp makes the following recommendations:

- Do not create more paths than are required. Excessive numbers of paths make overall management more complicated and can cause problems with path failover on some hosts. Furthermore, some hosts have unexpected path limitations for configurations such as SAN booting.
- Very few configurations should require more than four paths to a LUN. The value of having more than two nodes advertising paths to LUNs is limited because the aggregate hosting a LUN is inaccessible if the node that owns the LUN and its HA partner fail. Creating paths on nodes other than the primary HA pair is not helpful in such a situation.
- Although the number of visible LUN paths can be managed by selecting which ports are included in FC zones, it is generally easier to include all potential target points in the FC zone and control LUN visibility at the ONTAP level.
- In ONTAP 8.3 and later, the selective LUN mapping (SLM) feature is the default. With SLM, any new LUN is automatically advertised from the node that owns the underlying aggregate and the node's HA partner. This arrangement avoids the need to create port sets or configure zoning to limit port accessibility. Each LUN is available on the minimum number of nodes required for both optimal performance and resiliency. *In the event a LUN must be migrated outside of the two controllers, the additional nodes can be added with the `lun mapping add-reporting-nodes` command so that the LUNs are advertised on the new nodes. Doing so creates additional SAN paths to the LUNs for LUN migration. However, the host must perform a discovery operation to use the new paths.

- Do not be overly concerned about indirect traffic. It is best to avoid indirect traffic in a very I/O-intensive environment for which every microsecond of latency is critical, but the visible performance effect is negligible for typical workloads.

NFS LIF design

In contrast to SAN protocols, NFS has a limited ability to define multiple paths to data. The parallel NFS (pNFS) extensions to NFSv4 address this limitation, but as ethernet speeds have reached 100Gb and beyond there is rarely value in adding additional paths.

Performance and resiliency

Although measuring SAN LIF performance is primarily a matter of calculating the total bandwidth from all primary paths, determining NFS LIF performance requires taking a closer look at the exact network configuration. For example, two 10Gb ports can be configured as raw physical ports, or they can be configured as a Link Aggregation Control Protocol (LACP) interface group. If they are configured as an interface group, multiple load balancing policies are available that work differently depending on whether traffic is switched or routed. Finally, Oracle direct NFS (dNFS) offers load-balancing configurations that do not exist in any OS NFS clients at this time.

Unlike SAN protocols, NFS file systems require resiliency at the protocol layer. For example, a LUN is always configured with multipathing enabled, meaning that multiple redundant channels are available to the storage system, each of which uses the FC protocol. An NFS file system, on the other hand, depends on the availability of a single TCP/IP channel that can only be protected at the physical layer. This arrangement is why options such as port failover and LACP port aggregation exist.

In an NFS environment, both performance and resiliency are provided at the network protocol layer. As a result, both topics are intertwined and must be discussed together.

Bind LIFs to port groups

To bind a LIF to a port group, associate the LIF IP address with a group of physical ports. The primary method for aggregating physical ports together is LACP. The fault-tolerance capability of LACP is fairly simple; each port in an LACP group is monitored and is removed from the port group in the event of a malfunction. There are, however, many misconceptions about how LACP works with respect to performance:

- LACP does not require the configuration on the switch to match the endpoint. For example, ONTAP can be configured with IP-based load balancing, while a switch can use MAC-based load balancing.
- Each endpoint using an LACP connection can independently choose the packet transmission port, but it cannot choose the port used for receipt. This means that traffic from ONTAP to a particular destination is tied to a particular port, and the return traffic could arrive on a different interface. This does not cause problems, however.
- LACP does not evenly distribute traffic all the time. In a large environment with many NFS clients, the result is typically even use of all ports in an LACP aggregation. However, any one NFS file system in the environment is limited to the bandwidth of only one port, not the entire aggregation.
- Although robin-robin LACP policies are available on ONTAP, these policies do not address the connection from a switch to a host. For example, a configuration with a four-port LACP trunk on a host and a four-port LACP trunk on ONTAP is still only able to read a file system using a single port. Although ONTAP can transmit data through all four ports, no switch technologies are currently available that send from the switch to the host through all four ports. Only one is used.

The most common approach in larger environments consisting of many database hosts is to build an LACP aggregate of an appropriate number of 10Gb (or faster) interfaces by using IP load balancing. This approach

enables ONTAP to deliver even use of all ports, as long as enough clients exist. Load balancing breaks down when there are fewer clients in the configuration because LACP trunking does not dynamically redistribute load.

When a connection is established, traffic in a particular direction is placed on only one port. For example, a database performing a full table scan against an NFS file system connected through a four-port LACP trunk reads data though only one network interface card (NIC). If only three database servers are in such an environment, it is possible that all three are reading from the same port, while the other three ports are idle.

Bind LIFs to physical ports

Binding a LIF to a physical port results in more granular control over network configuration because a given IP address on a ONTAP system is associated with only one network port at a time. Resiliency is then accomplished through the configuration of failover groups and failover policies.

Failover policies and failover groups

The behavior of LIFs during network disruption is controlled by failover policies and failover groups. Configuration options have changed with the different versions of ONTAP. Consult the ONTAP network management documentation for failover groups and policies for specific details for the version of ONTAP being deployed.

ONTAP 8.3 and higher allow management of LIF failover based on broadcast domains. Therefore, an administrator can define all of the ports that have access to a given subnet and allow ONTAP to select an appropriate failover LIF. This approach can be used by some customers, but it has limitations in a high-speed storage network environment because of the lack of predictability. For example, an environment can include both 1Gb ports for routine file system access and 10Gb ports for datafile I/O. If both types of ports exist in the same broadcast domain, LIF failover can result in moving datafile I/O from a 10Gb port to a 1Gb port.

In summary, consider the following practices:

1. Configure a failover group as user-defined.
2. Populate the failover group with ports on the storage failover (SFO) partner controller so that the LIFs follow the aggregates during a storage failover. This avoids creating indirect traffic.
3. Use failover ports with matching performance characteristics to the original LIF. For example, a LIF on a single physical 10Gb port should include a failover group with a single 10Gb port. A four-port LACP LIF should fail over to another four-port LACP LIF. These ports would be a subset of the ports defined in the broadcast domain.
4. Set the failover policy to SFO-partner only. Doing so makes sure that the LIF follows the aggregate during failover.

Auto-revert

Set the `auto-revert` parameter as desired. Most customers prefer to set this parameter to `true` to have the LIF revert to its home port. However, in some cases, customers have set this to `false `so that an unexpected failover can be investigated before returning a LIF to its home port.

LIF-to-volume ratio

A common misconception is that there must be a 1:1 relationship between volumes and NFS LIFs. Although this configuration is required for moving a volume anywhere in a cluster while never creating additional interconnect traffic, it is categorically not a requirement. Intercluster traffic must be considered, but the mere presence of intercluster traffic does not create problems. Many of the published benchmarks created for

ONTAP include predominantly indirect I/O.

For example, a database project containing a relatively small number of performance-critical databases that only required a total of 40 volumes might warrant a 1:1 volume to LIF strategy, an arrangement that would require 40 IP addresses. Any volume could then be moved anywhere in the cluster along with the associated LIF, and traffic would always be direct, minimizing every source of latency even at microsecond levels.

As a counter example, a large, hosted environment might be more easily managed with a 1:1 relationship between customers and LIFs. Over time, a volume might need to be migrated to a different node, which would cause some indirect traffic. However, the performance effect should be undetectable unless the network ports on the interconnect switch are saturating. If there is concern, a new LIF can be established on additional nodes and the host can be updated at the next maintenance window to remove indirect traffic from the configuration.

TCP/IP and ethernet configuration

Many Oracle on ONTAP customers use ethernet, the network protocol of NFS, iSCSI, NVMe/TCP, and especially the cloud.

Host OS settings

Most application vendor documentation include specific TCP and ethernet settings intended to ensure the application is working optimally. These same settings are usually sufficient to also deliver optimal IP-based storage performance.

Ethernet flow control

This technology allows a client to request that a sender temporarily stop data transmission. This is usually done because the receiver is unable to process incoming data quickly enough. At one time, requesting that a sender cease transmission was less disruptive than having a receiver discard packets because buffers were full. This is no longer the case with the TCP stacks used in OSs today. In fact, flow control causes more problems than it solves.

Performance problems caused by Ethernet flow control have been increasing in recent years. This is because Ethernet flow control operates at the physical layer. If a network configuration permits any host OS to send an Ethernet flow control request to a storage system, the result is a pause in I/O for all connected clients. Because an increasing number of clients are served by a single storage controller, the likelihood of one or more of these clients sending flow control requests increases. The problem has been seen frequently at customer sites with extensive OS virtualization.

A NIC on a NetApp system should not receive flow-control requests. The method used to achieve this result varies based on the network switch manufacturer. In most cases, flow control on an Ethernet switch can be set to `receive desired` or `receive on`, which means that a flow control request is not forwarded to the storage controller. In other cases, the network connection on the storage controller might not allow flow-control disabling. In these cases, the clients must be configured to never send flow control requests, either by changing to the NIC configuration on the host server itself or the switch ports to which the host server is connected.

NetApp recommends making sure that NetApp storage controllers do not receive Ethernet flow-control packets. This can generally be done by setting the switch ports to which the controller is attached, but some switch hardware has limitations that might require client-side changes instead.

MTU Sizes

The use of jumbo frames has been shown to offer some performance improvement in 1Gb networks by reducing CPU and network overhead, but the benefit is not usually significant.

NetApp recommends implementing jumbo frames when possible, both to realize any potential performance benefits and to future-proof the solution.

Using jumbo frames in a 10Gb network is almost mandatory. This is because most 10Gb implementations reach a packets-per-second limit without jumbo frames before they reach the 10Gb mark. Using jumbo frames improves efficiency in TCP/IP processing because it allows the OS, server, NICs, and the storage system to process fewer but larger packets. The performance improvement varies from NIC to NIC, but it is significant.

For jumbo-frame implementations, there is the common but incorrect belief that all connected devices must support jumbo frames and that the MTU size must match end-to-end. Instead, the two network end points negotiate the highest mutually acceptable frame size when establishing a connection. In a typical environment, a network switch is set to an MTU size of 9216, the NetApp controller is set to 9000, and the clients are set to a mix of 9000 and 1514. Clients that can support an MTU of 9000 can use jumbo frames, and clients that can only support 1514 can negotiate a lower value.

Problems with this arrangement are rare in a completely switched environment. However, take care in a routed environment that no intermediate router is forced to fragment jumbo frames.

NetApp recommends configuring the following:

- Jumbo frames are desirable but not required with 1Gb Ethernet (GbE).
- Jumbo frames are required for maximum performance with 10GbE and faster.

TCP parameters

Three settings are often misconfigured: TCP timestamps, selective acknowledgment (SACK), and TCP window scaling. Many out-of-date documents on the Internet recommend disabling one or more of these parameters to improve performance. There was some merit to this recommendation many years ago when CPU capabilities were much lower and there was a benefit to reducing the overhead on TCP processing whenever possible.

However, with modern OSs, disabling any of these TCP features usually results in no detectable benefit while also potentially damaging performance. Performance damage is especially likely in virtualized networking environments because these features are required for efficient handling of packet loss and changes in network quality.

NetApp recommends enabling TCP timestamps, SACK, and TCP window scaling on the host, and all three of these parameters should be on by default in any current OS.

FC SAN configuration

Configuring FC SAN for Oracle databases is primarily about following everyday SAN best practices.

This includes typical planning measures such as ensuring sufficient bandwidth exists on the SAN in between the host and storage system, checking that all SAN paths exist between all required devices, using the FC port settings required by your FC switch vendor, avoiding ISL contention, and using proper SAN fabric monitoring.

Zoning

An FC zone should never contain more than one initiator. Such an arrangement might appear to work initially, but crosstalk between initiators eventually interferes with performance and stability.

Multitarget zones are generally regarded as safe, although in rare circumstances the behavior of FC target ports from different vendors has caused problems. For example, avoid including the target ports from both a NetApp and an non-NetApp storage array in the same zone. In addition, placing a NetApp storage system and a tape device in the same zone is even more likely to cause problems.

Direct-connect networking

Storage administrators sometimes prefer to simplify their infrastructures by removing network switches from the configuration. This can be supported in some scenarios.

iSCSI and NVMe/TCP

A host using iSCSI or NVMe/TCP can be directly connected to a storage system and operate normally. The reason is pathing. Direct connections to two different storage controllers results in two independent paths for data flow. The loss of path, port, or controller does not prevent the other path from being used.

NFS

Direct-connected NFS storage can be used, but with a significant limitation - failover will not work without a significant scripting effort, which would be the responsibility of the customer.

The reason nondisruptive failover is complicated with direct-connected NFS storage is the routing that occurs on the local OS. For example, assume a host has an IP address of 192.168.1.1/24 and is directly connected to an ONTAP controller with an IP address of 192.168.1.50/24. During failover, that 192.168.1.50 address can fail over to the other controller, and it will be available to the host, but how does the host detect its presence? The original 192.168.1.1 address still exists on the host NIC that no longer connects to an operational system. Traffic destined for 192.168.1.50 would continue to be sent to an inoperable network port.

The second OS NIC could be configured as 19 2.168.1.2 and would be capable of communicating with the failed over 192.168.1.50 address, but the local routing tables would have a default of using one **and only one** address to communicate with the 192.168.1.0/24 subnet. A sysadmin could create a scripting framework that would detect a failed network connection and alter the local routing tables or bring interfaces up and down. The exact procedure would depend on the OS in use.

In practice, NetApp customers do have direct-connected NFS, but normally only for workloads where IO pauses during failovers are acceptable. When hard mounts are used, there should not be any IO errors during such pauses. The IO should hang until services are restored, either by a failback or manual intervention to move IP addresses between NICs on the host.

FC direct connect

It is not possible to directly connect a host to an ONTAP storage system using the FC protocol. The reason is the use of NPIV. The WWN that identifies an ONTAP FC port to the FC network uses a type of virtualization called NPIV. Any device connected to an ONTAP system must be able to recognize an NPIV WWN. There are no current HBA vendors who offer an HBA that can be installed in a host that would be able to support an NPIV target.

Network configuration on ASA r2 systems

Logical interfaces

Oracle databases need access to storage. Logical interfaces (LIFs) are the network plumbing that connects a storage virtual machine (SVM) to the network and in turn to the database. Proper LIF design is required to ensure sufficient bandwidth exists for each database workload, and failover does not result in a loss of storage services.

This section provides an overview of key LIF design principles for ASA r2 systems, which are optimized for SAN-only environments. For more comprehensive documentation, see the ONTAP Network Management documentation. As with other aspects of database architecture, the best options for storage virtual machine (SVM, known as a vserver at the CLI) and logical interface (LIF) design depend heavily on scaling requirements and business needs.

Consider the following primary topics when building a LIF strategy:

- **Performance.** Is the network bandwidth sufficient for Oracle workloads?
- **Resiliency.** Are there any single points of failure in the design?
- **Manageability.** Can the network be scaled nondisruptively?

These topics apply to the end-to-end solution, from the host through the switches to the storage system.

LIF types

There are multiple LIF types. ONTAP documentation on LIF types provide more complete information on this topic, but from a functional perspective LIFs can be divided into the following groups:

- **Cluster and node management LIFs.** LIFs used to manage the storage cluster.
- **SVM management LIFs.** Interfaces that permit access to an SVM through the REST API or ONTAPI (also known as ZAPI) for functions such as snapshot creation or volume resizing. Products such as SnapManager for Oracle (SMO) must have access to an SVM management LIF.
- **Data LIFs.** Interfaces for SAN protocols only: FC, iSCSI, NVMe/FC, NVMe/TCP. NAS protocols (NFS, SMB/CIFS) are not supported on ASA r2 systems.

> It is not possible to configure an interface for both iSCSI (or NVMe/TCP) and management traffic, despite the fact that both use an IP protocol. A separate management LIF is required in iSCSI or NVMe/TCP environments. For resiliency and performance, configure multiple SAN data LIFs per protocol per node and distribute them across different physical ports and fabrics. Unlike AFF/FAS systems, ASA r2 does not allow NFS or SMB traffic, so there is no option to repurpose a NAS data LIF for management.

SAN LIF design

LIF design in a SAN environment is relatively simple for one reason: multipathing. All modern SAN implementations allow a client to access data over multiple, independent, network paths and select the best path or paths for access. As a result, performance with respect to LIF design is simpler to address because SAN clients automatically load-balance I/O across the best available paths.

If a path becomes unavailable, the client automatically selects a different path. The resulting simplicity of design makes SAN LIFs generally more manageable. This does not mean that a SAN environment is always

more easily managed, because there are many other aspects of SAN storage that are much more complicated than NFS. It simply means that SAN LIF design is easier.

Performance

The most important consideration with LIF performance in a SAN environment is bandwidth. For example, a two-node ASA r2 cluster with two 32Gb FC ports per node allows up to 64Gb of bandwidth to/from each node. Similarly, for NVMe/TCP or iSCSI, ensure sufficient 25GbE or 100GbE connectivity for Oracle workloads.

Resiliency

SAN LIFs do not fail over in the same way NAS LIFs do. ASA r2 systems rely on host multipathing (MPIO/ALUA) for resiliency. If a SAN LIF becomes unavailable due to controller failover, the client's multipathing software detects the loss of a path and redirects I/O to an alternate path. ASA r2 may perform LIF relocation after a short delay to restore full path availability, but this does not interrupt I/O because active paths already exist on the partner node. The failover process occurs in order to restore host access on all defined ports.

Manageability

There is no need to migrate a LIF in a SAN environment when volumes are relocated within the HA pair. That is because, after the volume move has completed, ONTAP sends a notification to the SAN about a change in paths, and the SAN clients automatically reoptimize. LIF migration with SAN is primarily associated with major physical hardware changes. For example, if a nondisruptive upgrade of the controllers is required, a SAN LIF is migrated to the new hardware. If an FC port is found to be faulty, a LIF can be migrated to an unused port.

Design recommendations

NetApp makes the following recommendations for ASA r2 SAN environments:

- Do not create more paths than are required. Excessive numbers of paths make overall management more complicated and can cause problems with path failover on some hosts. Furthermore, some hosts have unexpected path limitations for configurations such as SAN booting.
- Very few configurations should require more than four paths to a LUN. The value of having more than two nodes advertising paths to LUNs is limited because the aggregate hosting a LUN is inaccessible if the node that owns the LUN and its HA partner fail. Creating paths on nodes other than the primary HA pair is not helpful in such a situation.
- Although the number of visible LUN paths can be managed by selecting which ports are included in FC zones, it is generally easier to include all potential target points in the FC zone and control LUN visibility at the ONTAP level.
- Use selective LUN mapping (SLM) feature, which is enabled by default. With SLM, any new LUN is automatically advertised from the node that owns the underlying aggregate and the node's HA partner. This arrangement avoids the need to create port sets or configure zoning to limit port accessibility. Each LUN is available on the minimum number of nodes required for both optimal performance and resiliency.
- In the event a LUN must be migrated outside of the two controllers, the additional nodes can be added with the `lun mapping add-reporting-nodes` command so that the LUNs are advertised on the new nodes. Doing so creates additional SAN paths to the LUNs for LUN migration. However, the host must perform a discovery operation to use the new paths.
- Do not be overly concerned about indirect traffic. It is best to avoid indirect traffic in a very I/O-intensive environment for which every microsecond of latency is critical, but the visible performance effect is negligible for typical workloads.

TCP/IP and ethernet configuration

Many Oracle on ASA r2 ONTAP customers use ethernet, the network protocol of iSCSI and NVMe/TCP.

Host OS settings

Most application vendor documentation include specific TCP and ethernet settings intended to ensure the application is working optimally. These same settings are usually sufficient to also deliver optimal IP-based storage performance.

Ethernet flow control

This technology allows a client to request that a sender temporarily stop data transmission. This is usually done because the receiver is unable to process incoming data quickly enough. At one time, requesting that a sender cease transmission was less disruptive than having a receiver discard packets because buffers were full. This is no longer the case with the TCP stacks used in OSs today. In fact, flow control causes more problems than it solves.

Performance problems caused by Ethernet flow control have been increasing in recent years. This is because Ethernet flow control operates at the physical layer. If a network configuration permits any host OS to send an Ethernet flow control request to a storage system, the result is a pause in I/O for all connected clients. Because an increasing number of clients are served by a single storage controller, the likelihood of one or more of these clients sending flow control requests increases. The problem has been seen frequently at customer sites with extensive OS virtualization.

A NIC on a NetApp system should not receive flow-control requests. The method used to achieve this result varies based on the network switch manufacturer. In most cases, flow control on an Ethernet switch can be set to `receive desired` or `receive on`, which means that a flow control request is not forwarded to the storage controller. In other cases, the network connection on the storage controller might not allow flow-control disabling. In these cases, the clients must be configured to never send flow control requests, either by changing to the NIC configuration on the host server itself or the switch ports to which the host server is connected.

For ASA r2 systems, which are SAN-only, Ethernet flow control considerations apply primarily to iSCSI and NVMe/TCP traffic.

NetApp recommends making sure that NetApp ASA r2 storage controllers do not receive Ethernet flow-control packets. This can generally be done by setting the switch ports to which the controller is attached, but some switch hardware has limitations that might require client-side changes instead.

MTU Sizes

The use of jumbo frames has been shown to offer some performance improvement in 1Gb networks by reducing CPU and network overhead, but the benefit is not usually significant.

NetApp recommends implementing jumbo frames when possible, both to realize any potential performance benefits and to future-proof the solution.

For ASA r2 systems, which are SAN-only, jumbo frames apply only to Ethernet-based SAN protocols (iSCSI and NVMe/TCP).

Using jumbo frames in a 10Gb network is almost mandatory. This is because most 10Gb implementations reach a packets-per-second limit without jumbo frames before they reach the 10Gb mark. Using jumbo frames improves efficiency in TCP/IP processing because it allows the OS, server, NICs, and the storage system to process fewer but larger packets. The performance improvement varies from NIC to NIC, but it is significant.

For jumbo-frame implementations, there is the common but incorrect belief that all connected devices must support jumbo frames and that the MTU size must match end-to-end. Instead, the two network end points negotiate the highest mutually acceptable frame size when establishing a connection. In a typical environment, a network switch is set to an MTU size of 9216, the NetApp controller is set to 9000, and the clients are set to a mix of 9000 and 1514. Clients that can support an MTU of 9000 can use jumbo frames, and clients that can only support 1514 can negotiate a lower value.

Problems with this arrangement are rare in a completely switched environment. However, take care in a routed environment that no intermediate router is forced to fragment jumbo frames.

NetApp recommends configuring the following for ASA r2 SAN environments:

- Jumbo frames are desirable but not required with 1GbE.
- Jumbo frames are required for maximum performance with 10GbE and faster for iSCSI and NVMe/TCP traffic.

TCP parameters

Three settings are often misconfigured: TCP timestamps, selective acknowledgment (SACK), and TCP window scaling. Many out-of-date documents on the Internet recommend disabling one or more of these parameters to improve performance. There was some merit to this recommendation many years ago when CPU capabilities were much lower and there was a benefit to reducing the overhead on TCP processing whenever possible.

However, with modern OSs, disabling any of these TCP features usually results in no detectable benefit while also potentially damaging performance. Performance damage is especially likely in virtualized networking environments because these features are required for efficient handling of packet loss and changes in network quality.

NetApp recommends enabling TCP timestamps, SACK, and TCP window scaling on the host, and all three of these parameters should be on by default in any current OS.

FC SAN configuration

Configuring FC SAN for Oracle databases on ASA r2 systems is primarily about following standard SAN best practices.

ASA r2 is optimized for SAN-only workloads, so the principles remain the same as AFF/FAS, with a focus on performance, resiliency, and simplicity. This includes typical planning measures such as ensuring sufficient bandwidth exists on the SAN in between the host and storage system, checking that all SAN paths exist between all required devices, using the FC port settings required by your FC switch vendor, avoiding ISL contention, and using proper SAN fabric monitoring.

Zoning

An FC zone should never contain more than one initiator. Such an arrangement might appear to work initially, but crosstalk between initiators eventually interferes with performance and stability.

Multitarget zones are generally regarded as safe, although in rare circumstances the behavior of FC target

ports from different vendors has caused problems. For example, avoid including the target ports from both a NetApp and an non-NetApp storage array in the same zone. In addition, placing a NetApp storage system and a tape device in the same zone is even more likely to cause problems.

- ASA r2 uses Storage Availability Zones instead of aggregates, but this does not change FC zoning principles.
- Multipathing (MPIO) remains the primary resiliency mechanism; however, for ASA r2 systems that support symmetric active-active multipathing, all paths to a LUN are active and used for I/O simultaneously.

Direct-connect networking

Storage administrators sometimes prefer to simplify their infrastructures by removing network switches from the configuration. This can be supported in some scenarios.

iSCSI and NVMe/TCP

A host using iSCSI or NVMe/TCP can be directly connected to an ASA r2 storage system and operate normally. The reason is pathing. Direct connections to two different storage controllers results in two independent paths for data flow. The loss of path, port, or controller does not prevent the other path from being used, provided multipathing is configured correctly.

FC direct connect

It is not possible to directly connect a host to an ASA r2 storage system using the FC protocol. The reason is the same as with AFF/FAS systems, use of NPIV. The WWN that identifies an ONTAP FC port to the FC network uses a type of virtualization called NPIV. Any device connected to an ONTAP system must be able to recognize an NPIV WWN. There are no current HBA vendors who offer an HBA that can be installed in a host that would be able to support an NPIV target.

Storage configuration on AFF/FAS systems

FC SAN

LUN Alignment

LUN alignment refers to optimizing I/O with respect to the underlying file system layout.

On a ONTAP system, storage is organized in 4KB units. A database or file system 8KB block should map to exactly two 4KB blocks. If an error in LUN configuration shifts the alignment by 1KB in either direction, each 8KB block would exist on three different 4KB storage blocks rather than two. This arrangement would cause increased latency and cause additional I/O to be performed within the storage system.

Alignment also affects LVM architectures. If a physical volume within a logical volume group is defined on the whole drive device (no partitions are created), the first 4KB block on the LUN aligns with the first 4KB block on the storage system. This is a correct alignment. Problems arise with partitions because they shift the starting location where the OS uses the LUN. As long as the offset is shifted in whole units of 4KB, the LUN is aligned.

In Linux environments, build logical volume groups on the whole drive device. When a partition is required, check alignment by running `fdisk -u` and verifying that the start of each partition is a multiple of eight. This means that the partition starts at a multiple of eight 512-byte sectors, which is 4KB.

Also see the discussion about compression block alignment in the section Efficiency. Any layout that is aligned with 8KB compression block boundaries is also aligned with 4KB boundaries.

Misalignment warnings

Database redo/transaction logging normally generates unaligned I/O that can cause misleading warnings about misaligned LUNs on ONTAP.

Logging performs a sequential write of the log file with writes of varying size. A log write operation that does not align to 4KB boundaries does not ordinarily cause performance problems because the next log write operation completes the block. The result is that ONTAP is able to process almost all writes as complete 4KB blocks, even though the data in some 4KB blocks was written in two separate operations.

Verify alignment by using by using utilities such as `sio` or `dd` that can generate I/O at a defined block size. The I/O alignment statistics on the storage system can be viewed with the `stats` command. See WAFL alignment verification for more information.

Alignment in Solaris environments is more complicated. Refer to ONTAP SAN Host Configuration for more information.

Caution
In Solaris x86 environments, take additional care about proper alignment because most configurations have several layers of partitions. Solaris x86 partition slices usually exist on top of a standard master boot record partition table.

LUN sizing and LUN count

Selecting the optimal LUN size and the number of LUNs to be used is critical for optimal performance and manageability of Oracle databases.

A LUN is a virtualized object on ONTAP that exists across all of the drives in the hosting aggregate. As a result, the performance of the LUN is unaffected by its size because the LUN draws on the full performance potential of the aggregate no matter which size is chosen.

As a matter of convenience, customers might wish to use a LUN of a particular size. For example, if a database is built on an LVM or Oracle ASM diskgroup composed of two LUNs of 1TB each, then that diskgroup must be grown in increments of 1TB. It might be preferable to build the diskgroup from eight LUNs of 500GB each so that the diskgroup can be increased in smaller increments.

The practice of establishing a universal standard LUN size is discouraged because doing so can complicate manageability. For example, a standard LUN size of 100GB might work well when a database or datastore is in the range of 1TB to 2TB, but a database or datastore of 20TB in size would require 200 LUNs. This means that server reboot times are longer, there are more objects to manage in the various UIs, and products such as SnapCenter must perform discovery on many objects. Using fewer, larger LUNs avoids such problems.

- The LUN count is more important than the LUN size.
- LUN size is mostly controlled by LUN count requirements.
- Avoid creating more LUNs than required.

LUN count

Unlike the LUN size, the LUN count does affect performance. Application performance often depends on the ability to perform parallel I/O through the SCSI layer. As a result, two LUNs offer better performance than a

single LUN. Using an LVM such as Veritas VxVM, Linux LVM2, or Oracle ASM is the simplest method to increase parallelism.

NetApp customers have generally experienced minimal benefit from increasing the number of LUNs beyond sixteen, although the testing of 100%-SSD environments with very heavy random I/O has demonstrated further improvement up to 64 LUNs.

NetApp recommends the following:

In general, four to sixteen LUNs are sufficient to support the I/O needs of any given database workload. Less than four LUNs might create performance limitations because of limitations in host SCSI implementations.

LUN placement

Optimal placement of database LUNs within ONTAP volumes primarily depends on how various ONTAP features will be used.

Volumes

One common point of confusion with customers new to ONTAP is the use of FlexVols, commonly referred to as simply "volumes".

A volume is not a LUN. These terms are used synonymously with many other vendor products, including cloud providers. ONTAP volumes are simply management containers. They do not serve data by themselves, nor do they occupy space. They are containers for files or LUNs and exist to improve and simplify manageability, especially at scale.

Volumes and LUNs

Related LUNs are normally co-located in a single volume. For example, a database that requires 10 LUNs would typically have all 10 LUNs placed on the same volume.

- Using a 1:1 ratio of LUNs to volumes, meaning one LUN per volume, is **not** a formal best practice.
- Instead, volumes should be viewed as containers for workloads or datasets. There may be a single LUN per volume, or there could be many. The right answer depends on manageability requirements.
- Scattering LUNs across an unnecessary number of volumes can lead to additional overhead and scheduling problems for operations such as snapshot operations, excessive numbers of objects displayed in the UI, and result in reaching platform volume limits before the LUN limit is reached.

Volumes, LUNs, and snapshots

Snapshot policies and schedules are placed on the volume, not the LUN. A dataset that consists of 10 LUNs would require only a single snapshot policy when those LUNs are co-located in the same volume.

Additionally, co-locating all related LUNs for a given dataset in a single volume delivers atomic snapshot operations. For example, a database that resided on 10 LUNs, or a VMware-based application environment consisting of 10 different OSs could be protected as a single, consistent object if the underlying LUNs are all placed on a single volume. If they are placed on different volumes, the snapshots may or may not be 100% in sync, even if scheduled at the same time.

In some cases, a related set of LUNs might need to be split into two different volumes because of recovery requirements. For example, a database might have four LUNs for datafiles and two LUNs for logs. In this case, a datafile volume with 4 LUNs and a log volume with 2 LUNs might be the best option. The reason is independent recoverability. For example, the datafile volume could be selectively restored to an earlier state, meaning all four LUNs would be reverted to the state of the snapshot, while the log volume with its critical data would be unaffected.

Volumes, LUNs, and SnapMirror

SnapMirror policies and operations are, like snapshot operations, performed on the volume, not the LUN.

Co-locating related LUNs in a single volume allows you to create a single SnapMirror relationship and update all contained data with a single update. As with snapshots, the update will also be an atomic operation. The SnapMirror destination would be guaranteed to have a single point-in-time replica of the source LUNs. If the LUNs were spread across multiple volumes, the replicas may or may not be consistent with one another.

Volumes, LUNs, and QoS

While QoS can be selectively applied to individual LUNs, it is usually easier to set it at the volume level. For example, all of the LUNs used by the guests in a given ESX server could be placed on a single volume, and then an ONTAP adaptive QoS policy could be applied. The result is a self-scaling IOPS-per-TB limit that applies to all LUNs.

Likewise, if a database required 100K IOPS and occupied 10 LUNs, it would be easier to set a single 100K IOPS limit on a single volume than to set 10 individual 10K IOPS limits, one on each LUN.

Multi-volume layouts

There are some cases where distributing LUNs across multiple volumes may be beneficial. The primary reason is controller striping. For example, an HA storage system might be hosting a single database where the full processing and caching potential of each controller is required. In this case, a typical design would be to place half of the LUNs in a single volume on controller 1, and the other half of the LUNs in a single volume on controller 2.

Similarly, controller striping might be used for load balancing. An HA system that hosted 100 databases of 10 LUNs each might be designed where each database receives a 5-LUN volume on each of the two controllers. The result is guaranteed symmetric loading of each controller as additional databases are provisioned.

None of these examples involve a 1:1 volume to LUN ratio, though. The goal remains to optimize manageability by co-locating related LUNs in volumes.

One example where a 1:1 LUN to volume ratio makes sense is containerization, where each LUN might really represent a single workload and need to be each managed on an individual basis. In such cases, a 1:1 ratio may be optimal.

LUN resizing and LVM resizing

When a SAN-based file system has reached its capacity limit, there are two options for increasing the space available:

- Increase the size of the LUNs
- Add a LUN to an existing volume group and grow the contained logical volume

Although LUN resizing is an option to increase capacity, it is generally better to use an LVM, including Oracle

ASM. One of the principal reasons LVMs exist is to avoid the need for a LUN resize. With an LVM, multiple LUNs are bonded together into a virtual pool of storage. The logical volumes carved out of this pool are managed by the LVM and can be easily resized. An additional benefit is the avoidance of hotspots on a particular drive by distributing a given logical volume across all available LUNs. Transparent migration can usually be performed by using the volume manager to relocate the underlying extents of a logical volume to new LUNs.

LVM striping

LVM striping refers to distributing data across multiple LUNs. The result is dramatically improved performance for many databases.

Before the era of flash drives, striping was used to help overcome the performance limitations of spinning drives. For example, if an OS needs to perform a 1MB read operation, reading that 1MB of data from a single drive would require a lot of drive head seeking and reading as the 1MB is slowly transferred. If that 1MB of data was striped across 8 LUNs, the OS could issue eight 128K read operations in parallel and reduce the time required to complete the 1MB transfer.

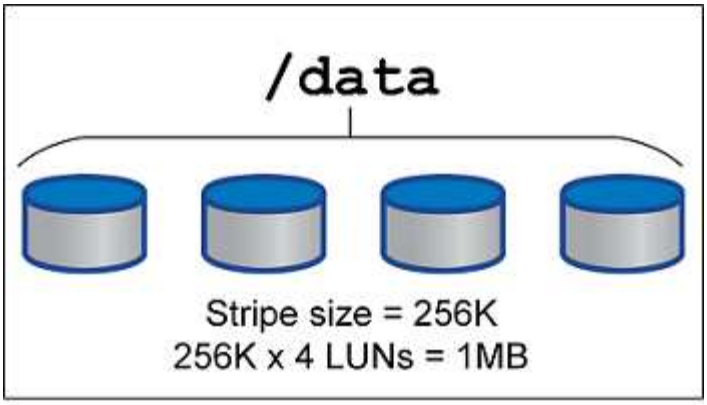
Striping with spinning drives was more difficult because the I/O pattern had to be known in advance. If the striping wasn't correctly tuned for the true I/O patterns, striped configurations could damage performance. With Oracle databases, and especially with all-flash configurations, striping is much easier to configure and has been proven to dramatically improve performance.

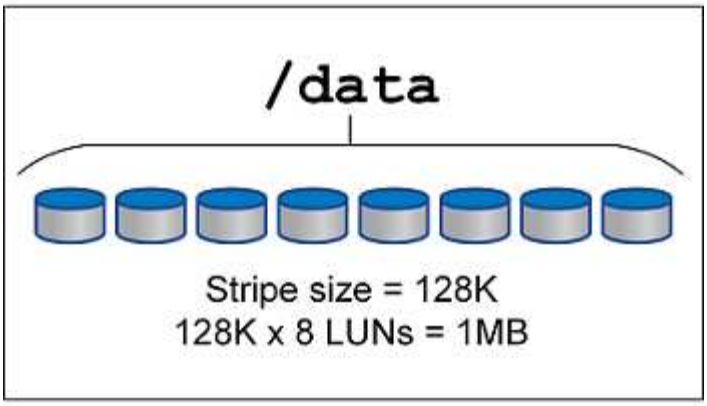
Logical volume managers such as Oracle ASM stripe by default, but native OS LVM do not. Some of them bond multiple LUNs together as a concatenated device, which results in datafiles that exist on one and only one LUN device. This causes hot spots. Other LVM implementations default to distributed extents. This is similar to striping, but it's coarser. The LUNs in the volume group are sliced into large pieces, called extents and typically measured in many megabytes, and the logical volumes are then distributed across those extents. The result is random I/O against a file should be well distributed across LUNs, but sequential I/O operations are not as efficient as they could be.

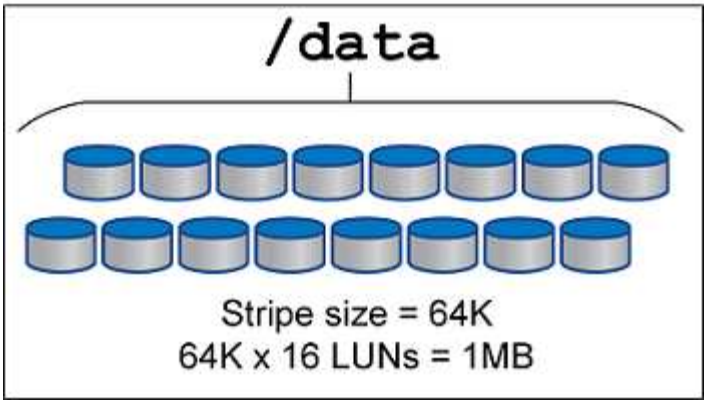
Performance-intensive application I/O is nearly always either (a) in units of the basic block size or (b) one megabyte.

The primary goal of a striped configuration is to ensure that single-file I/O can be performed as a single unit, and multiblock I/Os, which should be 1MB in size, can be parallelized evenly across all LUNs in the striped volume. This means that the stripe size must not be smaller than the database block size, and the stripe size multiplied by the number of LUNs should be 1MB.

The following figure shows three possible options for stripe size and width tuning. The number of LUNs is selected to meet performance requirements as described above, but in all cases the total data within a single stripe is 1MB.







NFS

Overview

NetApp has been providing enterprise-grade NFS storage for over 30 years, and its use is growing with the push toward cloud-based infrastructures because of it's simplicity.

The NFS protocol includes multiple versions with varying requirements. For a complete description of NFS configuration with ONTAP, please see TR-4067 NFS on ONTAP Best Practices. The following sections cover some of the more critical requirements and common user errors.

NFS versions

The operating system NFS client must be supported by NetApp.

- NFSv3 is supported with OSs that follow the NFSv3 standard.

- NFSv3 is supported with the Oracle dNFS client.
- NFSv4 is supported with all OSs that follow the NFSv4 standard.
- NFSv4.1 and NFSv4.2 require specific OS support. Consult the NetApp IMT for supported OSs.
- Oracle dNFS support for NFSv4.1 requires Oracle 12.2.0.2 or higher.

The NetApp support matrix for NFSv3 and NFSv4 does not include specific operating systems. All OSs that obey the RFC are generally supported. When searching the online IMT for NFSv3 or NFSv4 support, do not select a specific OS because there will be no matches displayed. All OSs are implicitly supported by the general policy.

Linux NFSv3 TCP slot tables

TCP slot tables are the NFSv3 equivalent of host bus adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables.

Run the `sysctl -a | grep tcp.*.slot_table` command, and observe the following parameters:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

All Linux systems should include `sunrpc.tcp_slot_table_entries`, but only some include `sunrpc.tcp_max_slot_table_entries`. They should both be set to 128.

Failure to set these parameters may have significant effects on performance. In some cases, performance is limited because the linux OS is not issuing sufficient I/O. In other cases, I/O latencies increases as the linux OS attempts to issue more I/O than can be serviced.

ADR and NFS

Some customers have reported performance problems resulting from an excessive amount of I/O on data in the `ADR` location. The problem does not generally occur until a lot of performance data has accumulated. The reason for the excessive I/O is unknown, but this problem appears to be a result of Oracle processes repeatedly scanning the target directory for changes.

Removal of the `noac` and/or `actimeo=0` mount options allows host OS caching to occur and reduces storage I/O levels.

NetApp recommends to not place `ADR` data on a file system with `noac` or `actimeo=0` because performance problems are likely. Separate `ADR` data into a different mount point if necessary.

nfs-rootonly and mount-rootonly

ONTAP includes an NFS option called `nfs-rootonly` that controls whether the server accepts NFS traffic connections from high ports. As a security measure, only the root user is permitted to open TCP/IP connections using a source port below 1024 because such ports are normally reserved for OS use, not user processes. This restriction helps ensure that NFS traffic is from an actual operating system NFS client, and not a malicious process emulating an NFS client. The Oracle dNFS client is a userspace driver, but the process runs as root, so it is generally not required to change the value of `nfs-rootonly`. The connections is made from low ports.

The `mount-rootonly` option only applies to NFSv3. It controls whether the RPC MOUNT call be accepted from ports greater than 1024. When dNFS is used, the client is again running as root, so it able to open ports below 1024. This parameter has no effect.

Processes opening connections with dNFS over NFS versions 4.0 and higher do not run as root and therefore require ports over 1024. The `nfs-rootonly` parameter must be set to disabled for dNFS to complete the connection.

If `nfs-rootonly` is enabled, the result is a hang during the mount phase opening dNFS connections. The sqlplus output looks similar to this:

```
SQL>startup
ORACLE instance started.
Total System Global Area 4294963272 bytes
Fixed Size                  8904776 bytes
Variable Size             822083584 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7868416 bytes
```

The parameter can be changed as follows:

```
Cluster01::> nfs server modify -nfs-rootonly disabled
```

> In rare situations, you might need to change both nfs-rootonly and mount-rootonly to disabled. If a server is managing an extremely large number of TCP connections, it is possible that no ports below 1024 is available, and the OS is forced to use higher ports. These two ONTAP parameters would need to be changed to allow the connection to complete.

NFS export polices: superuser and setuid

If Oracle binaries are located on an NFS share, the export policy must include superuser and setuid permissions.

Shared NFS exports used for generic file services such as user home directories usually squash the root user. This means a request from the root user on a host that has mounted a filesystem is remapped as a different user with lower privileges. This helps secure data by preventing a root user on a particular server from accessing data on the shared server. The setuid bit can also be a security risk on a shared environment. The setuid bit allows a process to be run as a different user than the user invoking the command. For example, a shell script that was owned by root with the setuid bit runs as root. If that shell script could be changed by other users, any non-root user could issue a command as root by updating the script.

The Oracle binaries include files owned by root and use the setuid bit. If Oracle binaries are installed on an NFS share, the export policy must include the appropriate superuser and setuid permissions. In the example below, the rule includes both `allow-suid` and permits `superuser` (root) access for NFS clients using system authentication.

```
Cluster01::> export-policy rule show -vserver vserver1 -policyname orabin
-fields allow-suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
vserver1  orabin     1         sys       true
```

NFSv4/4.1 configuration

For most applications, there is very little difference between NFSv3 and NFSv4. Application I/O is usually very simple I/O and does not benefit significantly from some of the advanced features available in NFSv4. Higher versions of NFS should not be viewed as an “upgrade” from a database storage perspective, but instead as versions of NFS that include additional features. For example, if the end-to-end security of kerberos privacy mode (krb5p) is required, then NFSv4 is required.

NetApp recommends using NFSv4.1 if NFSv4 capabilities are required. There are some functional enhancements to the NFSv4 protocol in NFSv4.1 that improve resiliency in certain edge cases.

Switching to NFSv4 is more complicated than simply changing the mount options from vers=3 to vers=4.1. A more complete explanation of NFSv4 configuration with ONTAP, including guidance on configuring the OS, see TR-4067 NFS on ONTAP best practices. The following sections of this TR explain some of the basic requirements for using NFSv4.

NFSv4 domain

A complete explanation of NFSv4/4.1 configuration is beyond the scope of this document, but one commonly encountered problem is a mismatch in domain mapping. From a sysadmin point of view, the NFS file systems appear to behave normally, but applications report errors about permissions and/or setuid on the certain files. In some cases, administrators have incorrectly concluded that the permissions of the application binaries have been damaged and have run chown or chmod commands when the actual problem was the domain name.

The NFSv4 domain name is set on the ONTAP SVM:

```
Cluster01::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
--------- ------------
vserver1  my.lab
```

The NFSv4 domain name on the host is set in `/etc/idmap.cfg`

```
[root@host1 etc]# head /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = my.lab
```

The domain names must match. If they do not, mapping errors similar to the following appear in `/var/log/messages`:

```
Apr 12 11:43:08 host1 nfsidmap[16298]: nss_getpwnam: name 'root@my.lab'
does not map into domain 'default.com'
```

Application binaries, such as Oracle database binaries, include files owned by root with the setuid bit, which means a mismatch in the NFSv4 domain names causes failures with Oracle startup and a warning about the ownership or permissions of a file called `oradism`, which is located in the `$ORACLE_HOME/bin` directory. It should appear as follows:

```
[root@host1 etc]# ls -l /orabin/product/19.3.0.0/dbhome_1/bin/oradism
-rwsr-x--- 1 root oinstall 147848 Apr 17  2019
/orabin/product/19.3.0.0/dbhome_1/bin/oradism
```

If this file appears with ownership of nobody, there may be an NFSv4 domain mapping problem.

```
[root@host1 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17  2019 oradism
```

To fix this, check the `/etc/idmap.cfg` file against the v4-id-domain setting on ONTAP and ensure they are consistent. If they are not, make the required changes, run `nfsidmap -c`, and wait a moment for the changes to propagate. The file ownership should then be properly recognized as root. If a user had attempted to run `chown root` on this file before the NFS domains configure was corrected, it might be necessary to run `chown root` again.

Oracle direct NFS (dNFS)

Oracle databases can use NFS in two ways.

First, it can use a filesystem mounted using the native NFS client that is part of the operating system. This is sometimes called kernel NFS, or kNFS. The NFS filesystem is mounted and used by the Oracle database exactly the same as any other application would use an NFS filesystem.

The second method is Oracle Direct NFS (dNFS). This is an implementation of the NFS standard within the Oracle database software. It does not change the way Oracle databases are configured or managed by the DBA. As long as the storage system itself has the correct settings, the use of dNFS should be transparent to the DBA team and end users.

A database with the dNFS feature enabled still has the usual NFS filesystems mounted. Once the database is open, the Oracle database opens a set of TCP/IP sessions and performs NFS operations directly.

Direct NFS

The primary value of Oracle's Direct NFS is to bypass the host NFS client and perform NFS file operations directly on an NFS server. Enabling it only requires changing the Oracle Disk Manager (ODM) library. Instructions for this process are provided in the Oracle documentation.

Using dNFS results in a significant improvement in I/O performance and decreases the load on the host and the storage system because I/O is performed in the most efficient way possible.

In addition, Oracle dNFS includes an **option** for network interface multipathing and fault-tolerance. For example, two 10Gb interfaces can be bound together to offer 20Gb of bandwidth. A failure of one interface results in I/O being retried on the other interface. The overall operation is very similar to FC multipathing. Multipathing was common years ago when 1Gb ethernet was the most common standard. A 10Gb NIC is sufficient for most Oracle workloads, but if more is required 10Gb NICs can be bonded.

When dNFS is used, it is critical that all patches described in Oracle Doc 1495104.1 are installed. If a patch cannot be installed, the environment must be evaluated to make sure that the bugs described in that document do not cause problems. In some cases, an inability to install the required patches prevents the use of dNFS.

Do not use dNFS with any type of round-robin name resolution, including DNS, DDNS, NIS or any other method. This includes the DNS load balancing feature available in ONTAP. When an Oracle database using dNFS resolves a host name to an IP address it must not change on subsequent lookups. This can result in Oracle database crashes and possible data corruption.

Enabling dNFS

Oracle dNFS can work with NFSv3 with no configuration required beyond enabling the dNFS library (See Oracle documentation for the specific command required) but if dNFS is unable to establish connectivity, it can silently revert back to the kernel NFS client. If this happens, performance can be severely affected.

If you wish to use dNFS multiplexing across multiple interface, with NFSv4.X, or use encryption, you must configure an oranfstab file. The syntax is extremely strict. Small errors in the file can result in startup hanging or bypassing the oranfstab file.

At the time of writing, dNFS multipathing does not work with NFSv4.1 with recent versions of Oracle Database. An oranfstab file that specifies NFSv4.1 as a protocol can only use a single path statement for a given export. The reason is ONTAP does not support clientID trunking. Oracle Database patches to resolve this limitation may be available in the future.

The only way to be certain dNFS is operating as expected is to query the v$dnfs tables.

Below is a sample oranfstab file located at /etc. This is one of multiple locations an oranfstab file can be placed.

```
[root@jfs11 trace]# cat /etc/oranfstab
server: NFSv3test
path: jfs_svmdr-nfs1
path: jfs_svmdr-nfs2
export: /dbf mount: /oradata
export: /logs mount: /logs
nfs_version: NFSv3
```

The first step is to check that dNFS is operational for the specified filesystems:

```
SQL> select dirname,nfsversion from v$dnfs_servers;

DIRNAME
------------------------------------
NFSVERSION
----------------
/logs
NFSv3.0

/dbf
NFSv3.0
```

This output indicates that dNFS is in use with these two filesystems, but it does **not** mean that oranfstab is operational. If an error was present, dNFS would have autodiscovered the host's NFS filesystems and you may still see the same output from this command.

Multipathing can be checked as follows:

```
SQL> select svrname,path,ch_id from v$dnfs_channels;

SVRNAME
------------------------------------
PATH
------------------------------------
     CH_ID
----------
NFSv3test
jfs_svmdr-nfs1
         0

NFSv3test
jfs_svmdr-nfs2
         1
```

```
SVRNAME
------------------------------------
PATH
------------------------------------
     CH_ID
----------

NFSv3test
jfs_svmdr-nfs1
         0

NFSv3test
jfs_svmdr-nfs2

[output truncated]

SVRNAME
------------------------------------
PATH
------------------------------------
     CH_ID
----------
NFSv3test
jfs_svmdr-nfs2
         1

NFSv3test
jfs_svmdr-nfs1
         0

SVRNAME
------------------------------------
PATH
------------------------------------
     CH_ID
----------

NFSv3test
jfs_svmdr-nfs2
         1


66 rows selected.
```

These are the connections that dNFS is using. Two paths and channels are visible for each SVRNAME entry. This means multipathing is working, which means the oranfstab file was recognized and processed.

Direct NFS and host file system access

Using dNFS can occasionally cause problems for applications or user activities that rely on the visible file systems mounted on the host because the dNFS client accesses the file system out of band from the host OS. The dNFS client can create, delete, and modify files without the knowledge of the OS.

When the mount options for single-instance databases are used, they enable caching of file and directory attributes, which also means that the contents of a directory are cached. Therefore, dNFS can create a file, and there is a short lag before the OS rereads the directory contents and the file becomes visible to the user. This is not generally a problem, but, on rare occasions, utilities such as SAP BR*Tools might have issues. If this happens, address the problem by changing the mount options to use the recommendations for Oracle RAC. This change results in the disabling of all host caching.

Only change mount options when (a) dNFS is used and (b) a problem results from a lag in file visibility. If dNFS is not in use, using Oracle RAC mount options on a single-instance database results in degraded performance.

See the note about `nosharecache` in Linux NFS mount options for a Linux-specific dNFS issue that can produce unusual results.

NFS leases and locks

NFSv3 is stateless. That effectively means that the NFS server (ONTAP) doesn't keep track of which file systems are mounted, by whom, or which locks are truly in place.

ONTAP does have some features that will record mount attempts so you have an idea which clients may be accessing data, and there may be advisory locks present, but that information isn't guaranteed to be 100% complete. It can't be complete, because tracking NFS client state is not part of the NFSv3 standard.

NFSv4 statefulness

In contrast, NFSv4 is stateful. The NFSv4 server tracks which clients are using which file systems, which files exist, which files and/or regions of files are locked, etc. This means there needs to be regular communication between an NFSv4 server to keep the state data current.

The most important states being managed by the NFS server are NFSv4 Locks and NFSv4 Leases, and they are very much intertwined. You need to understand how each works by itself, and how they relate to one another.

NFSv4 locks

With NFSv3, locks are advisory. An NFS client can still modify or delete a "locked" file. An NFSv3 lock doesn't expire by itself, it must be removed. This creates problems. For example, if you have a clustered application that creates NFSv3 locks, and one of the nodes fails, what do you do? You can code the application on the surviving nodes to remove the locks, but how do you know that's safe? Maybe the "failed" node is operational, but isn't communicating with the rest of the cluster?

With NFSv4, locks have a limited duration. As long as the client holding the locks continues to check in with the NFSv4 server, no other client is permitted to acquire those locks. If a client fails to check in with the NFSv4, the locks eventually get revoked by the server and other clients will be able to request and obtain locks.

NFsv4 leases

NFSv4 locks are associated with an NFSv4 lease. When an NFSv4 client establishes a connection with an NFSv4 server, it gets a lease. If the client obtains a lock (there are many types of locks) then the lock is

associated with the lease.

This lease has a defined timeout. By default, ONTAP will set the timeout value to 30 seconds:

```
Cluster01::*> nfs server show -vserver vserver1 -fields v4-lease-seconds

vserver   v4-lease-seconds
--------- ----------------
vserver1  30
```

This means that an NFSv4 client needs to check in with the NFSv4 server every 30 seconds to renew its leases.

The lease is automatically renewed by any activity, so if the client is doing work there's no need to perform addition operations. If an application becomes quiet and is not doing real work, it's going to need to perform a sort of keep-alive operation (called a SEQUENCE) instead. It's essentially just saying "I'm still here, please refresh my leases."

```
*Question:* What happens if you lose network connectivity for 31 seconds?
```

NFSv3 is stateless. It's not expecting communication from the clients. NFSv4 is stateful, and once that lease period elapses, the lease expires, and locks are revoked and the locked files are made available to other clients.

With NFSv3, you could move network cables around, reboot network switches, make configuration changes, and be fairly sure that nothing bad would happen. Applications would normally just wait patiently for the network connection to work again.

With NFSv4, you have 30 seconds (unless you've increased the value of that parameter within ONTAP) to complete your work. If you exceed that, your leases time out. Normally this results in application crashes.

As an example, if you have an Oracle database, and you experience a loss of network connectivity (sometimes called a "network partition") that exceeds the lease timeout, you will crash the database.

Here's an example of what happens in the Oracle alert log if this happens:

```
2022-10-11T15:52:55.206231-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00202: control file: '/redo0/NTAP/ctrl/control01.ctl'
ORA-27072: File I/O error
Linux-x86_64 Error: 5: Input/output error
Additional information: 4
Additional information: 1
Additional information: 4294967295
2022-10-11T15:52:59.842508-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: '/redo1/NTAP/ctrl/control02.ctl'
ORA-27061: waiting for async I/Os failed
```

If you look at the syslogs, you should see several of these errors:

```
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
```

The log messages are usually the first sign of a problem, other than the application freeze. Typically, you see nothing at all during the network outage because processes and the OS itself are blocked attempting to access the NFS file system.

The errors appear after the network is operational again. In the example above, once connectivity was reestablished, the OS attempted to reacquire the locks, but it was too late. The lease had expired and the locks were removed. That results in an error that propagates up to the Oracle layer and causes the message in the alert log. You might see variations on these patterns depending on the version and configuration of the database.

In summary, NFSv3 tolerates network interruption, but NFSv4 is more sensitive and imposes a defined lease period.

What if a 30 second timeout isn't acceptable? What if you manage a dynamically changing network where switches are rebooted or cables are relocated and the result is the occasional network interruption? You could choose to extend the lease period, but whether you want to do that requires an explanation of NFSv4 grace periods.

NFSv4 grace periods

If an NFSv3 server is rebooted, it's ready to serve IO almost instantly. It was not maintaining any sort of state about clients. The result is that an ONTAP takeover operation often appears to be close to instantaneous. The moment a controller is ready to start serving data it will send an ARP to the network that signals the change in topology. Clients normally detect this almost instantly and data resumes flowing.

NFSv4, however, will produce a brief pause. It's just part of how NFSv4 works.

> The following sections are current as of ONTAP 9.15.1, but the lease and lock behavior as well as tuning options can change from version to version. If you need to tune NFSv4 lease/lock timeouts, please consult NetApp support for the latest information.

NFSv4 servers need to track the leases, locks, and who's using what data. If an NFS server panics and reboots, or loses power for a moment, or is restarted during maintenance activity, the result is the lease/lock and other client information is lost. The server needs to figure out which client is using what data before resuming operations. This is where the grace period comes in.

If you suddenly power cycle your NFSv4 server. When it comes back up, clients that attempt to resume IO will get a response that essentially says, "I have lost lease/lock information. Would you like to re-register your locks?" That's the start of the grace period. It defaults to 45 seconds on ONTAP:

```
Cluster01::> nfs server show -vserver vserver1 -fields v4-grace-seconds

vserver   v4-grace-seconds
--------- ----------------
vserver1  45
```

The result is that, after a restart, a controller will pause IO while all the clients reclaim their leases and locks. Once the grace period ends, the server will resume IO operations.

This grace period controls lease reclamation during network interface changes, but there is a second grace period that controls reclamation during storage failover, `locking.grace_lease_seconds`. This is a node-level option.

```
cluster01::> node run [node names or *] options
locking.grace_lease_seconds
```

For example, if you frequently needed to perform LIF failovers, and needed to reduce the grace period, you would change `v4-grace-seconds`. If you wanted to improve the IO resumption time during controller failover, you would need to change `locking.grace_lease_seconds`.

Only alter these values with caution and after fully understanding the risks and consequences. The IO pauses involved with failover and migration operations with NFSv4.X cannot be avoided entirely. Lock, lease, and grace periods are part of the NFS RFC. For many customers, NFSv3 is preferable because failover times are faster.

Lease timeouts vs grace periods

The grace period and the lease period are connected. As mentioned above, the default lease timeout is 30 seconds, which means NFSv4 clients must check in with the server at least every 30 seconds or they lose their leases and, in turn, their locks. The grace period exists to allow an NFS server to rebuild lease/lock data, and it defaults to 45 seconds. The grace period must be longer than the lease period. This ensures that an NFS client environment that is designed to renew leases at least every 30 seconds will have the ability to check in with the server after a restart. A grace period of 45 seconds ensures that all those clients that expect to renew their leases at least every 30 seconds definitely have the opportunity to do so.

If a 30 second timeout isn't acceptable, you could choose to extend the lease period.

If you want to increase the lease timeout to 60 seconds in order to withstand a 60 second network outage, you're going to have to increase the grace period as well. That means you're going to experience longer IO pauses during controller failover.

This shouldn't normally be a problem. Typical users only update ONTAP controllers once or twice per year, and unplanned failover due to hardware failures are extremely rare. In addition, if you had a network where a 60-second network outage was a concerning possibility, and you needed to the lease timeout to 60 seconds, then you probably wouldn't object to rare storage system failover resulting in a 61 second pause either. You've already acknowledged you have a network that's pausing for 60+ seconds rather frequently.

NFS caching

The presence of any of the following mount options causes host caching to be disabled:

```
cio, actimeo=0, noac, forcedirectio
```

These settings can have a severe negative effect on the speed of software installation, patching, and backup/restore operations. In some cases, especially with clustered applications, these options are required as an inevitable result of the need to deliver cache-coherency across all nodes in the cluster. In other cases, customers mistakenly use these parameters and the result is unnecessary performance damage.

Many customers temporarily remove these mount options during installation or patching of the application binaries. This removal can be performed safely if the user verifies that no other processes are actively using the target directory during the installation or patching process.

NFS transfer sizes

By default, ONTAP limits NFS I/O sizes to 64K.

Random I/O with an most applications and databases uses a much smaller block size which is well below the 64K maximum. Large-block I/O is usually parallelized, so the 64K maximum is also not a limitation to obtaining maximum bandwidth.

There are some workloads where the 64K maximum does create a limitation. In particular, single-threaded operations such as backup or recovery operation or a database full table scan run faster and more efficiently if the database can perform fewer but larger I/Os. The optimum I/O handling size for ONTAP is 256K.

The maximum transfer size for a given ONTAP SVM can be changed as follows:

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

Never decrease the maximum allowable transfer size on ONTAP below the value of rsize/wsize of currently mounted NFS file systems. This can create hangs or even data corruption with some operating systems. For example, if NFS clients are currently set at an rsize/wsize of 65536, then the ONTAP maximum transfer size could be adjusted between 65536 and 1048576 with no effect because the clients themselves are limited. Reducing the maximum transfer size below 65536 can damage availability or data.

NVFAIL

NVFAIL is a feature within ONTAP that ensures the integrity during catastrophic failover scenarios.

Databases are vulnerable to corruption during storage failover events because they maintain large internal caches. If a catastrophic event requires forcing an ONTAP failover or forcing MetroCluster switchover, irrespective of the health of the overall configuration, the result is previously acknowledged changes may be 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 protects databases and operating systems from this scenario with NVFAIL and its associated settings.

ASM Reclamation Utility (ASMRU)

ONTAP efficiently removes zeroed blocks written to a file or LUN when inline compression is enabled. Utilities such as the Oracle ASM Reclamation Utility (ASRU) work by writing zeros to unused ASM extents.

This allows DBAs to reclaim space on the storage array after data is deleted. ONTAP intercepts the zeros and deallocates the space from the LUN. The reclamation process is extremely fast because no data is being written within the storage system.

From a database perspective, the ASM diskgroup contains zeros, and reading those regions of the LUNs would result in a stream of zeros, but ONTAP does not store the zeros on drives. Instead, simple metadata changes are made that internally mark the zeroed regions of the LUN as empty of any data.

For similar reasons, performance testing involving zeroed data is not valid since blocks of zeros are not actually processed as writes within the storage array.

When using ASRU, ensure that all Oracle-recommended patches are installed.

Storage configuration on ASA r2 systems

FC SAN

LUN Alignment

LUN alignment refers to optimizing I/O with respect to the underlying file system layout.

ASA r2 systems use the same ONTAP architecture as AFF/FAS but with a simplified configuration model. ASA r2 systems use Storage Availability Zones (SAZ) instead of aggregates, but the alignment principles remain the same because ONTAP manages block layout consistently across platforms. However, note these ASA-specific points:

- ASA r2 systems provide active-active symmetric paths for all LUNs, which eliminates path asymmetry concerns during alignment.
- Storage units (LUNs) are thin-provisioned by default; alignment does not change this behavior.
- Snapshot reserve and automatic snapshot deletion can be configured during LUN creation (ONTAP 9.18.1 and later).

On a ONTAP system, storage is organized in 4KB units. A database or file system 8KB block should map to exactly two 4KB blocks. If an error in LUN configuration shifts the alignment by 1KB in either direction, each 8KB block would exist on three different 4KB storage blocks rather than two. This arrangement would cause increased latency and cause additional I/O to be performed within the storage system.

Alignment also affects LVM architectures. If a physical volume within a logical volume group is defined on the whole drive device (no partitions are created), the first 4KB block on the LUN aligns with the first 4KB block on the storage system. This is a correct alignment. Problems arise with partitions because they shift the starting location where the OS uses the LUN. As long as the offset is shifted in whole units of 4KB, the LUN is aligned.

In Linux environments, build logical volume groups on the whole drive device. When a partition is required, check alignment by running `fdisk -u` and verifying that the start of each partition is a multiple of eight. This means that the partition starts at a multiple of eight 512-byte sectors, which is 4KB.

Also see the discussion about compression block alignment in the section Efficiency. Any layout that is aligned with 8KB compression block boundaries is also aligned with 4KB boundaries.

Misalignment warnings

Database redo/transaction logging normally generates unaligned I/O that can cause misleading warnings about misaligned LUNs on ONTAP.

Logging performs a sequential write of the log file with writes of varying size. A log write operation that does not align to 4KB boundaries does not ordinarily cause performance problems because the next log write operation completes the block. The result is that ONTAP is able to process almost all writes as complete 4KB blocks, even though the data in some 4KB blocks was written in two separate operations.

Verify alignment by using utilities such as `sio` or `dd` that can generate I/O at a defined block size. The I/O alignment statistics on the storage system can be viewed with the `stats` command. See WAFL alignment verification for more information.

Alignment in Solaris environments is more complicated. Refer to ONTAP SAN Host Configuration for more information.

In Solaris x86 environments, take additional care about proper alignment because most configurations have several layers of partitions. Solaris x86 partition slices usually exist on top of a standard master boot record partition table.

Additional best practices:

- Verify HBA firmware and OS settings against the NetApp Interoperability Matrix Tool (IMT).
- Use sanlun utilities to confirm path health and alignment.
- For Oracle ASM and LVM, ensure configuration files (/etc/lvm/lvm.conf, /etc/sysconfig/oracleasm) are properly set to avoid alignment issues.

LUN sizing and LUN count

Selecting the optimal LUN size and the number of LUNs to be used is critical for optimal performance and manageability of Oracle databases.

A LUN is a virtualized object on ONTAP that exists across all of the drives in the hosting Storage Availability Zone (SAZ) on ASA r2 systems. As a result, the performance of the LUN is unaffected by its size because the LUN draws on the full performance potential of the SAZ no matter which size is chosen.

As a matter of convenience, customers might wish to use a LUN of a particular size. For example, if a database is built on an LVM or Oracle ASM diskgroup composed of two LUNs of 1TB each, then that diskgroup must be grown in increments of 1TB. It might be preferable to build the diskgroup from eight LUNs of 500GB each so that the diskgroup can be increased in smaller increments.

The practice of establishing a universal standard LUN size is discouraged because doing so can complicate manageability. For example, a standard LUN size of 100GB might work well when a database or datastore is in the range of 1TB to 2TB, but a database or datastore of 20TB in size would require 200 LUNs. This means that server reboot times are longer, there are more objects to manage in the various UIs, and products such as SnapCenter must perform discovery on many objects. Using fewer, larger LUNs avoids such problems.

ASA r2 considerations:

- Maximum LUN size for ASA r2 is 128TB, which allows for fewer, larger LUNs without performance impact.
- ASA r2 uses Storage Availability Zones (SAZ) instead of aggregates, but this does not change LUN sizing logic for Oracle workloads.
- Thin provisioning is enabled by default; resizing LUNs is non-disruptive and does not require taking them offline.

LUN count

Unlike the LUN size, the LUN count does affect performance. Application performance often depends on the ability to perform parallel I/O through the SCSI layer. As a result, two LUNs offer better performance than a single LUN. Using an LVM such as Veritas VxVM, Linux LVM2, or Oracle ASM is the simplest method to increase parallelism.

With ASA r2, the principles for LUN count remain the same as AFF/FAS because ONTAP handles parallel I/O similarly across platforms. However, ASA r2's SAN-only architecture and active-active symmetric paths ensure consistent performance across all LUNs.

NetApp customers have generally experienced minimal benefit from increasing the number of LUNs beyond sixteen, although the testing of 100%-SSD environments with very heavy random I/O has demonstrated further

improvement up to 64 LUNs.

NetApp recommends the following:

In general, four to sixteen LUNs are sufficient to support the I/O needs of any given Oracle database workload. Less than four LUNs might create performance limitations because of limitations in host SCSI implementations. Increasing beyond sixteen LUNs rarely improves performance except in extreme cases (such as very high random I/O SSD workloads).

LUN placement

Optimal placement of database LUNs within ASA r2 systems primarily depends on how various ONTAP features will be used.

In ASA r2 systems, storage units (LUNs or NVMe namespaces) are created from a simplified storage layer called Storage Availability Zones (SAZs), which act as common pools of storage for an HA pair.

There is typically only one storage availability zone (SAZ) per HA pair.

Storage Availability Zones (SAZ)

In ASA r2 systems, volumes are still there, but they are automatically created when storage units are created. Storage units (LUNs or NVMe namespaces) are provisioned directly within the automatically created volumes in Storage Availability Zones (SAZs). This design eliminates the need for manual volume management and makes provisioning more direct and streamlined for block workloads like Oracle databases.

SAZs and Storage units

Related storage units (LUNs or NVMe namespaces) are normally co-located within a single Storage Availability Zone (SAZ). For example, a database that requires 10 storage units (LUNs) would typically have all 10 units placed in the same SAZ for simplicity and performance.

- Using a 1:1 ratio of storage units to volumes, meaning one storage unit (LUN) per volume, is the ASA r2 default behavior.
- In case of more than one HA pair in the ASA r2 system, storage units (LUNs) for a given database can be distributed across multiple SAZs to optimize controller utilization and performance.

In context of FC SAN, here storage unit refers to LUN.

Consistency Groups (CGs), LUNs, and snapshots

In ASA r2, snapshot policies and schedules are applied at the Consistency Group level, which is a logical construct that groups multiple LUNs or NVMe namespaces for coordinated data protection. A dataset that consists of 10 LUNs would require only a single snapshot policy when those LUNs are part of the same Consistency Group.

Consistency Groups ensure atomic snapshot operations across all included LUNs. For example, a database that resides on 10 LUNs, or a VMware-based application environment consisting of 10 different OSs, can be protected as a single, consistent object if the underlying LUNs are grouped in the same consistency group. If they are placed in different consistency groups, snapshots may or may not be perfectly synchronized, even if scheduled at the same time.

In some cases, a related set of LUNs might need to be split into two different consistency groups because of recovery requirements. For example, a database might have four LUNs for datafiles and two LUNs for logs. In this case, a datafile consistency group with 4 LUNs and a log consistency group with 2 LUNs might be the best option. The reason is independent recoverability: the datafile consistency group could be selectively restored to an earlier state, meaning all four LUNs would be reverted to the state of the snapshot, while the log consistency group with its critical data would remain unaffected.

CGs, LUNs, and SnapMirror

SnapMirror policies and operations are, like snapshot operations, performed on the consistency group, not the LUN.

Co-locating related LUNs in a single consistency group allows you to create a single SnapMirror relationship and update all contained data with a single update. As with snapshots, the update will also be an atomic operation. The SnapMirror destination would be guaranteed to have a single point-in-time replica of the source LUNs. If the LUNs were spread across multiple consistency groups, the replicas may or may not be consistent with one another.

SnapMirror replication on ASA r2 systems has the following limitations:

- SnapMirror synchronous replication is not supported.
- SnapMirror active sync is supported only between two ASA r2 systems.
- SnapMirror asynchronous replication is supported only between two ASA r2 systems.
- SnapMirror asynchronous replication is not supported between an ASA r2 system and an ASA, AFF or FAS system or the cloud.

Learn more about SnapMirror replication policies supported on ASA r2 systems.

CGs, LUNs, and QoS

While QoS can be selectively applied to individual LUNs, it is usually easier to set it at the consistency group level. For example, all of the LUNs used by the guests in a given ESX server could be placed in a single consistency group, and then an ONTAP adaptive QoS policy could be applied. The result is a self-scaling IOPS-per-TiB limit that applies to all LUNs.

Likewise, if a database required 100K IOPS and occupied 10 LUNs, it would be easier to set a single 100K IOPS limit on a single consistency group than to set 10 individual 10K IOPS limits, one on each LUN.

Multiple CG layouts

There are some cases where distributing LUNs across multiple consistency groups may be beneficial. The primary reason is controller striping. For example, an HA ASA r2 storage system might be hosting a single Oracle database where the full processing and caching potential of each controller is required. In this case, a typical design would be to place half of the LUNs in a single consistency group on controller 1, and the other half of the LUNs in a single consistency group on controller 2.

Similarly, for environments hosting many databases, distributing LUNs across multiple consistency groups can ensure balanced controller utilization. For example, an HA system hosting 100 databases of 10 LUNs each might assign 5 LUNs to a consistency group on controller 1 and 5 LUNs to a consistency group on controller 2 per database. This guarantees symmetric loading as additional databases are provisioned.

None of these examples involve a 1:1 LUN-to-consistency group ratio, though. The goal remains to optimize manageability by grouping related LUNs logically in consistency group.

One example where a 1:1 LUN to consistency group ratio makes sense is containerized workloads, where each LUN might really represent a single workload requiring separate snapshot and replication policies and thus need to be managed on an individual basis. In such cases, a 1:1 ratio may be optimal.

LUN resizing and LVM resizing

When a SAN-based file system or Oracle ASM disk group reaches its capacity limit on ASA r2, there are two options for increasing available space:

- Increase the size of the existing LUNs (storage units)
- Add a new LUN to an existing ASM disk group or LVM volume group and grow the contained logical volume

Although LUN resizing is supported on ASA r2, it is generally better to use a Logical Volume Manager (LVM) such as Oracle ASM. One of the principal reasons LVMs exist is to avoid the need for frequent LUN resizing. With an LVM, multiple LUNs are combined into a virtual pool of storage. Logical volumes carved from this pool can be easily resized without impacting the underlying storage configuration.

Additional benefits of using LVM or ASM include:

- Performance optimization: Distributes I/O across multiple LUNs, reducing hotspots.
- Flexibility: Add new LUNs without disrupting existing workloads.
- Transparent migration: ASM or LVM can relocate extents to new LUNs for balancing or tiering without host downtime.

Key ASA r2 considerations:

- LUN resizing is performed at the storage unit level within a Storage VM (SVM) using capacity from the Storage Availability Zone (SAZ).
- For Oracle, best practice is to add LUNs to ASM disk groups rather than resizing existing LUNs, to maintain striping and parallelism.

LVM striping

LVM striping refers to distributing data across multiple LUNs. The result is dramatically improved performance for many databases.

Before the era of flash drives, striping was used to help overcome the performance limitations of spinning drives. For example, if an OS needs to perform a 1MB read operation, reading that 1MB of data from a single drive would require a lot of drive head seeking and reading as the 1MB is slowly transferred. If that 1MB of data was striped across 8 LUNs, the OS could issue eight 128K read operations in parallel and reduce the time required to complete the 1MB transfer.

Striping with spinning drives was more difficult because the I/O pattern had to be known in advance. If the striping wasn't correctly tuned for the true I/O patterns, striped configurations could damage performance. With Oracle databases, and especially with all-flash storage configurations, striping is much easier to configure and has been proven to dramatically improve performance.

Logical volume managers such as Oracle ASM stripe by default, but native OS LVM do not. Some of them bond multiple LUNs together as a concatenated device, which results in datafiles that exist on one and only one LUN device. This causes hot spots. Other LVM implementations default to distributed extents. This is similar to striping, but it's coarser. The LUNs in the volume group are sliced into large pieces, called extents

and typically measured in many megabytes, and the logical volumes are then distributed across those extents. The result is random I/O against a file should be well distributed across LUNs, but sequential I/O operations are not as efficient as they could be.

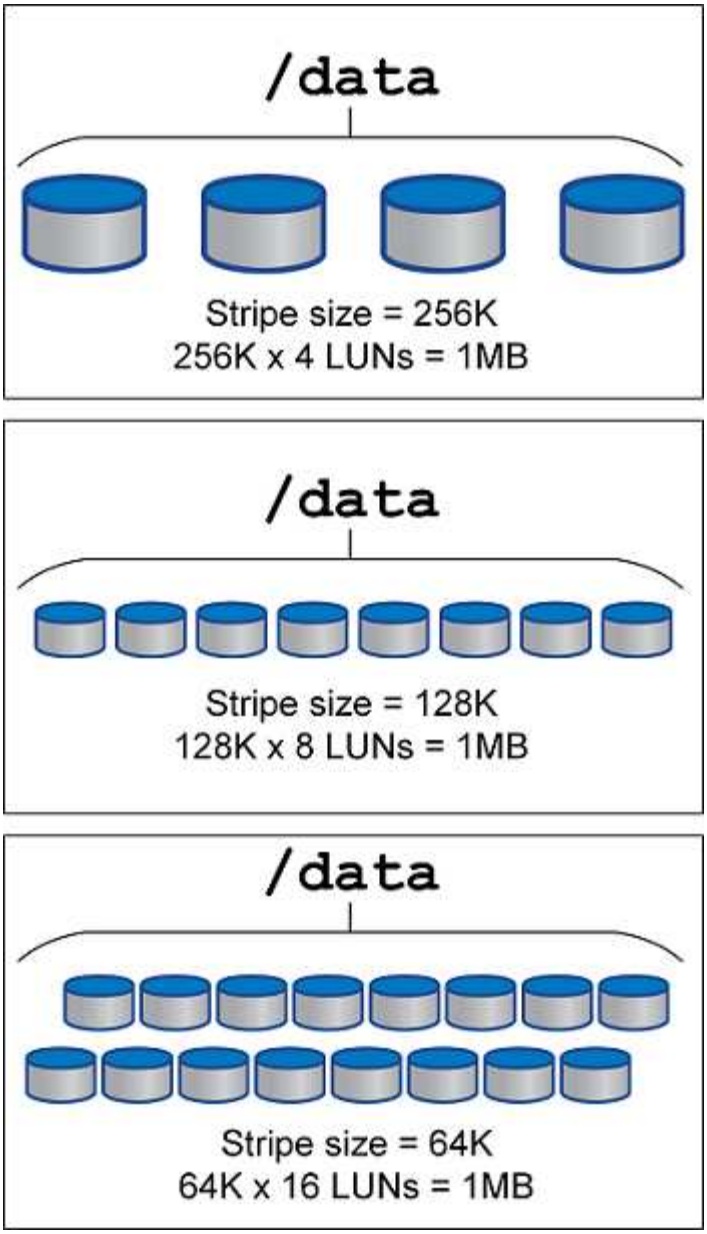
Performance-intensive application I/O is nearly always either (a) in units of the basic block size or (b) one megabyte.

The primary goal of a striped configuration is to ensure that single-file I/O can be performed as a single unit, and multiblock I/Os, which should be 1MB in size, can be parallelized evenly across all LUNs in the striped volume. This means that the stripe size must not be smaller than the database block size, and the stripe size multiplied by the number of LUNs should be 1MB.

Best practice for LVM striping with Oracle database:

- Stripe size ≥ database block size.
- Stripe size * number of LUNs ≈ 1MB for optimal parallelism.
- Use multiple LUNs per ASM disk group to maximize throughput and avoid hotspots.

The following figure shows three possible options for stripe size and width tuning. The number of LUNs is selected to meet performance requirements as described above, but in all cases the total data within a single stripe is 1MB.



NVFAIL

NVFAIL is an ONTAP feature that ensures data integrity during catastrophic failover scenarios.

This functionality is still applicable on ASA r2 systems, even though ASA r2 uses a simplified SAN architecture (SAZs and storage units instead of volumes).

Databases are vulnerable to corruption during storage failover events because they maintain large internal caches. If a catastrophic event requires forcing an ONTAP failover, irrespective of the health of the overall configuration, the result is previously acknowledged changes may be 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 failover 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. 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 in these situations could result in file system corruption.

ONTAP protects databases and operating systems against this scenario using NVFAIL and its associated settings, which signal the host to invalidate cached data and remount the affected file systems after failover. This mechanism applies to ASA r2 LUNs and namespaces just as it does on AFF/FAS.

Key ASA r2 considerations:

- NVFAIL operates at the LUN level (storage unit), not at the SAZ level.
- For Oracle databases, NVFAIL should be enabled on all LUNs hosting critical components (datafiles, redo logs, control files).
- MetroCluster is not supported on ASA r2, so NVFAIL primarily applies to local HA failover scenarios.
- NFS is not supported on ASA r2, so NVFAIL considerations apply only to SAN-based workloads (FC/iSCSI/NVMe).

ASM Reclamation Utility (ASRU)

ONTAP on ASA r2 efficiently removes zeroed blocks written to a LUN (storage unit) when inline compression is enabled. Utilities such as the Oracle ASM Reclamation Utility (ASRU) work by writing zeros to unused ASM extents.

This allows DBAs to reclaim space on the storage array after data is deleted. ONTAP intercepts the zeros and deallocates the space from the LUN. The reclamation process is extremely fast because no actual data is being written within the storage system.

From a database perspective, the ASM diskgroup contains zeros, and reading those regions of the LUNs would result in a stream of zeros, but ONTAP does not store the zeros on drives. Instead, simple metadata changes are made that internally mark the zeroed regions of the LUN as empty of any data.

For similar reasons, performance testing involving zeroed data is not valid since blocks of zeros are not actually processed as writes within the storage array.

Key ASRU considerations with ASA r2 ONTAP:

- Works the same way as AFF/FAS for SAN workloads because ASA r2 is block-only.
- Applies to LUNs and NVMe namespaces provisioned within SAZs.
- No FlexVol volumes exist, but the zero-block reclamation behavior is identical.

When using ASRU, ensure that all Oracle-recommended patches are installed.

Virtualization

Virtualization of databases with VMware, Oracle OLVM, or KVM is an increasingly common choice for NetApp customers who chose virtualization for even their most

mission-critical databases.

Supportability

Many misconceptions exist about the Oracle support policies for virtualization, particularly for VMware products. It is not uncommon to hear that Oracle outright does not support virtualization. This notion is incorrect and leads to missed opportunities to benefit from virtualization. Oracle Doc ID 249212.1 discusses the actual requirements, and is rarely considered by customers to be a concern.

If a problem occurs on a virtualized server and that problem is previously unknown to Oracle Support, the customer might be asked to reproduce the problem on physical hardware. An Oracle customer running a bleeding-edge version of a product might not want to use virtualization because of the potential for supportability problems, but this situation has not been a real-world for virtualization customers using generally available Oracle product versions.

Storage presentation

Customers considering virtualization of their databases should base their storage decisions on their business needs. Although this is a generally true statement for all IT decisions, it is especially important for database projects, because the size and scope of requirements vary considerably.

There are three basic options for storage presentation:

- Virtualized LUNs on hypervisor datastores
- iSCSI LUNs managed by the iSCSI initiator on the VM, not the hypervisor
- NFS file systems mounted by the VM (not from an NFS-based datastore)
- Direct device mappings. VMware RDMs are disfavored by customers, but physical devices are still often similarly directly mapped with KVM and OLVM virtualization.

Performance

The method of presenting storage to a virtualized guest does not generally affect performance. Host OSs, virtualized network drivers, and hypervisor datastore implementations are all highly optimized and can generally consume all available FC or IP network bandwidth between the hypervisor and the storage system as long as basic best practices are followed. In some cases, obtaining optimal performance might be slightly easier using one storage presentation approach as compared to another, but the end result should be comparable.

Manageability

The key factor in deciding how to present storage to a virtualized guest is mangeability. There is no right or wrong method. The best approach depends on IT operational needs, skills, and preferences.

Factors to consider include:

- **Transparency.** When a VM manages its file systems, it is easier for a database administrator or a system administrator to identify the source of the file systems for their data. The filesystems and LUNs are accessed no differently than with a physical server.
- **Consistency.** When a VM owns its file systems, the use or nonuse of a hypervisor layer affects manageability. The same procedures for provisioning, monitoring, data protection, and so on can be used across the entire estate, including both virtualized and nonvirtualized environments.

On the other hand, in a otherwise 100% virtualized data center it may be preferable to also use datastore-based storage across the entire footprint on the same rationale mentioned above - consistency - the ability to use the same procedures for provisioning, protection, montoring, and data protection.

- **Stability and troubleshooting.** When a VM owns its file systems, delivering good, stable performance and troubleshooting problems are simpler because the entire storage stack is present on the VM. The hypervisor's only role is to transport FC or IP frames. When a datastore is included in a configuration, it complicates the configuration by introducing another set of timeouts, parameters, log files, and potential bugs.
- **Portability.** When a VM owns its file systems, the process of moving an Oracle environment becomes much simpler. File systems can easily be moved between virtualized and nonvirtualized guests.
- **Vendor lock-in.** After data is placed in a datastore, using a different hypervisor or taking the data out of the virtualized environment entirely becomes difficult.
- **Snapshot enablement.** Traditional backup procedures in a virtualized environment can become a problem because of the relatively limited bandwidth. For example, a four-port 10GbE trunk might be sufficient to support the day-to-day performance needs of many virtualized databases, but such a trunk would be insufficient to perform backups using RMAN or other backup products that require streaming a full-sized copy of the data. The result is that an increasingly consolidated virtualized environment needs to perform backups via storage snapshots. This avoids the need to overbuild the hypervisor configuration purely to support the bandwidth and CPU requirements in the backup window.

  Using guest-owned file systems sometimes makes it easier to leverage snapshot-based backups and restores because the storage objects in need of protection can be targeted more easily. However, there are an increasingly large number of virtualization data protection products that integrate well with datastores and snapshots. The backup strategy should be fully considred before making a decision on how to present storage to a virtualized host.

Paravirtualized drivers

For optimum performance, the use of paravirtualized network drivers is critical. When a datastore is used, a paravirtualized SCSI driver is required. A paravirtualized device driver allows a guest to integrate more deeply into the hypervisor, as opposed to an emulated driver in which the hypervisor spends more CPU time mimicking the behavior of physical hardware.

Overcommitting RAM

Overcommitting RAM means configuring more virtualized RAM on various hosts than exists on the physical hardware. Doing so can cause unexpected performance problems. When virtualizing a database, the underlying blocks of the Oracle SGA must not be swapped out to storage by the hypervisor. Doing so causes highly unstable performance results.

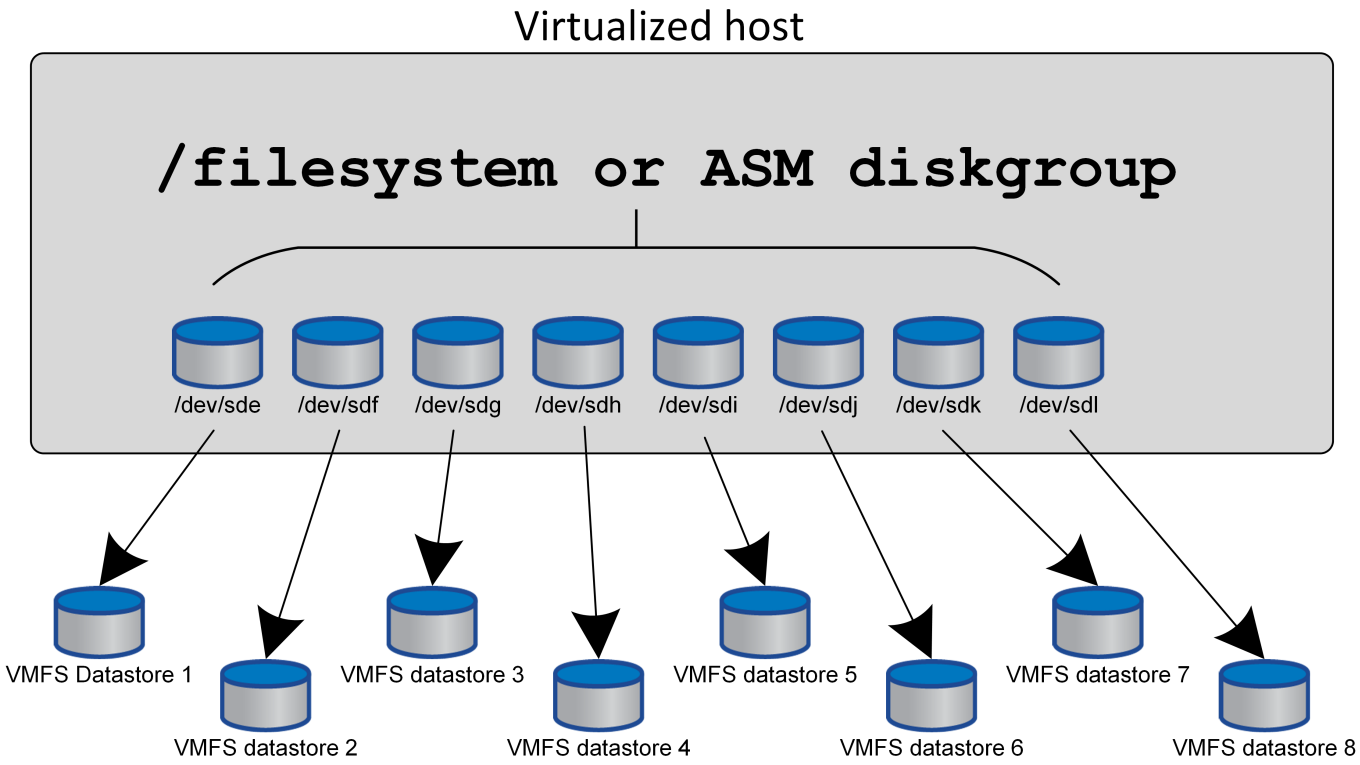
Datastore striping

When using databases with datastores, there is one critical factor to consider with respect to performance - striping.

Datastore technologies such as VMFS are able to span multiple LUNs, but they are not striped devices. The LUNs are concatenated. The end result can be LUN hot spots. For example, a typical Oracle database might have an 8-LUN ASM diskgroup. All 8 virtualized LUNs could be provisioned on an 8-LUN VMFS datastore, but there is no guarantee on which LUNs the data will reside. The resulting configuration could be all 8 virtualized LUN occupying a single LUN within the VMFS datastore. This becomes a performance bottleneck.

Striping is usually required. With some hypervisors, including KVM, it is possible to build a datastore using LVM striping as described here. With VMware, the architecture looks a little different. Each virtualized LUN needs to be placed on a different VMFS datastore.

For example:



The primary driver for this approach is not ONTAP, it's because of inherent limitation of the number of operations a single VM or hypervisor LUN can service in parallel. A single ONTAP LUN can generally support far more IOPS than a host can request. The single-LUN performance limit is almost universally a result of the host OS. The result is that most databases need between 4 and 8 LUNs to meet their performance needs.

VMware architectures need to plan their architectures carefully to ensure that datastore and/or LUN path maximumes are not encountered with this approach. Addtionally, there is no requirement for a unique set of VMFS datastores for every database. The primary need is to ensure each host has a clean set of 4-8 IO paths from the virtualized LUNs to the backend LUNs on the storage system itself. In rare occasions, even more datatores may be beneficial for truly extreme performance demands, but 4-8 LUNs is generally sufficient for 95% of all databases. A single ONTAP volume containing 8 LUNs can support up to 250,000 random Oracle block IOPS with a typical OS/ONTAP/network configuration.

Tiering

Overview

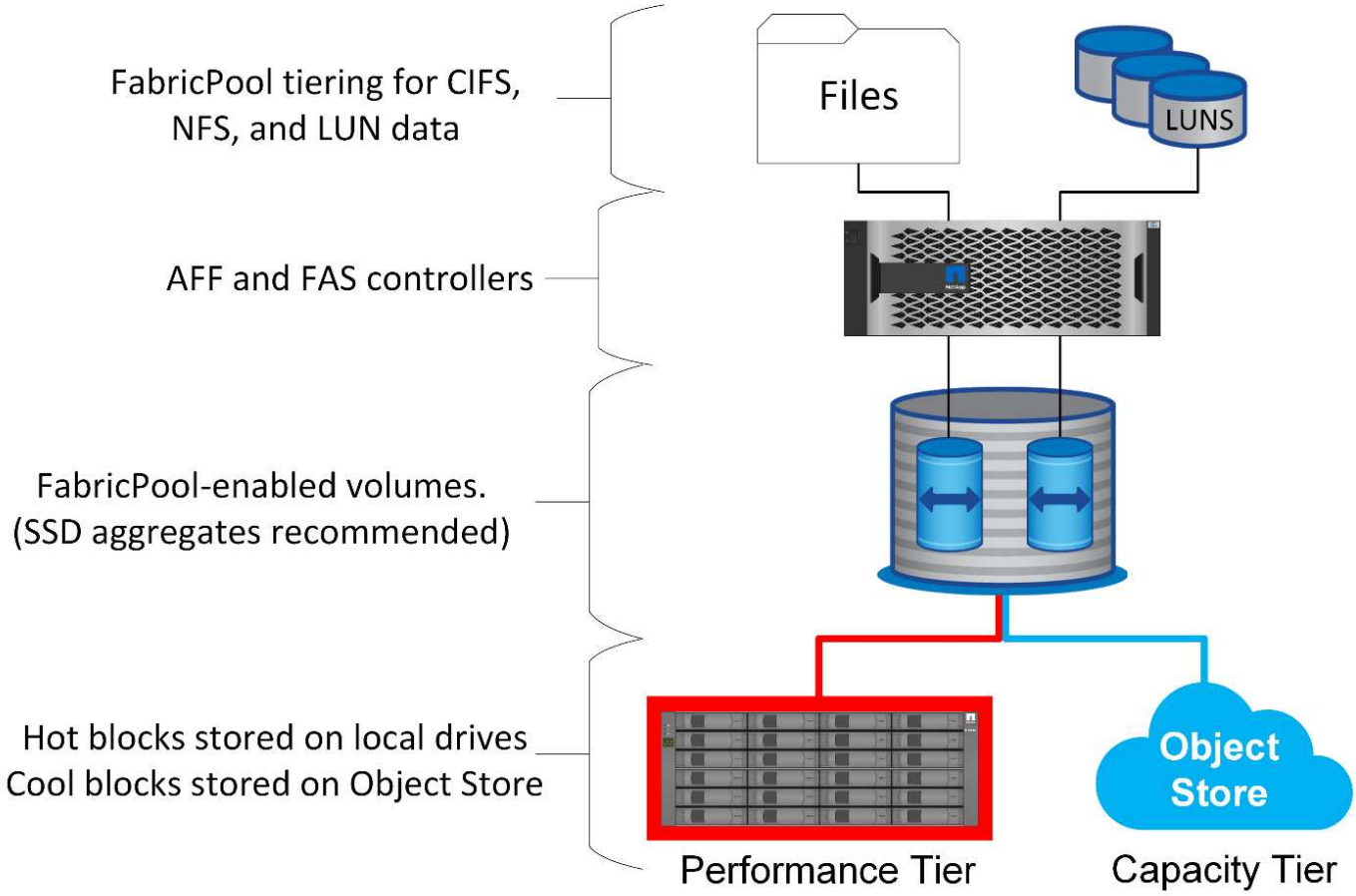
Understanding how FabricPool tiering affects Oracle and other databases requires an understanding of low-level FabricPool architecture.

Architecture

FabricPool is a tiering technology that classifies blocks as hot or cool and places them in the most appropriate

tier of storage. The performance tier is most often located on SSD storage and hosts the hot data blocks. The capacity tier is located on an object store and hosts the cool data blocks. Object storage support includes NetApp StorageGRID, ONTAP S3, Microsoft Azure Blob storage, Alibaba Cloud Object Storage service, IBM Cloud Object Storage, Google Cloud storage, and Amazon AWS S3.

Multiple tiering policies are available that control how blocks are classified as hot or cool, and policies can be set on a per-volume basis and changed as required. Only the data blocks are moved between the performance and capacity tiers. The metadata that defines the LUN and file system structure always remains on the performance tier. As a result, management is centralized on ONTAP. Files and LUNs appear no different from data stored on any other ONTAP configuration. The NetApp AFF or FAS controller applies the defined policies to move data to the appropriate tier.



Object store providers

Object storage protocols use simple HTTP or HTTPS requests for storing large numbers of data objects. Access to the object storage must be reliable, because data access from ONTAP depends on prompt servicing of requests. Options include the Amazon S3 Standard and Infrequent Access options, and Microsoft Azure Hot and Cool Blob Storage, IBM Cloud, and Google Cloud. Archival options such as Amazon Glacier and Amazon Archive are not supported because the time required to retrieve data can exceed the tolerances of host operating systems and applications.

NetApp StorageGRID is also supported and is an optimal enterprise-class solution. It is a high-performance, scalable, and highly secure object storage system that can provide geographic redundancy for FabricPool data as well as other object store applications that are increasingly likely to be part of enterprise application environments.

StorageGRID can also reduce costs by avoiding the egress charges imposed by many public cloud providers

for reading data back from their services.

Data and metadata

Note that the term "data" here applies to the actual data blocks, not the metadata. Only data blocks are tiered, while metadata remains in the performance tier. In addition, the status of a block as hot or cool is only affected by reading the actual data block. Simply reading the name, timestamp, or ownership metadata of a file does not affect the location of the underlying data blocks.

Backups

Although FabricPool can significantly reduce storage footprints, it is not by itself a backup solution. NetApp WAFL metadata always stays on the performance tier. If a catastrophic disaster destroys the performance tier, a new environment cannot be created using the data on the capacity tier because it contains no WAFL metadata.

FabricPool can, however, become part of a backup strategy. For example, FabricPool can be configured with NetApp SnapMirror replication technology. Each half of the mirror can have its own connection to an object storage target. The result is two independent copies of the data. The primary copy consists of the blocks on the performance tier and associated blocks in the capacity tier, and the replica is a second set of performance and capacity blocks.

Tiering policies

Tiering policies

Four policies are available in ONTAP which control how Oracle data on the performance tier become a candidate to be relocated to the capacity tier.

Snapshot-only

The `snapshot-only` `tiering-policy` applies only to blocks that are not shared with the active file system. It essentially results in tiering of database backups. Blocks become candidates for tiering after a snapshot is created and the block is then overwritten, resulting in a block that exists only within the snapshot. The delay before a `snapshot-only` block is considered cool is controlled by the `tiering-minimum-cooling-days` setting for the volume. The range as of ONTAP 9.8 is from 2 to 183 days.

Many datasets have low change rates, resulting in minimal savings from this policy. For example, a typical database observed on ONTAP has a change rate of less than 5% per week. Database archive logs can occupy extensive space, but they usually continue to exist in the active file system and thus would not be candidates for tiering under this policy.

Auto

The `auto` tiering policy extends tiering to both snapshot-specific blocks as well as blocks within the active file system. The delay before a block is considered cool is controlled by the `tiering-minimum-cooling-days` setting for the volume. The range as of ONTAP 9.8 is from 2 to 183 days.

This approach enables tiering options that are not available with the `snapshot-only` policy. For example, a data protection policy might require 90 days of certain log files to be retained. Setting a cooling period of 3 days results in any log files older than 3 days to be tiered out from the performance layer. This action frees up substantial space on the performance tier while still allowing you to view and manage the full 90 days of data..

None

The `none` tiering policy prevents any additional blocks from being tiered from the storage layer, but any data still in the capacity tier remains in the capacity tier until it is read. If the block is then read, it is pulled back and placed on the performance tier.

The primary reason to use the `none` tiering policy is to prevent blocks from being tiered, but it could become useful to change the policies over time. For example, let's say that a specific dataset is extensively tiered to the capacity layer, but an unexpected need for full performance capabilities arises. The policy can be changed to prevent any additional tiering and to confirm that any blocks read back as IO increases remain in the performance tier.

All

The `all` tiering policy replaces the `backup` policy as of ONTAP 9.6. The `backup` policy applied only to data protection volumes, meaning a SnapMirror or NetApp SnapVault destination. The `all` policy functions the same, but is not restricted to data protection volumes.

With this policy, blocks are immediately considered cool and eligible to be tiered to the capacity layer immediately.

This policy is especially appropriate for long-term backups. It can also be used as a form of Hierarchical Storage Management (HSM). In the past, HSM was commonly used to tier the data blocks of a file to tape while keeping the file itself visible on the file system. A FabricPool volume with the `all` policy allows you to store files in a visible and manageable yet consuming nearly no space on the local storage tier.

Retrieval policies

The tiering policies control which Oracle database blocks are tiered from the performance tier to the capacity tier. Retrieval policies control what happens when a block that has been tiered is read.

Default

All FabricPool volumes are initially set at `default`, which means the behavior is controlled by the `cloud-retrieval-policy. `The exact behavior depends on the tiering policy used.

- `auto`- only retrieve randomly read data
- `snapshot-only`- retrieve all sequentially or randomly read data
- `none`- retrieve all sequentially or randomly read data
- `all`- do not retrieve data from the capacity tier

On-read

Setting `cloud-retrieval-policy` to on-read overrides the default behavior so a read of any tiered data results in that data being returned to the performance tier.

For example, a volume might have been lightly used for a long time under the `auto` tiering policy and most of the blocks are now tiered out.

If an unexpected change in business needs required some of the data to be repeatedly scanned in order to prepare a certain report, it may be desirable to change the `cloud-retrieval-policy` to `on-read` to

ensure that all data that is read is returned to the performance tier, including both sequentially and randomly read data. This would improve performance of sequential I/O against the volume.

Promote

The behavior of the promote policy depends on the tiering policy. If the tiering policy is `auto`, then setting the `cloud-retrieval-policy `to `promote` brings back all blocks from the capacity tier on the next tiering scan.

If the tiering policy is `snapshot-only`, then the only blocks that are returned are the blocks that are associated with the active file system. Normally this would not have any effect because the only blocks tiered under the `snapshot-only` policy would be blocks associated exclusively with snapshots. There would be no tiered blocks in the active file system.

If, however, data on a volume was restored by a volume SnapRestore or file-clone operation from a snapshot, some of the blocks that were tiered out because they were only associated with snapshots may now be required by the active file system. It may be desirable to temporarily change the `cloud-retrieval-policy` policy to `promote` to quickly retrieve all locally required blocks.

Never

Do not retrieve blocks from the capacity tier.

Tiering strategies

Full file tiering

Although FabricPool tiering operates at the block level, in some cases it can be used to provide file-level tiering.

Many applications datasets are organized by date, and such data is generally increasingly less likely to be accessed as it ages. For example, a bank might have a repository of PDF files that contain five years of customer statements, but only the most recent few months are active. FabricPool can be used to relocate older datafiles to the capacity tier. A cooling period of 14 days would ensure the more recent 14 days of PDF files remain on the performance tier. Furthermore, files that are read at least every 14 days would remain hot and therefore remain on the performance tier.

Policies

To implement a file-based tiering approach, you must have files that are written and not subsequently modified. The `tiering-minimum-cooling-days` policy should be set high enough so that files that you might need remain on the performance tier. For example, a dataset for which the most recent 60 days of data is required with optimal performance warrants setting the `tiering-minimum-cooling-days` period to 60. Similar results can also be achieved based on the file access patterns. For example, if the most recent 90 days of data is required and the application is accessing that 90-day span of data, then the data would remain on the performance tier. By setting the `tiering-minimum-cooling-days` period to 2, you get prompt tiering after the data becomes less active.

The `auto` policy is required to drive tiering of these blocks because only the `auto` policy affects blocks that are in the active file system.

Any type of access to data resets the heat map data. Virus scanning, indexing, and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Partial file tiering

Because FabricPool works at the block level, files that are subject to change can be partially tiered to object storage while also remaining partially on performance tier.

This is common with databases. Databases that are known to contain inactive blocks are also candidates for FabricPool tiering. For example, a supply chain management database might contain historical information that must be available if needed but is not accessed during normal operations. FabricPool can be used to selectively relocate the inactive blocks.

For example, datafiles running on a FabricPool volume with a `tiering-minimum-cooling-days` period of 90 days retains any blocks accessed in the preceding 90 days on the performance tier. However, anything that is not accessed for 90 days is relocated to the capacity tier. In other cases, normal application activity preserves the correct blocks on the correct tier. For example, if a database is normally used to process the previous 60 days of data on a regular basis, a much lower `tiering-minimum-cooling-days` period can be set because the natural activity of the application makes sure that blocks are not relocated prematurely.

The `auto` policy should be used with care with databases. Many databases have periodic activities such as end-of-quarter process or reindexing operations. If the period of these operations is greater than the `tiering-minimum-cooling-days` performance problems can occur. For example, if end-of-quarter processing requires 1TB of data that was otherwise untouched, that data might now be present on the capacity tier. Reads from the capacity tier is often extremely fast and may not cause performance problems, but the exact results will depend on the object store configuration.

Policies

The `tiering-minimum-cooling-days` policy should be set high enough to retain files that might be required on the performance tier. For example, a database in which the most recent 60 days of data might be required with optimal performance would warrant setting the `tiering-minimum-cooling-days` period to 60 days. Similar results could also be achieved based on the access patterns of files. For example, if the most recent 90 days of data is required and the application is accessing that 90- day span of data, then the data would remain on the performance tier. Setting the `tiering-minimum-cooling-days` period to 2 days would tier the data promptly after the data becomes less active.

The `auto` policy is required to drive tiering of these blocks because only the `auto` policy affects blocks that are in the active file system.

Any type of access to data resets the heat map data. Therefore, database full table scans and even backup activity that reads the source files prevents tiering because the required `tiering-minimum-cooling-days` threshold is never reached.

Archive log tiering

Perhaps the most important use for FabricPool is improving the efficiency of known cold data, such as database transaction logs.

Most relational databases operate in transaction log archival mode to deliver point-in-time recovery. Changes to the databases are committed by recording the changes in the transaction logs, and the transaction log is retained without being overwritten. The result can be a requirement to retain an enormous volume of archived transaction logs. Similar examples exist with many other application workflows that generate data that must be retained, but is highly unlikely to ever be accessed.

FabricPool solves these problems by delivering a single solution with integrated tiering. Files are stored and remain accessible in their usual location, but take up virtually no space on the primary array.

Policies

Use a `tiering-minimum-cooling-days` policy of a few days results in retention of blocks in the recently created files (which are the files most likely to be required in the near term) on the performance tier. The data blocks from older files are then moved to the capacity tier.

The `auto` enforces prompt tiering when the cooling threshold has been reached irrespective of whether the logs have been deleted or continue to exist in the primary file system. Storing all the potentially required logs in a single location in the active file system also simplifies management. There is no reason to search through snapshots to locate a file that needs to be restored.

Some applications, such as Microsoft SQL Server, truncate transaction log files during backup operations so that the logs are no longer in the active file system. Capacity might be saved by using the `snapshot-only` tiering policy, but the `auto` policy is not useful for log data because there should rarely cooled log data in the active file system.

Snapshot tiering

The initial release of FabricPool targeted the backup use case. The only type of blocks that could be tiered were blocks that were no longer associated with data in the active file system. Therefore, only the snapshot data blocks could be moved to the capacity tier. This remains one of the safest tiering options when you need to ensure performance is never affected.

Policies - local snapshots

Two options exist for tiering inactive snapshot blocks to the capacity tier. First, the `snapshot-only` policy only targets the snapshot blocks. Although the `auto` policy includes the `snapshot-only` blocks, it also tiers blocks from the active file system. This might not be desirable.

The `tiering-minimum-cooling-days` value should be set to a time period that makes data that might be required during a restoration available on the performance tier. For example, most restore scenarios of a critical production database include a restore point at some time in the previous few days. Setting a `tiering-minimum-cooling-days` value of 3 would make sure that any restoration of the file results in a file that immediately delivers maximum performance. All blocks in the active files are still present on fast storage without needing to recover them from the capacity tier.

Policies - replicated snapshots

A snapshot that is replicated with SnapMirror or SnapVault that is only used for recovery should generally use the FabricPool `all` policy. With this policy, metadata is replicated, but all data blocks are immediately sent to the capacity tier, which yields maximum performance. Most recovery processes involve sequential I/O, which is inherently efficient. The recovery time from the object store destination should be evaluated, but, in a well-designed architecture, this recovery process does not need to be significantly slower than recovery from local

data.

If the replicated data is also intended to be used for cloning, the `auto` policy is more appropriate, with a `tiering-minimum-cooling-days` value that encompasses data that is expected to be regularly used in a cloning environment. For example, a database's active working set might include data read or written in the previous three days, but it could also include another 6 months of historical data. If so, then the `auto` policy at the SnapMirror destination makes the working set available on the performance tier.

Backup tiering

Traditional application backups include products such as Oracle Recovery Manager, which create file-based backups outside the location of the original database.

```
`tiering-minimum-cooling-days` policy of a few days preserves the most
recent backups, and therefore the backups most likely to be required for
an urgent recovery situation, on the performance tier. The data blocks of
the older files are then moved to the capacity tier.
```

The `auto` policy is the most appropriate policy for backup data. This ensures prompt tiering when the cooling threshold has been reached irrespective of whether the files have been deleted or continue to exist in the primary file system. Storing all the potentially required files in a single location in the active file system also simplifies management. There is no reason to search through snapshots to locate a file that needs to be restored.

The `snapshot-only` policy could be made to work, but that policy only applies to blocks that are no longer in the active file system. Therefore, files on an NFS or SMB share must be deleted first before the data can be tiered.

This policy would be even less efficient with a LUN configuration because deletion of a file from a LUN only removes the file references from the file system metadata. The actual blocks on the LUNs remain in place until they are overwritten. This situation can create a lengthy delay between the time a file is deleted and the time that the blocks are overwritten and become candidates for tiering. There is some benefit to moving the `snapshot-only` blocks to the capacity tier, but, overall, FabricPool management of backup data works best with the `auto` policy.

This approach helps users manage the space required for backups more efficiently, but FabricPool itself is not a backup technology. Tiering backup files to object store simplifies management because the files are still visible on the original storage system, but the data blocks in the object store destination depend on the original storage system. If the source volume is lost, the object store data is no longer useable.

Object store access interruptions

Tiering a dataset with FabricPool results in a dependency between the primary storage array and the object store tier. There are many object storage options that offer varying levels of availability. It is important to understand the impact of a possible loss of connectivity between the primary storage array and the object storage tier.

If an I/O issued to ONTAP requires data from the capacity tier and ONTAP cannot reach the capacity tier to retrieve blocks, then the I/O eventually times out. The effect of this timeout depends on the protocol used. In an

NFS environment, ONTAP responds with either an EJUKEBOX or EDELAY response, depending on the protocol. Some older operating systems might interpret this as an error, but current operating systems and current patch levels of the Oracle Direct NFS client treat this as a retriable error and continue waiting for the I/O to complete.

A shorter timeout applies to SAN environments. If a block in the object store environment is required and remains unreachable for two minutes, a read error is returned to the host. The ONTAP volume and LUNs remain online, but the host OS might flag the file system as being in an error state.

Object storage connectivity problems `snapshot-only` policy is less of a concern because only backup data is tiered. Communication problems would slow data recovery but would not otherwise affect data being actively used. The `auto` and `all` policies allow tiering of cold data from the active LUN, which means that an error during object store data retrieval could affect database availability. A SAN deployment with these policies should only be used with enterprise-class object storage and network connections designed for high availability. NetApp StorageGRID is the superior option.

Oracle data protection

Data protection with ONTAP

NetApp knows the most mission-critical data is found in databases.

An enterprise cannot operate without access to its data, and sometimes, the data defines the business. This data must be protected; however, data protection is more than just ensuring a usable backup—it is about performing the backups quickly and reliably in addition to storing them safely.

The other side of data protection is data recovery. When data is inaccessible, the enterprise is affected and might be inoperative until data is restored. This process must be fast and reliable. Finally, most databases must be protected against disasters, which means maintaining a replica of the database. The replica must be sufficiently up to date. It must also be quick and simple to make the replica a fully operational database.

This documentation replaces previously published technical report *TR-4591: Oracle data protection: Backup, recovery, and replication.*

Planning

The right enterprise data protection architecture depends on the business requirements surrounding data retention, recoverability, and tolerance for disruption during various events.

For example, consider the number of applications, databases, and important datasets in scope. Building a backup strategy for a single dataset that ensures compliance with typical SLAs is fairly straightforward because there are not many objects to manage. As the number of datasets increases, monitoring becomes more complicated and administrators might be forced to spend an increasing amount of time addressing backup failures. As an environment reaches cloud and service provider scales, a wholly different approach is needed.

Dataset size also affects strategy. For example, many options exist for backup and recovery with a 100GB database because the data set is so small. Simply copying the data from backup media with traditional tools typically delivers a sufficient RTO for recovery. A 100TB database normally needs a completely different strategy unless the RTO allows for a multiday outage, in which case a traditional copy-based backup and recovery procedure might be acceptable.

Finally, there are factors outside the backup and recovery process itself. For example, are there databases supporting critical production activities, making recovery a rare event that is only performed by skilled DBAs?

Alternatively, are the databases part of a large development environment in which recovery is a frequent occurrence and managed by a generalist IT team?

RTO, RPO, and SLA planning

ONTAP allows you to easily tailor an Oracle database data protection strategy to your business requirements.

These requirements include factors such as 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.

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, although it would be very inconvenient to lose access to this data during the workday, the business can still operate. 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.

Recovery point objective

The recovery point objective (RPO) defines the maximum tolerable data loss. In many cases, the RPO is solely determined by the frequency of snapshots or snapmirror updates.

In some cases, the RPO can be made more aggressive be selectively protecting certain data more frequently. 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 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.

Disaster recovery

Disaster recovery includes the IT architecture, policies, and procedures required to recover a service in the event of a physical disaster. This can include floods, fire, or person acting with malicious or negligent intent.

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 applications 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 data 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. For this threat, NetApp offers tamperproof snapshots where not even the storage administrator can change protected data before the configured expiry date.

Physical disasters

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

In an enterprise 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. Such replication 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/database can proceed with processing. The resulting performance effect is sometimes 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 ensuring 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 performance and flexibility. In addition, NetApp technology integrates seamlessly with many third-party replication solutions, such as Oracle DataGuard

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.

Database availability

ONTAP is designed to deliver maximum Oracle database availability. A complete description of ONTAP high availability 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 high availability is the HA pair. Each pair contains redundant links to support replication of data to NVRAM. NVRAM is not a write cache. The RAM inside the 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 databases 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, take control of the drives, and applies the required changes that have been stored in NVRAM.

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 subnet(s) 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 includes a complete HA pair which results in a primary path and a secondary path.

Additional paths to additional controllers can also be configured to support relocating data between nodes in a larger cluster, but this 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 causes the network switches to update their media access control (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 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 where 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 an application environment are shown in the table below. There should not be any errors in any application environment, the takeover should instead appear as a short pause in IO processing.

	NFS	AFF	ASA
Planned takeover	15 sec	6-10 sec	2-3 sec
Unplanned takeover	30 sec	6-10 sec	2-3 sec

Checksums and data integrity

ONTAP and its supported protocols include multiple features that protect Oracle database integrity, including both data at rest and data being transmitted over the network network.

Logical data protection within ONTAP consists of three key requirements:

- Data must be protected against data corruption.
- Data must be protected against drive failure.
- Changes to data must be protected against loss.

These three needs are discussed in the following sections.

Network corruption: checksums

The most basic level of data protection is the checksum, which is a special error-detecting code stored alongside the 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. This 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 is done in ONTAP by using parity drives. Data is striped across multiple data drives, and then parity data is generated. This is stored separately from the original 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 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 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 from an unexpected event such as a power failure.

This means any write operation must be safely stored in at least two locations.

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 node. 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 datafiles 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.

Hardware failure protection: 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 HA partner fails, then this in-flight data would no longer be mirrored.

If the local NVRAM reports an error, the node shuts down. This shutdown results in failover to a HA partner controller. No data is lost because the controller experiencing the failure has not acknowledged the write operation.

ONTAP does not permit a failover when the data is out of sync unless the failover 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 is forced because databases maintain large internal caches of data on disk. If a forced failover 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.

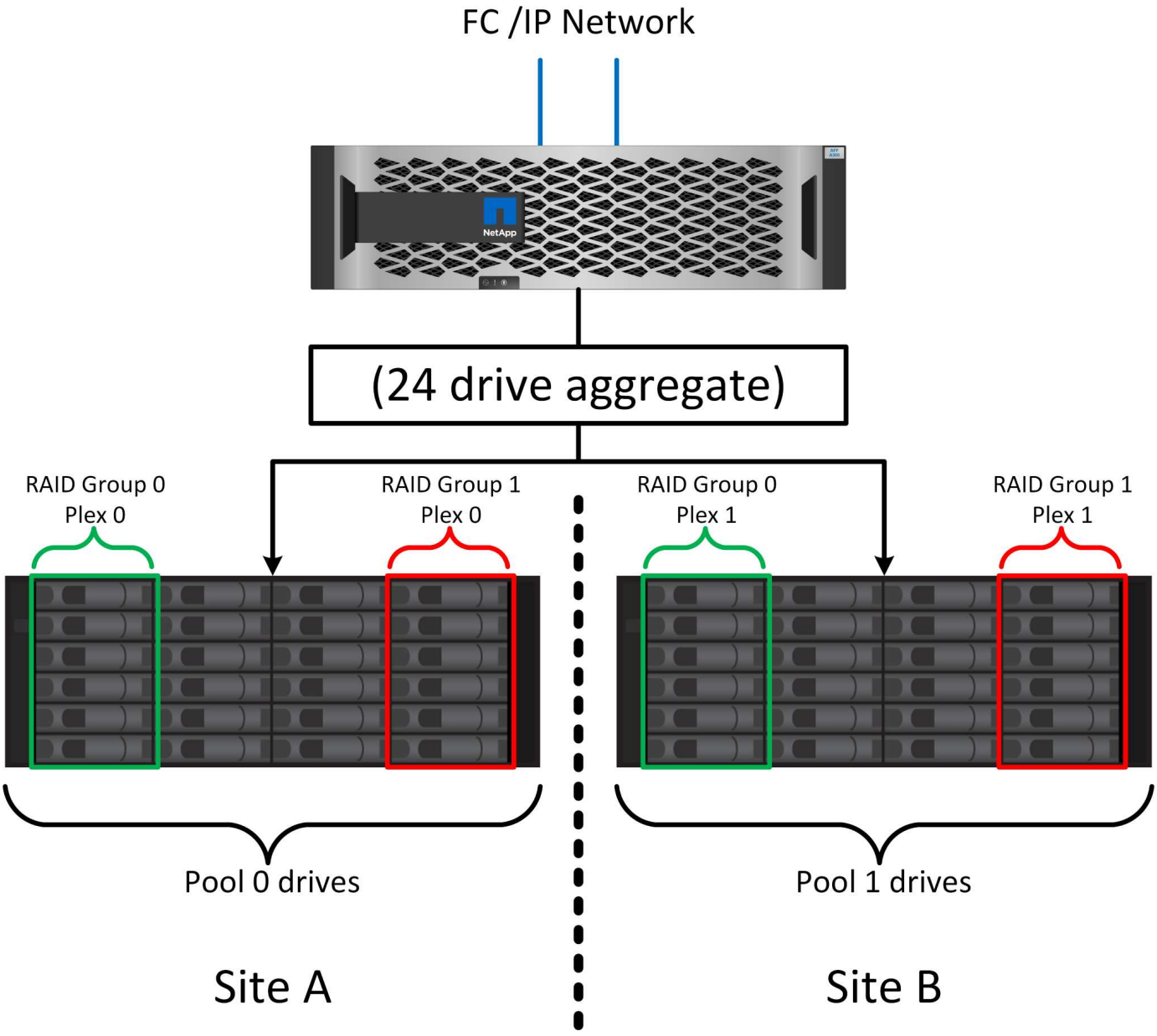
To protect data from this situation, ONTAP allows volumes to be configured for special protection against NVRAM failure. When triggered, this protection mechanism results in a volume entering a state called NVFAIL. This state results in I/O errors that cause a an application shutdown so that they do not use stale data. Data should not be lost because any acknowledged write should be present on the storage array.

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.

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:

- 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.
- 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.
- 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 provides continuous data protection, even after the loss of a site.



The figure above 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 by independent power feeds from independent UPS systems.

Checksums

The topic of checksums is of particular interest to DBAs who are accustomed to using Oracle RMAN streaming backups migrates to snapshot-based backups. One feature of RMAN is that it performs integrity checks during backup operations. Although this feature has some value, its primary benefit is for a database that is not used on a modern storage array. When physical drives are used for an Oracle database, it is nearly certain that corruption eventually occurs as the drives age, a problem that is addressed by array-based checksums in true storage arrays.

With a real storage array, data integrity is protected by using checksums at multiple levels. If data is corrupted in an IP-based network, the Transmission Control Protocol (TCP) layer rejects the packet data and requests retransmission. The FC protocol includes checksums, as does encapsulated SCSI data. After it is on the array, ONTAP has RAID and checksum protection. Corruption can occur, but, as in most enterprise arrays, it is detected and corrected. Typically, an entire drive fails, prompting a RAID rebuild, and database integrity is unaffected. It is still possible for individual bytes on a drive to be damaged by cosmic radiation or failing flash cells. If this happens, the parity check would fail, the drive would be failed out and a RAID rebuild would begin. Once again, data integrity is unaffected. The final line of defense is the use of checksums. If, for example, a catastrophic firmware error on a drive corrupted data in a way that somehow was not detected by a RAID parity check, the checksum would not match and ONTAP would prevent the transfer of a corrupted block before the Oracle database could receive it.

The Oracle datafile and redo log architecture is also designed to deliver the highest possible level of data integrity, even under extreme circumstances. At the most basic level, Oracle blocks include checksum and basic logical checks with almost every I/O. If Oracle has not crashed or taken a tablespace offline, then the data is intact. The degree of data integrity checking is adjustable, and Oracle can also be configured to confirm writes. As a result, almost all crash and failure scenarios can be recovered, and in the extremely rare event of an unrecoverable situation, corruption is promptly detected.

Most NetApp customers using Oracle databases discontinue the use of RMAN and other backup products after migrating to snapshot-based backups. There are still options in which RMAN can be used to perform block-level recovery with SnapCenter. However, on a day-to-day basis, RMAN, NetBackup, and other products are only used occasionally to create monthly or quarterly archival copies.

Some customers choose to run `dbv` periodically to perform integrity checks on their existing databases. NetApp discourages this practice because it creates unnecessary I/O load. As discussed above, if the database was not previously experiencing problems, the chance of `dbv` detecting a problem is close to zero, and this utility creates a very high sequential I/O load on the network and storage system. Unless there is reason to believe corruption exists, such as exposure to a known Oracle bug, there is no reason to run `dbv`.

Backup and recovery basics

Snapshot-based backups

The foundation of Oracle database data protection on ONTAP is NetApp Snapshot technology.

The key values are as follows:

- **Simplicity.** A snapshot is a read-only copy of the contents of a container of data at a specific point in time.
- **Efficiency.** Snapshots require no space at the moment of creation. Space is only consumed when data is changed.
- **Manageability.** A backup strategy based on snapshots is easy to configure and manage because snapshots are a native part of the storage OS. If the storage system is powered on, it is ready to create backups.
- **Scalability.** Up to 1024 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 snapshots.
- Performance is unaffected, whether a volume contains 1024 snapshots or none.

Although many storage vendors offer snapshot technology, the Snapshot technology within ONTAP is unique and offers significant benefits to enterprise application and database environments:

- Snapshot copies are part of the underlying Write-Anywhere File Layout (WAFL). They are not an add-on or external technology. This simplifies management because the storage system is the backup system.
- Snapshot copies do not affect performance, except for some edge cases such as when so much data is stored in snapshots that the underlying storage system fills up.
- The term "consistency group" is often used to refer to a grouping of storage objects that are managed as a consistent collection of data. A snapshot of a particular ONTAP volume constitutes consistency group backup.

ONTAP snapshots also scale better than competing technology. Customers can store 5, 50, or 500 snapshots without affecting performance. The maximum number of snapshots currently allowed in a volume is 1024. If additional snapshot retention is required, there are options to cascade the snapshots to additional volumes.

As a result, protecting a dataset hosted on ONTAP is simple and highly scalable. 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.

Is a snapshot a backup?

One commonly asked question about the use of snapshots as a data protection strategy is the fact that the "real" data and the snapshot data are located on the same drives. Loss of those drives would result in the loss of both the primary data and the backup.

This is a valid concern. Local snapshots are used for day-to-day backup and recovery needs, and in that respect the snapshot is a backup. Close to 99% of all recovery scenarios in NetApp environments rely on snapshots to meet even the most aggressive RTO requirements.

Local snapshots should, however, never be the only backup strategy, which is why NetApp offers technology such as SnapMirror and SnapVault replication to quickly and efficiently replicate snapshots to an independent set of drives. In a properly architected solution with snapshots plus snapshot replication, the use of tape can be minimized to perhaps a quarterly archive or eliminated entirely.

Snapshot-based backups

There are many options for using ONTAP Snapshot copies to protect your data, and snapshots are the basis for many other ONTAP features, including replication, disaster recovery, and cloning. A complete description of snapshot technology is beyond the scope of this document, but the following sections provide a general overview.

There are two primary approaches to creating a snapshot of a dataset:

- Crash-consistent backups
- Application-consistent backups

A crash-consistent backup of a dataset refers to the capture of the entire dataset structure at a single point in time. If the dataset is stored in a single volume, then the process is simple; a Snapshot can be created at any time. If a dataset spans volumes, a consistency group (CG) snapshot must be created. Several options exist for creating CG snapshots, including NetApp SnapCenter software, native ONTAP consistency group features, and user-maintained scripts.

Crash-consistent backups are primarily used when point-of-the-backup recovery is sufficient. When more granular recover is required, application-consistent backups are usually required.

The word "consistent" in "application-consistent" is often a misnomer. For example, placing an Oracle database in backup mode is referred to as an application-consistent backup, but the data is not made consistent or quiesced in any way. The data continue to change throughout the backup. In contrast, most MySQL and Microsoft SQL Server backups do indeed quiesce the data before executing the backup. VMware may or may not make certain files consistent.

Consistency groups

The term "consistency group" refers to the ability of a storage array to manage multiple storage resources as a single image. For example, a database might consist of 10 LUNs. The array must be able to back up, restore, and replicate those 10 LUNs in a consistent manner. Restoration is not possible if the images of the LUNs were not consistent at the point of backup. Replicating those 10 LUNs requires that all the replicas are perfectly synchronized with each other.

The term "consistency group" is not often used when discussing ONTAP because consistency has always been a basic function of the volume and aggregate architecture within ONTAP. Many other storage arrays manage LUNs or file systems as individual units. They could then be optionally configured as a "consistency group" for purposes of data protection, but this is an extra step in the configuration.

ONTAP has always been able to capture consistent local and replicated images of data. Although the various volumes on an ONTAP system are not usually formally described as a consistency group, that is what they are. A snapshot of that volume is a consistency group image, restoration for that snapshot is a consistency group restoration, and both SnapMirror and SnapVault offer consistency group replication.

Consistency group snapshots

Consistency group snapshots (cg-snapshots) are an extension of the basic ONTAP Snapshot technology. A standard snapshot operation creates a consistent image of all data within a single volume, but sometimes it is necessary to create a consistent set of snapshots across multiple volumes and even across multiple storage systems. The result is a set of snapshots that can be used in the same way as a snapshot of just one individual volume. They can be used for local data recovery, replicated for disaster recovery purposes, or cloned as a single consistent unit.

The largest known use of cg-snapshots is for a database environment of approximately 1PB in size spanning 12 controllers. The cg-snapshots created on this system have been used for backup, recovery and cloning.

Most of the time, when a data set spans volumes and write order must be preserved, a cg-snapshot is automatically used by the chosen management software. There is no need to understand the technical details of cg-snapshots in such cases. However, there are situations in which complicated data protection requirements require detailed control over the data protection and replication process. Automation workflows or the use of custom scripts to call the cg-snapshot APIs are some of options. Understanding the best option and the role of cg-snapshot requires a more detailed explanation of the technology.

Creation of a set of cg-snapshots is a two-step process:

1. Establish write fencing on all target volumes.
2. Create snapshots of those volumes while in the fenced state.

Write fencing is established serially. This means that as the fencing process is set up across multiple volumes, write I/O is frozen on the first volume in the sequence as it continues to be committed to volumes that appear later. This might initially appear to violate the requirement for write order to be preserved, but that only applies to I/O that is issued asynchronously on the host and does not depend on any other writes.

For example, a database might issue a lot of asynchronous datafile updates and allow the OS to reorder the I/O and complete them according to its own scheduler configuration. The order of this type of I/O cannot be guaranteed because the application and operating system have already released the requirement to preserve write order.

As a counter example, most database logging activity is synchronous. The database does not proceed with further log writes until the I/O is acknowledged, and the order of those writes must be preserved. If a log I/O arrives on a fenced volume, it is not acknowledged and the application blocks on further writes. Likewise, file system metadata I/O is usually synchronous. For example, a file deletion operation must not be lost. If an operating system with an xfs file system deleted a file and the I/O that updated the xfs file system metadata to remove the reference to that file landed on a fenced volume, then the file system activity would pause. This guarantees the integrity of the file system during cg-snapshot operations.

After write fencing is set up across the target volumes, they are ready for snapshot creation. The snapshots need not be created at precisely the same time because the state of the volumes is frozen from a dependent write point of view. To guard against a flaw in the application creating the cg-snapshots, the initial write fencing includes a configurable timeout in which ONTAP automatically releases the fencing and resumes write processing after a defined number of seconds. If all the snapshots are created before the timeout period lapses, then the resulting set of snapshots are a valid consistency group.

Dependent write order

From a technical point of view, the key to a consistency group is preserving write order and, specifically, dependent write order. For example, a database writing to 10 LUNs writes simultaneously to all of them. Many writes are issued asynchronously, meaning that the order in which they are completed is unimportant and the actual order they are completed varies based on operating system and network behavior.

Some write operations must be present on disk before the database can proceed with additional writes. These critical write operations are called dependent writes. Subsequent write I/O depends on the presence of these writes on disk. Any snapshot, recovery, or replication of these 10 LUNs must make sure that dependent write order is guaranteed. File system updates are another example of write-order dependent writes. The order in which file system changes are made must be preserved or the entire file system could become corrupt.

Strategies

There are two primary approaches to snapshot-based backups:

- Crash-consistent backups
- Snapshot-protected hot backups

A crash-consistent backup of a database refers to the capture of the entire database structure, including datafiles, redo logs, and control files, at a single point in time. If the database is stored in a single volume, then the process is simple; a Snapshot can be created at any time. If a database spans volumes, a consistency group (CG) snapshot must be created. Several options exist for creating CG snapshots, including NetApp

SnapCenter software, native ONTAP consistency group features, and user-maintained scripts.

Crash-consistent Snapshot backups are primarily used when point-of-the-backup recovery is sufficient. Archive logs can be applied under some circumstances, but when more granular point-in-time recovery is required, a online backup is preferable.

The basic procedure for a snapshot-based online backup is as follows:

1. Place the database in `backup` mode.
2. Create a snapshot of all volumes hosting datafiles.
3. Exit `backup` mode.
4. Run the command `alter system archive log current` to force log archiving.
5. Create snapshots of all volumes hosting the archive logs.

This procedure yields a set of snapshots containing datafiles in backup mode and the critical archive logs generated while in backup mode. These are the two requirements for recovering a database. Files such as control files should also be protected for convenience, but the only absolute requirement is protection for datafiles and archive logs.

Although different customers might have very different strategies, almost all of these strategies are ultimately based on the the same principles outlined below.

Snapshot-based recovery

When designing volume layouts for Oracle databases, the first decision is whether to use volume-based NetApp SnapRestore (VBSR) technology.

Volume-based SnapRestore allows a volume to be almost instantly reverted to an earlier point in time. Because all of the data on the volume is reverted, VBSR might not be appropriate for all use cases. For example, if an entire database, including datafiles, redo logs, and archive logs, is stored on a single volume and this volume is restored with VBSR, then data is lost because the newer archive log and redo data are discarded.

VBSR is not required for restore. Many databases can be restored by using file-based single-file SnapRestore (SFSR) or by simply copying files from the snapshot back into the active file system.

VBSR is preferred when a database is very large or when it must be recovered as quickly as possible, and the use of VBSR requires isolation of the datafiles. In an NFS environment, the datafiles of a given database must be stored in dedicated volumes that are uncontaminated by any other type of file. In a SAN environment, datafiles must be stored in dedicated LUNs on dedicated volumes. If a volume manager is used (including Oracle Automatic Storage Management [ASM]), the diskgroup must also be dedicated to datafiles.

Isolating datafiles in this manner allows them to be reverted to an earlier state without damaging other file systems.

Snapshot reserve

For each volume with Oracle data in a SAN environment, the `percent-snapshot-space` should be set to zero because reserving space for a snapshot in a LUN environment is not useful. If the fractional reserve is set to 100, a snapshot of a volume with LUNs requires enough free space in the volume, excluding the snapshot reserve, to absorb 100% turnover of all of the data. If the fractional reserve is set to a lower value, then a correspondingly smaller amount of free space is required, but it always excludes the snapshot reserve. This means that the snapshot reserve space in a LUN environment is wasted.

In an NFS environment, there are two options:

- Set the `percent-snapshot-space` based on expected snapshot space consumption.
- Set the `percent-snapshot-space` to zero and manage active and snapshot space consumption collectively.

With the first option, `percent-snapshot-space` is set to a nonzero value, typically around 20%. This space is then hidden from the user. This value does not, however, create a limit on utilization. If a database with a 20% reservation experiences 30% turnover, the snapshot space can grow beyond the bounds of the 20% reserve and occupy unreserved space.

The main benefit of setting a reserve to a value such as 20% is to verify that some space is always available for snapshots. For example, a 1TB volume with a 20% reserve would only permit a database administrator (DBA) to store 800GB of data. This configuration guarantees at least 200GB of space for snapshot consumption.

When `percent-snapshot-space` is set to zero, all space in the volume is available to the end user, which delivers better visibility. A DBA must understand that, if he or she sees a 1TB volume that leverages snapshots, this 1TB of space is shared between active data and Snapshot turnover.

There is no clear preference between option one and option two among end users.

ONTAP and third-party snapshots

Oracle Doc ID 604683.1 explains the requirements for third-party snapshot support and the multiple options available for backup and restore operations.

The third-party vendor must guarantee that the company's snapshots conform to the following requirements:

- Snapshots must integrate with Oracle's recommended restore and recovery operations.
- Snapshots must be database crash consistent at the point of the snapshot.
- Write ordering is preserved for each file within a snapshot.

ONTAP and NetApp Oracle management products comply with these requirements.

SnapRestore

Rapid data restoration in ONTAP from a snapshot is delivered by NetApp SnapRestore technology.

When a critical dataset is unavailable, 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 datasets. Even petabyte-scale databases can be completely restored with just a few minutes of effort.

There are two forms of SnapRestore - file/LUN-based and volume-based.

- Individual files or LUNs can be restored in seconds, whether it is a 2TB LUN or a 4KB file.
- The container of files or LUNs can be restored in seconds, whether it is 10GB or 100TB of data.

A "container of files or LUNs" would typically refer to a FlexVol volume. For example, you may have 10 LUNs that make up a LVM diskgroup in a single volume, or a volume might store the NFS home directories of 1000 users. Rather than executing a restore operation for each individual file or LUN, you can restore the entire

volume as a single operation. This process also work with scale-out containers that include multiple volumes, such as a FlexGroup or an ONTAP Consistency Group.

The reason SnapRestore works so quickly and efficiently is due to the nature of a snapshot, which is essentially a parallel read-only view of the contents of a volume at a specific point in time. The active blocks are the real blocks that can be changed, while the snapshot is a read-only view into the state of the blocks that constitute the files and LUNs at the time the snapshot was created.

ONTAP only permits read-only access to snapshot data, but the data can be reactivated with SnapRestore. The snapshot is reenabled as a read-write view of the data, returning the data to its prior state. SnapRestore can operate at the volume or the file level. The technology is essentially the same with a few minor differences in behavior.

Volume SnapRestore

Volume-based SnapRestore returns the entire volume of data to an earlier state. This operation does not require data movement, meaning that the restore process is essentially instantaneous, although the API or CLI operation might take a few seconds to be processed. Restoring 1GB of data is no more complicated or time-consuming than restoring 1PB of data. This capability is the primary reason many enterprise customers migrate to ONTAP storage systems. It delivers an RTO measured in seconds for even the largest datasets.

One drawback to volume-based SnapRestore is caused by the fact that changes within a volume are cumulative over time. Therefore, each snapshot and the active file data are dependent on the changes leading up to that point. Reverting a volume to an earlier state means discarding all the subsequent changes that had been made to the data. What is less obvious, however, is that this includes subsequently created snapshots. This is not always desirable.

For example, a data retention SLA might specify 30 days of nightly backups. Restoring a dataset to a snapshot created five days ago with volume SnapRestore would discard all the snapshots created on the previous five days, violating the SLA.

There are a number of options available to address this limitation:

1. Data can be copied from a prior snapshot, as opposed to performing a SnapRestore of the entire volume. This method works best with smaller datasets.
2. A snapshot can be cloned rather than restored. The limitation to this approach is that the source snapshot is a dependency of the clone. Therefore, it cannot be deleted unless the clone is also deleted or is split into an independent volume.
3. Use of file-based SnapRestore.

File SnapRestore

File-based SnapRestore is a more granular snapshot-based restoration process. Rather than reverting the state of an entire volume, the state of an individual file or LUN is reverted. No snapshots need to be deleted, nor does this operation create any dependency on a prior snapshot. The file or LUN becomes immediately available in the active volume.

No data movement is required during a SnapRestore restore of a file or LUN. However, some internal metadata updates are required to reflect the fact that the underlying blocks in a file or LUN now exist in both a snapshot and the active volume. There should be no effect on performance, but this process blocks the creation of snapshots until it is complete. The processing rate is approximately 5GBps (18TB/hour) based on the total size of the files restored.

Online backups

Two sets of data are required to protect and recover an Oracle database in backup mode. Note that this is not the only Oracle backup option, but it is the most common.

- A snapshot of the datafiles in backup mode
- The archive logs created while the datafiles were in backup mode

If complete recovery including all committed transactions is required, a third item is required:

- A set of current redo logs

There are a number of ways to drive recovery of an online backup. Many customers restore snapshots by using the ONTAP CLI and then using Oracle RMAN or sqlplus to complete the recovery. This is especially common with large production environments in which the probability and frequency of database restores is extremely low and any restore procedure is handled by a skilled DBA. For complete automation, solutions such as NetApp SnapCenter include an Oracle plug-in with both command- line and graphical interfaces.

Some large-scale customers have taken a simpler approach by configuring basic scripting on the hosts to place the databases in backup mode at a specific time in preparation for a scheduled snapshot. For example, schedule the command `alter database begin backup` at 23:58, `alter database end backup` at 00:02, and then schedule snapshots directly on the storage system at midnight. The result is a simple, highly scalable backup strategy that requires no external software or licenses.

Data layout

The simplest layout is to isolate datafiles into one or more dedicated volumes. They must be uncontaminated by any other file type. This is to make sure that the datafile volumes can be rapidly restored through a SnapRestore operation without destroying an important redo log, controlfile, or archive log.

SAN has similar requirements for datafile isolation within dedicated volumes. With an operating system such as Microsoft Windows, a single volume might contain multiple datafile LUNs, each with an NTFS file system. With other operating systems, there is generally a logical volume manager. For example, with Oracle ASM, the simplest option would be to confine the LUNs of an ASM disk group to a single volume that can be backed up and restored as a unit. If additional volumes are required for performance or capacity management reasons, creating an additional disk group on the new volume results in simpler management.

If these guidelines are followed, snapshots can be scheduled directly on the storage system with no requirement for performing a consistency group snapshot. The reason is that Oracle backups do not require datafiles to be backed up at the same time. The online backup procedure was designed to allow datafiles to continue to be updated as they are slowly streamed to tape over the course of hours.

A complication arises in situations such as the use of an ASM disk group that is distributed across volumes. In these cases, a cg-snapshot must be performed to make sure that the ASM metadata is consistent across all constituent volumes.

Caution: Verify that the ASM `spfile` and `passwd` files are not in the disk group hosting the datafiles. This interferes with the ability to selectively restore datafiles and only datafiles.

Local recovery procedure—NFS

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Recover the datafile volume(s) to the snapshot immediately prior to the desired restore point.
3. Replay archive logs to the desired point.
4. Replay current redo logs if complete recovery is desired.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored or rman/sqlplus can be directed to the data in the snapshot directory.

In addition, for smaller databases, datafiles can be recovered by an end user directly from the `.snapshot` directory without assistance from automation tools or storage administrators to execute a `snaprestore` command.

Local recovery procedure—SAN

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Quiesce the disk group(s) hosting the datafiles. The procedure varies depending on the logical volume manager chosen. With ASM, the process requires dismounting the disk group. With Linux, the file systems must be dismounted, and the logical volumes and volume groups must be deactivated. The objective is to stop all updates on the target volume group to be restored.
3. Restore the datafile disk groups to the snapshot immediately prior to the desired restore point.
4. Reactivate the newly restored disk groups.
5. Replay archive logs to the desired point.
6. Replay all redo logs if complete recovery is desired.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored by taking the archive log LUNs offline and performing a restore. This is also an example in which dividing up archive logs into dedicated volumes is useful. If the archive logs share a volume group with redo logs, then the redo logs must be copied elsewhere before restoration of the overall set of LUNs. This step prevents the loss of those final recorded transactions.

Storage Snapshot Optimized backups

Snapshot-based backup and recovery became even simpler back when Oracle 12c was released because there is no need to place a database in hot backup mode. The result is an ability to schedule snapshot-based backups directly on a storage system and still preserve the ability to perform complete or point-in-time recovery.

Although the hot backup recovery procedure is more familiar to DBAs, it has, for a long time, been possible to use snapshots that were not created while the database was in hot backup mode. Extra manual steps were required with Oracle 10g and 11g during recovery to make the database consistent. With Oracle 12c, `sqlplus` and `rman` contain the extra logic to replay archive logs on datafile backups that were not in hot backup mode.

As discussed previously, recovering a snapshot-based hot backup requires two sets of data:

- A snapshot of the datafiles created while in backup mode
- The archive logs generated while the datafiles were in hot backup mode

During recovery, the database reads metadata from the datafiles to select the required archive logs for recovery.

Storage snapshot-optimized recovery requires slightly different datasets to accomplish the same results:

- A snapshot of the datafiles, plus a method to identify the time the snapshot was created
- Archive logs from the time of the most recent datafile checkpoint through the exact time of the snapshot

During recovery, the database reads metadata from the datafiles to identify the earliest archive log required. Full or point-in-time recovery can be performed. When performing a point-in-time recovery, it is critical to know the time of the snapshot of the datafiles. The specified recovery point must be after the creation time of the snapshots. NetApp recommends adding at least a few minutes to the snapshot time to account for clock variation.

For complete details, see Oracle's documentation on the topic, "Recovery Using Storage Snapshot Optimization" available in various releases of the Oracle 12c documentation. Also, see Oracle Document ID Doc ID 604683.1 regarding Oracle third-party snapshot support.

Data layout

The simplest layout is to isolate the datafiles into one or more dedicated volumes. They must be uncontaminated by any other file type. This is to make sure that the datafile volumes can be rapidly restored with a SnapRestore operation without destroying an important redo log, controlfile, or archive log.

SAN has similar requirements for datafile isolation within dedicated volumes. With an operating system such as Microsoft Windows, a single volume might contain multiple datafile LUNs, each with an NTFS file system. With other operating systems, there is generally a logical volume manager as well. For example, with Oracle ASM, the simplest option would be to confine disk groups to a single volume that can be backed up and restored as a unit. If additional volumes are required for performance or capacity management reasons, creating an additional disk group on the new volume results in easier management.

If these guidelines are followed, snapshots can be scheduled directly on ONTAP with no requirement for performing a consistency group snapshot. The reason is that snapshot-optimized backups do not require that datafiles be backed up at the same time.

A complication arises in situations such as an ASM disk group that is distributed across volumes. In these cases, a cg-snapshot must be performed to make sure that the ASM metadata is consistent across all constituent volumes.

[Note]Verify that the ASM spfile and passwd files are not in the disk group hosting the datafiles. This interferes with the ability to selectively restore datafiles and only datafiles.

Local recovery procedure—NFS

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Recover the datafile volume(s) to the snapshot immediately prior to the desired restore point.
3. Replay archive logs to the desired point.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored, or `rman` or `sqlplus` can be directed to the data in the `.snapshot` directory.

In addition, for smaller databases, datafiles can be recovered by an end user directly from the `.snapshot` directory without assistance from automation tools or a storage administrator to execute a SnapRestore command.

Local recovery procedure—SAN

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Quiesce the disk group(s) hosting the datafiles. The procedure varies depending on the logical volume manager chosen. With ASM, the process requires dismounting the disk group. With Linux, the file systems must be dismounted, and the logical volumes and volume groups are deactivated. The objective is to stop all updates on the target volume group to be restored.
3. Restore the datafile disk groups to the snapshot immediately prior to the desired restore point.
4. Reactivate the newly restored disk groups.
5. Replay archive logs to the desired point.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored by taking the archive log LUNs offline and performing a restore. This is also an example in which dividing up archive logs into dedicated volumes is useful. If the archive logs share a volume group with redo logs, the redo logs must be copied elsewhere before restoration of the overall set of LUNs to avoid losing the final recorded transactions.

Full recovery example

Assume the datafiles have been corrupted or destroyed and full recovery is required. The procedure to do so is as follows:

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover automatic;
Media recovery complete.
SQL> alter database open;
Database altered.
SQL>
```

Point-in-time recovery example

The entire recovery procedure is a single command: `recover automatic`.

If point-in-time recovery is required, the timestamp of the snapshot(s) must be known and can be identified as follows:

```
Cluster01::> snapshot show -vserver vserver1 -volume NTAP_oradata -fields
create-time
vserver   volume        snapshot   create-time
--------  ------------  ---------  ------------------------
vserver1  NTAP_oradata  my-backup  Thu Mar 09 10:10:06 2017
```

The snapshot creation time is listed as March 9th and 10:10:06. To be safe, one minute is added to the snapshot time:

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover database until time '09-MAR-2017 10:44:15' snapshot time '09-
MAR-2017 10:11:00';
```

The recovery is now initiated. It specified a snapshot time of 10:11:00, one minute after the recorded time to account for possible clock variance, and a target recovery time of 10:44. Next, sqlplus requests the archive logs required to reach the desired recovery time of 10:44.

```
ORA-00279: change 551760 generated at 03/09/2017 05:06:07 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_31_930813377.dbf
ORA-00280: change 551760 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 552566 generated at 03/09/2017 05:08:09 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_32_930813377.dbf
ORA-00280: change 552566 for thread 1 is in sequence #32
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 553045 generated at 03/09/2017 05:10:12 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_33_930813377.dbf
ORA-00280: change 553045 for thread 1 is in sequence #33
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 753229 generated at 03/09/2017 05:15:58 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_930813377.dbf
ORA-00280: change 753229 for thread 1 is in sequence #34
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;
Database altered.
SQL>
```

Complete recovery of a database using snapshots using the `recover automatic` command does not require specific licensing, but point-in-time recovery using `snapshot time` requires the Oracle Advanced Compression license.

Database management and automation tools

The primary value of ONTAP in an Oracle database environment comes from the core ONTAP technologies such as instant Snapshot copies, simple SnapMirror replication, and efficient creation of FlexClone volumes.

In some cases, simple configuration of these core features directly on ONTAP meets requirements, but more complicated needs require an orchestration layer.

SnapCenter

SnapCenter is the flagship NetApp data protection product. At a very low level, it is similar to the SnapManager products in terms of how it executes database backups, but it was built from the ground up to deliver a single-pane-of-glass for data protection management on NetApp storage systems.

SnapCenter includes the basic functions such as snapshot-based backups and restores, SnapMirror and SnapVault replication, and other features required to operate at scale for large enterprises. These advanced

features include an expanded role-based access control (RBAC) capability, RESTful APIs to integrate with third-party orchestration products, nondisruptive central management of SnapCenter plug-ins on database hosts, and a user interface designed for cloud-scale environments.

REST

ONTAP also contains a rich RESTful API set. This allows 3rd party vendors to create data protection and other management application with deep integration with ONTAP. Furthermore, the RESTful API is easy to consume by customers who wish to create their own automation workflows and utilities.

Oracle disaster recovery

Overview

Disaster recovery refers to restoring data services after a catastrophic event, such as a fire that destroys a storage system or even an entire site.

This documentation replaces previously published technical reports *TR-4591: Oracle Data Protection* and *TR-4592: Oracle on MetroCluster.*

Disaster recovery can be accomplished by simple replication of data using SnapMirror, of course, with many customers updating mirrored replicas as often as hourly.

For most customers, DR requires more than just possessing a remote copy of data, it requires the ability to rapidly make use of that data. NetApp offers two technologies that address this need - MetroCluster and SnapMirror active sync

MetroCluster refers to ONTAP in a hardware configuration that includes low-level synchronously mirrored storage and numerous additional features. Integrated solutions such as MetroCluster simplify today's complicated, scale-out database, application, and virtualization infrastructures. It replaces multiple, external data protection products and strategies with one simple, central storage array. It also provides integrated backup, recovery, disaster recovery, and high availability (HA) within a single clustered storage system.

SnapMirror active sync (SM-as) is based on SnapMirror Synchronous. With MetroCluster, each ONTAP controller is responsible for replicating its drive data to a remote location. With SnapMirror active sync, you essentially have two different ONTAP systems maintaining independent copies of your LUN data, but cooperating to present a single instance of that LUN. From a host point of view, it's a single LUN entity.

SM-as and MCC comparison

SM-as and MetroCluster are similar in overall functionality, but there are important differences in the way in which RPO=0 replication was implemented and how it is managed. SnapMirror asynchronous and synchronous can also be used as part of a DR plan, but they are not designed as HA repliation technologies.

- A MetroCluster configuration is more like one integrated cluster with nodes distributed across sites. SM-as behaves like two otherwise independent clusters that are cooperating in serving up select RPO=0 synchronously replicated LUNs.
- The data in a MetroCluster configuration is only accessible from one particular site at any given time. A second copy of the data is present on the opposite site, but the data is passive. It cannot be accessed without a storage system failover.
- MetroCluster and SM-as perform mirroring occurs at different levels. MetroCluster mirroring is performed at the RAID layer. The low-level data is stored in a mirrored format using SyncMirror. The use of mirroring is

virtually invisible up at the LUN, volume, and protocol layers.

- In contrast, SM-as mirroring occurs at the protocol layer. The two clusters are overall independent clusters. Once the two copies of data are in sync, the two clusters only need to mirror writes. When a write occurs on one cluster, it is replicated to the other cluster. The write is only acknowledged to the host when the write has completed on both sites. Other than this protocol splitting behavior, the two clusters are otherwise normal ONTAP clusters.
- The primary role for MetroCluster is large-scale replication. You can replicate an entire array with RPO=0 and near-zero RTO. This simplifies the failover process because there is only one "thing" to fail over, and it scales extremely well in terms of capacity and IOPS.
- One key use case for SM-as is granular replication. Sometimes you don’t want to replicate all data as a single unit, or you need to be able to selectively fail over certain workloads.
- Another key use case for SM-as is for active-active operations, where you want fully usable copies of data to be available on two different clusters located in two different locations with identical performance characteristics and, if desired, no requirement to stretch the SAN across sites. You can have your applications already running on both sites, which reduces the overall RTO during failover operations.

MetroCluster

Disaster Recovery with MetroCluster

Metrocluster is an ONTAP feature that can protect your Oracle databases with RPO=0 synchronous mirroring across sites, and it scales up to support hundreds of databases on a single MetroCluster system.

It’s also simple to use. The use of MetroCluster does not necessarily add to or change any best ractices for operating a enterprise applications and databases.

The usual best practices still apply, and if your needs only require RPO=0 data protection then that need is met with MetroCluster. However, most customers use MetroCluster not only for RPO=0 data protection, but also to improve RTO during disaster scenarios as well as provide transparent failover as part of site maintenance activities.

Physical architecture

Understanding how Oracle databases operate in a MetroCluster environment requires some explanation of physical design of a MetroCluster system.

This documentation replaces previously published technical report *TR-4592: Oracle on MetroCluster.*

MetroCluster is avialable in 3 different configurations

- HA pairs with IP connectivity
- HA pairs with FC connectivity
- Single controller with FC connectivity

The term 'connectivity' refers to the cluster connection used for cross-site replication. It does not refer to the host protocols. All host-side protocols are supported as usual in a MetroCluster configuration irrespective of the type of connection used for inter-cluster communication.

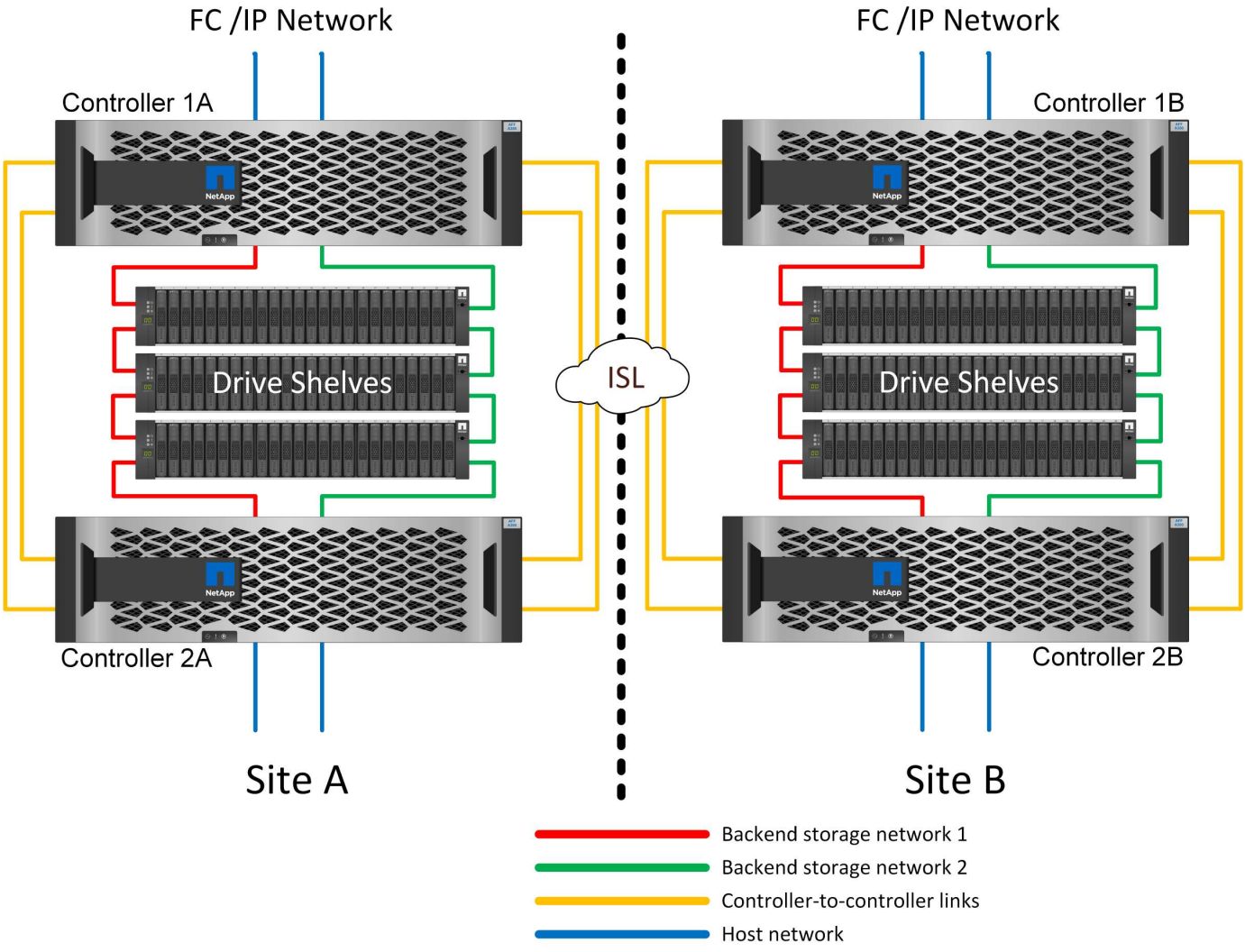
MetroCluster IP

The HA-pair MetroCluster IP configuration uses two or four 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.

Most customers are choosing IP connectivity because the infrastructure requirements are simpler. In the past, high-speed cross-site connectivity was generally easier to provision using dark fibre and FC switches, but today high-speed, low latency IP circuits are more readily available.

The architecture is also simpler because the only cross-site connections are for the controllers. In FC SAN attached MetroClusters, a controller writes directly to the drives on the opposite site and thus requires additional SAN connections, switches, and bridges. In contrast, a controller in an IP configuration writes to the opposite drives via the controller.

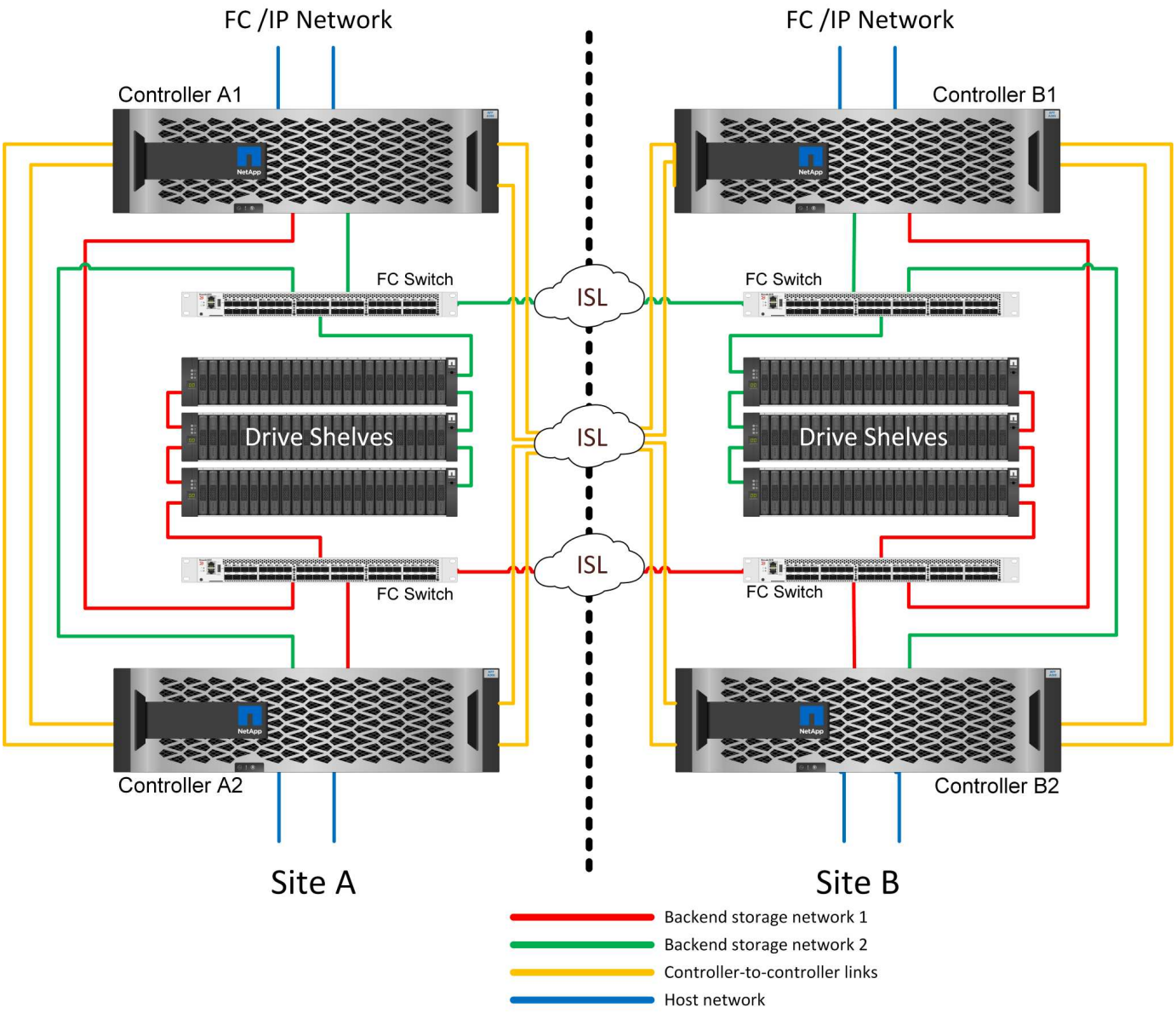
For additional information, refer to the official ONTAP documentation and MetroCluster IP Solution Architecture and Design.



HA-Pair FC SAN-attached MetroCluster

The HA-pair MetroCluster FC configuration uses two or four 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.



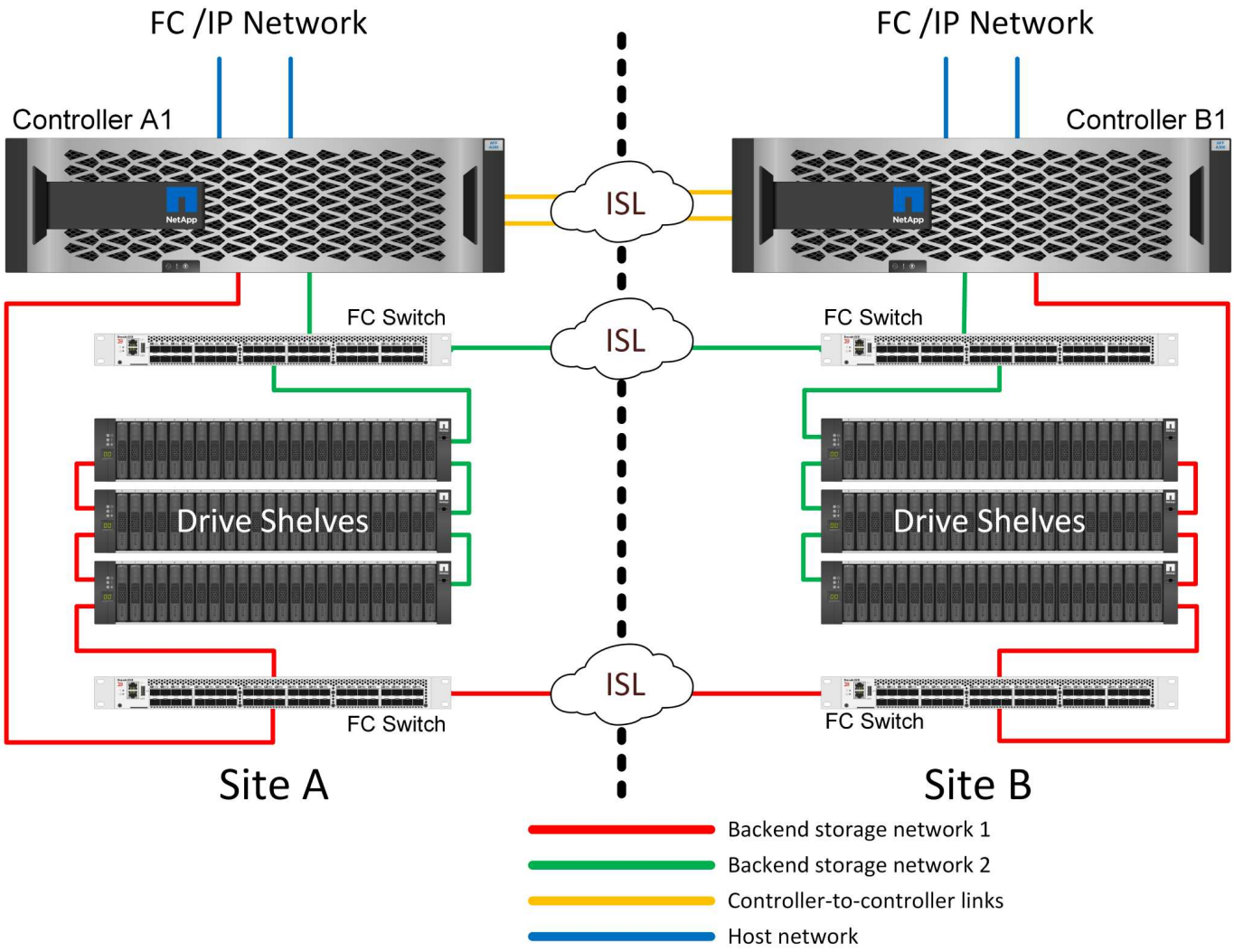
Some multisite infrastructures are not designed for active-active operations, but rather are used more as a primary site and disaster recovery site. In this situation, an HA-pair MetroCluster option is generally preferable for the following reasons:

- 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.

Two-node FC SAN-attached 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 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.

MetroCluster resiliency features

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.

Logical architecture

Understanding how Oracle databases operate in a MetroCluster environment alsop requires some explanation of the logical functionality of a MetroCluster system.

Site failure protection: NVRAM and MetroCluster

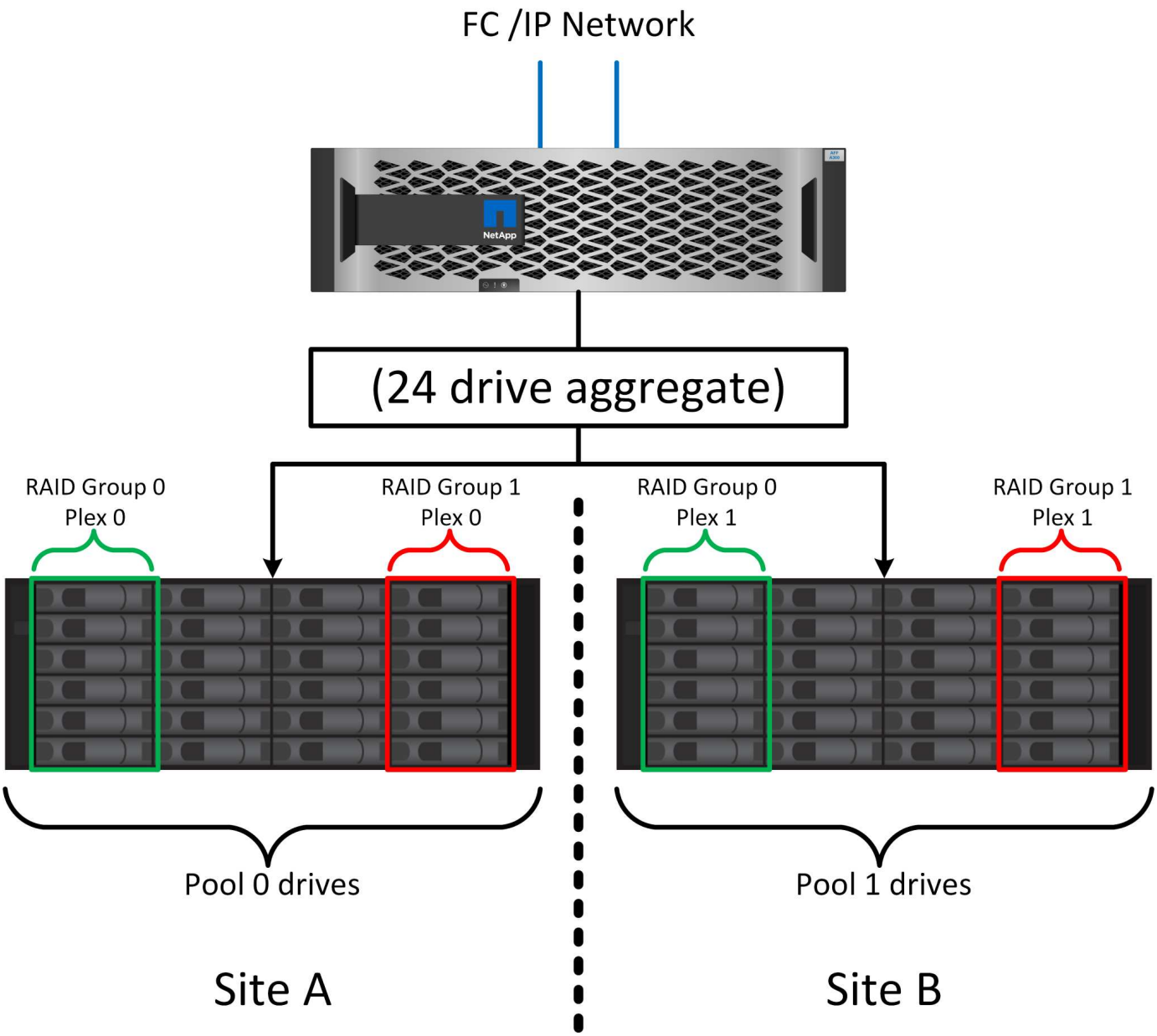
MetroCluster extends NVRAM data protection in the following ways:

- In a two-node configuration, NVRAM data is replicated using the Inter-Switch 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.



The figure above 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 and other applications are especially vulnerable to corruption if a failover or switchover is forced because they 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 cache no longer reflects the state of the data on disk.

To prevent 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 an application crash. This crash causes the applications 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.

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.

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 environments. However, most applications require 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 environments are kept online and serving data until connectivity is repaired or a formal decision is made to shut down the environment to protect data. A requirement for automatic shutdown of the application 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 with Fabric Attached MetroCluster

Automatic unattended switchover (AUSO) is a Fabric Attached 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 Fabric Attached 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.

MetroCluster tiebreaker with fabric attached MetroCluster

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.

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 application to continue operating would effectively discard that critical data.
- An application 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. Applications 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.

ONTAP Mediator with MetroCluster IP

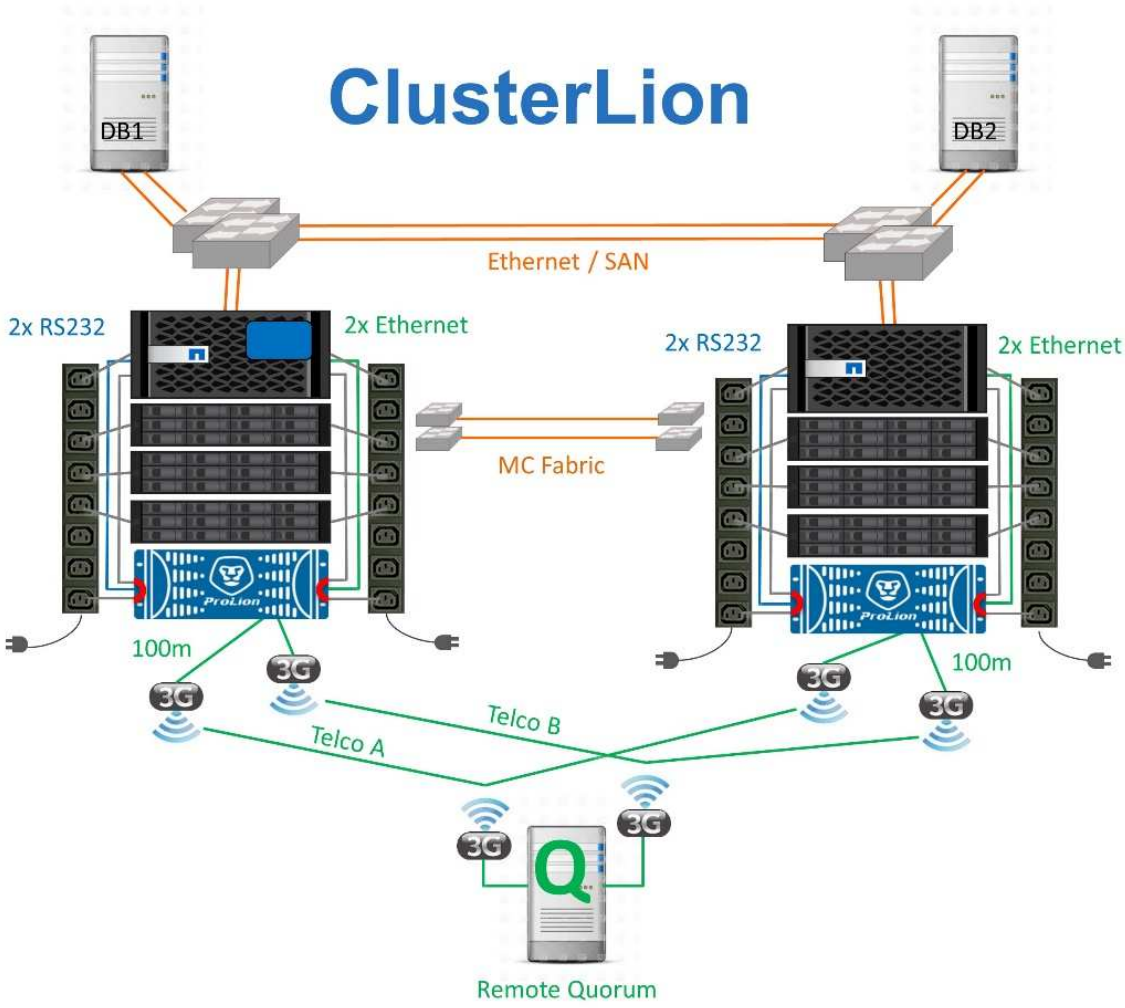
The ONTAP Mediator is used with MetroCluster IP and certain other ONTAP solutions. It functions as a traditional tiebreaker service, much like the MetroCluster Tiebreaker software discussed above, but also includes a critical feature - performing automated unattended switchover.

A fabric-attached MetroCluster has direct access to the storage devices on the opposite site. This allows one MetroCluster controller to monitor the health of the other controllers by reading heartbeat data from the drives. This allows one controller to recognize the failure of another controller and perform a switchover.

In contrast, the MetroCluster IP architecture routes all I/O exclusively through the controller-controller connection; there is no direct access to storage devices on the remote site. This limits the ability of a controller to detect failures and perform a switchover. The ONTAP Mediator is therefore required as a tiebreaker device to detect site loss and automatically perform a switchover.

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. Furthermore, ClusterLion can perform additional network level monitor and execute post-switchover operations. Complete documentation is available from ProLion.



- 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.
- Support includes both Fabric-attached MetroCluster and MetroCluster IP

SyncMirror

The foundation of Oracle data protection with a MetroCluster system is SyncMirror, a maximum-performance, scale-out synchronous mirroring technology.

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, if a database is writing a log, or a VMware guest is being patched, a write must never be lost. As a protocol level, 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. If you need 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?

SyncMirror data availability

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 non-replicated 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 un-synchronized. 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, NetApp offers SnapMirror Synchronous (SM-S). Application-level options also exist, such as Oracle DataGuard or SQL Server Always On Availability Groups. OS-level disk mirroring can be an option. Consult your NetApp or partner account team for additional information and options.

MetroCluster and NVFAIL

NVFAIL is a general data integrity feature in ONTAP that is designed to maximize data integrity protection with databases.

This section expands on the explanation of basic ONTAP NVFAIL to cover MetroCluster-specific topics.

With MetroCluster, 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 the risk of data loss. In a MetroCluster configuration, if NVRAM and the underlying aggregate plexes are in sync, it is safe to proceed with the switchover without the 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 applications 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 an application shutdown so that they do not use stale data. Data should not be lost because any acknowledged writes are still present on the storage system, and with databases 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.

Manually forced NVFAIL

The safest option to force a switchover with an application cluster (including VMware, Oracle RAC, and others) that is distributed across sites 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 is complete, 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.

dr-force-nvfail

As a general safety measure, set the `dr-force-nvfail` flag on all volumes that might be accessed from a remote site during normal operations, meaning they are activities used prior to failover. The result of this setting is that select remote volumes become unavailable when they enter `in-nvfailed-state` during a switchover. After the switchover is complete, the `in-nvfailed-state` flag must be cleared, and the LUNs must be placed online. After these activities are complete, the applications can be restarted. These tasks can be automated to reduce the RTO.

The result is like using the `-force-nvfail-all` flag for manual switchovers. 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.

There are two critical requirements for an environment that does not use `dr-force-nvfail` on application 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 have 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.

Oracle single-instance

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 the Oracle on ONTAP documentation.

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.

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.

Oracle Extended RAC

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 extended RAC clusters, 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 extended RAC is required, SnapMirror active sync should be considered in place of MetroCluster. SM-as replication allows a specific replica of the data to be preferred. Therefore, a extended 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.

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 Oracle RAC with ONTAP.

Two-site configuration

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

RAC voting files

The first consideration when deploying extended 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 files. With a single-site RAC configuration, a single voting resource is sufficient as long as the underlying storage system offers HA capabilities.

In earlier versions of Oracle, the voting files were placed on physical storage devices, but in current versions of Oracle the voting files are stored in ASM diskgroups.

> Oracle RAC is supported with NFS. During the grid installation process, a set of ASM processes is created to present the NFS location used for grid files as an ASM diskgroup. The process is nearly transparent to the end user and requires no ongoing ASM management after the installation is complete.

The first requirement in a two-site configuration is making sure that each site can always access more than half of the voting files in a way that guarantees a nondisruptive disaster recovery process. This task was simple before the voting files were stored in ASM diskgroups, but today administrators need to understand basic principles of ASM redundancy.

ASM diskgroups have three options for redundancy `external`, `normal`, and `high`. In other words, unmirrored, mirrored, and 3-way mirrored. A newer option called `Flex` is also available, but rarely used. The redundancy level and placement of the redundant devices controls what happens in failure scenarios. For example:

- Placing the voting files on a `diskgroup` with `external` redundancy resource guarantees eviction of one site if intersite connectivity is lost.
- Placing the voting files on a `diskgroup` with `normal` redundancy with only one ASM disk per site guarantees node eviction on both sites if intersite connectivity is lost because neither site would have a majority quorum.
- Placing the voting files on a `diskgroup` with `high` redundancy with two disks on one site and a single disk on the other site allows for active-active operations when both sites are operational and mutually reachable. However, if the single-disk site is 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@host-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@host-a lib]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
```

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

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@host-a ~]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
 [root@host-a ~]# grep -i 'master node' /grid/diag/crs/host-
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 `host-a` has an ID of `1`. This fact means that `host-a` is not the master node. The identity of the master node can be confirmed with the command `olsnodes -n`.

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

The node with an ID of `2` is `host-b`, which is the master node. In a configuration with equal numbers of nodes on each site, the site with `host-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@host-a ~]#  /grid/bin/ocrconfig -showbackup
host-b     2017/05/05 05:39:53     /grid/cdata/host-cluster/backup00.ocr
0
host-b     2017/05/05 01:39:53     /grid/cdata/host-cluster/backup01.ocr
0
host-b     2017/05/04 21:39:52     /grid/cdata/host-cluster/backup02.ocr
0
host-a     2017/05/04 02:05:36     /grid/cdata/host-cluster/day.ocr     0
host-a     2017/04/22 02:05:17     /grid/cdata/host-cluster/week.ocr     0
```

This example shows that the master node is `host-b`. It also indicates a change in the master node from `host-a` to `host-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 diskgroup that services the entire environment and an equal number of RAC nodes on each site. The diskgroup should be placed on the site that that contains 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 diskgroup 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.

Three-site configurations

An extended 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. The Oracle tiebreaker configuration may be as simple as placing a member of the ASM diskgroup used for voting on a 3rd site, and may also include an operational instance on the 3rd site to ensure there is an odd number of nodes in the RAC cluster.

Consult the Oracle documentation on “quorum failure group” for important information on using NFS in an extended RAC configuration. In summary, the NFS mount options may need to be modified to include the soft option to ensure that loss of connectivity to the 3rd site hosting quorum resources does not hang the primary Oracle servers or Oracle RAC processes.

SnapMirror active sync

Overview

SnapMirror active sync allows you to build ultra high availability Oracle database environments where LUNs are available from two different storage clusters.

With SnapMirror active sync, there is no "primary" and "secondary" copy of the data. Each cluster can serve read IO from its local copy of the data, and each cluster will replicate a write to its partner. The result is symmetric IO behavior.

Among other options, this allows you to run Oracle RAC as an extended cluster with operational instances on both sites. Alternatively, you could build RPO=0 active-passive database clusters where single instance databases can be moved across sites during a site outage, and this process can be automated through products like Pacemaker or VMware HA. The foundation for all of these options is synchronous replication managed by SnapMirror active sync.

Synchronous replication

In normal operation, SnapMirror active sync provides RPO=0 synchronous replica at all times, with one exception. If data cannot be replicated, ONTAP will release the requirement to replicate data and resume serving IO on one site while the LUNs on the other site are taken offline.

Storage hardware

Unlike other storage disaster recovery solutions, SnapMirror active sync offers asymmetric platform flexibility. The hardware at each site does not need to be identical. This capability allows you to right-size the hardware used to support SnapMirror active sync. The remote storage system can be identical to the primary site if it needs to support a full production workload, but if a disaster results in reduced I/O, than a smaller system at the remote site might be more cost-effective.

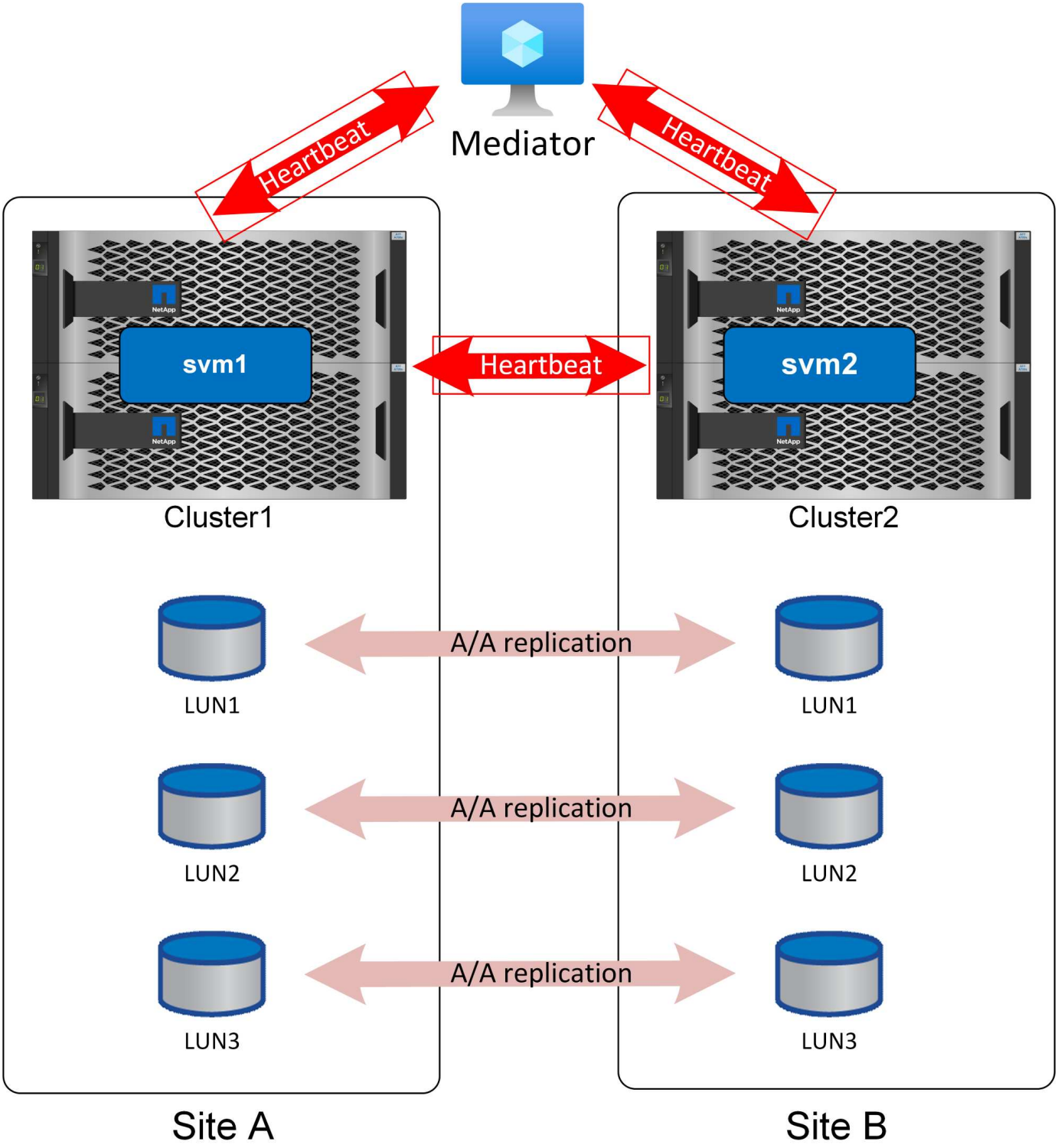
ONTAP mediator

The ONTAP Mediator is a software application that is downloaded from NetApp support, and is typically deployed on a small virtual machine. The ONTAP Mediator is not a tiebreaker when used with SnapMirror active sync. It is an alternate communication channel for the two clusters that participate in SnapMirror active sync replication. Automated operations are driven by ONTAP based on the responses received from the partner via direct connections and via the mediator.

ONTAP Mediator

The mediator is required for safely automating failover. Ideally, it would be placed on an independent 3rd site, but it can still function for most needs if colocated with one of the clusters participating in replication.

The mediator is not really a tiebreaker, although that is effectively the function it provides. The mediator helps in determining the state of the cluster nodes and assists in the automatic switchover process in the event of a site failure. Mediator does not transfer data under any circumstances.



The #1 challenge with automated failover is the split-brain problem, and that problem arises if your two sites lose connectivity with each other. What should happen? You do not want to have two different sites designate themselves as the surviving copies of the data, but how can a single site tell the difference between actual loss of the opposite site and an inability to communicate with the opposite site?

This is where the mediator enters the picture. If placed on a 3rd site, and each site has a separate network connection to that site, then you have an additional path for each site to validate the health of the other. Look at the picture above again and consider the following scenarios.

- What happens if the mediator fails or is unreachable from one or both sites?
  - The two clusters can still communicate with each other over the same link used for replication services.
  - Data is still served with RPO=0 protection
- What happens if Site A fails?
  - Site B will see both of the communication channels go down.
  - Site B will take over data services, but without RPO=0 mirroring
- What happens if Site B fails?
  - Site A will see both of the communication channels go down.
  - Site A will take over data services, but without RPO=0 mirroring

There is one other scenario to consider: Loss of the data replication link. If the replication link between sites is lost, RPO=0 mirroring will obviously be impossible. What should happen then?

This is controlled by the preferred site status. In an SM-as relationship, one of the sites is secondary to the other. This has no effect on normal operations, and all data access is symmetric, but if replication is interrupted then the tie will have to be broken to resume operations. The result is the preferred site will continue operations without mirroring and the secondary site will halt IO processing until replication communication is restored.

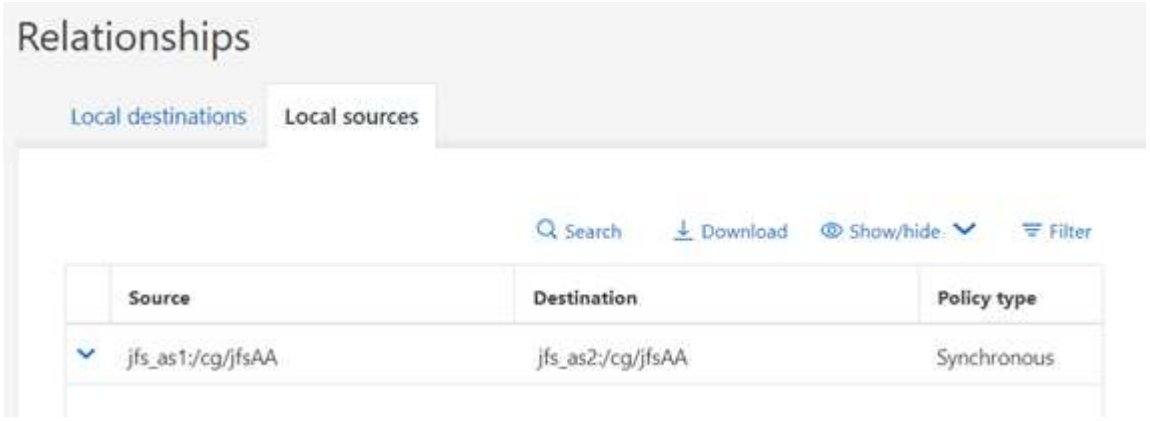
SnapMirror active sync preferred site

SnapMirror active sync behavior is symmetric, with one important exception - preferred site configuration.

SnapMirror active sync will consider one site the "source" and the other the "destination". This implies a one-way replication relationship, but this does not apply to IO behavior. Replication is bidirectional and symmetric and IO response times are the same on either side of the mirror.

The `source` designation is controls the preferred site. If the replication link is lost, the LUN paths on the source copy will continue to serve data while the LUN paths on the destination copy will become unavailable until replication is reestablished and SnapMirror reenters a synchronous state. The paths will then resume serving data.

The sourced/destination configuration can be viewed via SystemManager:



or at the CLI:

```
Cluster2::> snapmirror show -destination-path jfs_as2:/cg/jfsAA

                           Source Path: jfs_as1:/cg/jfsAA
                     Destination Path: jfs_as2:/cg/jfsAA
                    Relationship Type: XDP
                Relationship Group Type: consistencygroup
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: automated-failover-duplex
                      SnapMirror Policy: AutomatedFailOverDuplex
                            Tries Limit: -
                      Throttle (KB/sec): -
                           Mirror State: Snapmirrored
                    Relationship Status: InSync
```

The key is that the source is the SVM on cluster1. As mentioned above, the terms "source" and "destination" don’t describe the flow of replicated data. Both sites can process a write and replicate it to the opposite site. In effect, both clusters are sources and destinations. The effect of designating one cluster as a source simply controls which cluster survives as a read-write storage system if the replication link is lost.

Network topology

Uniform access

Uniform access networking means hosts are able to access paths on both sites (or failure domains within the same site).

An important feature of SM-as is the ability to configure the storage systems to know where the hosts are located. When you map the LUNs to a given host, you can indicate whether or not they are proximal to a given storage system.

Proximity settings

Proximity refers to a per-cluster configuration that indicates a particular host WWN or iSCSI initiator ID belongs to a local host. It is a second, optional step for configuring LUN access.

The first step is the usual igroup configuration. Each LUN must be mapped to an igroup that contains the WWN/iSCSi IDs of the hosts that need access to that LUN. This controls which host has *access* to a LUN.

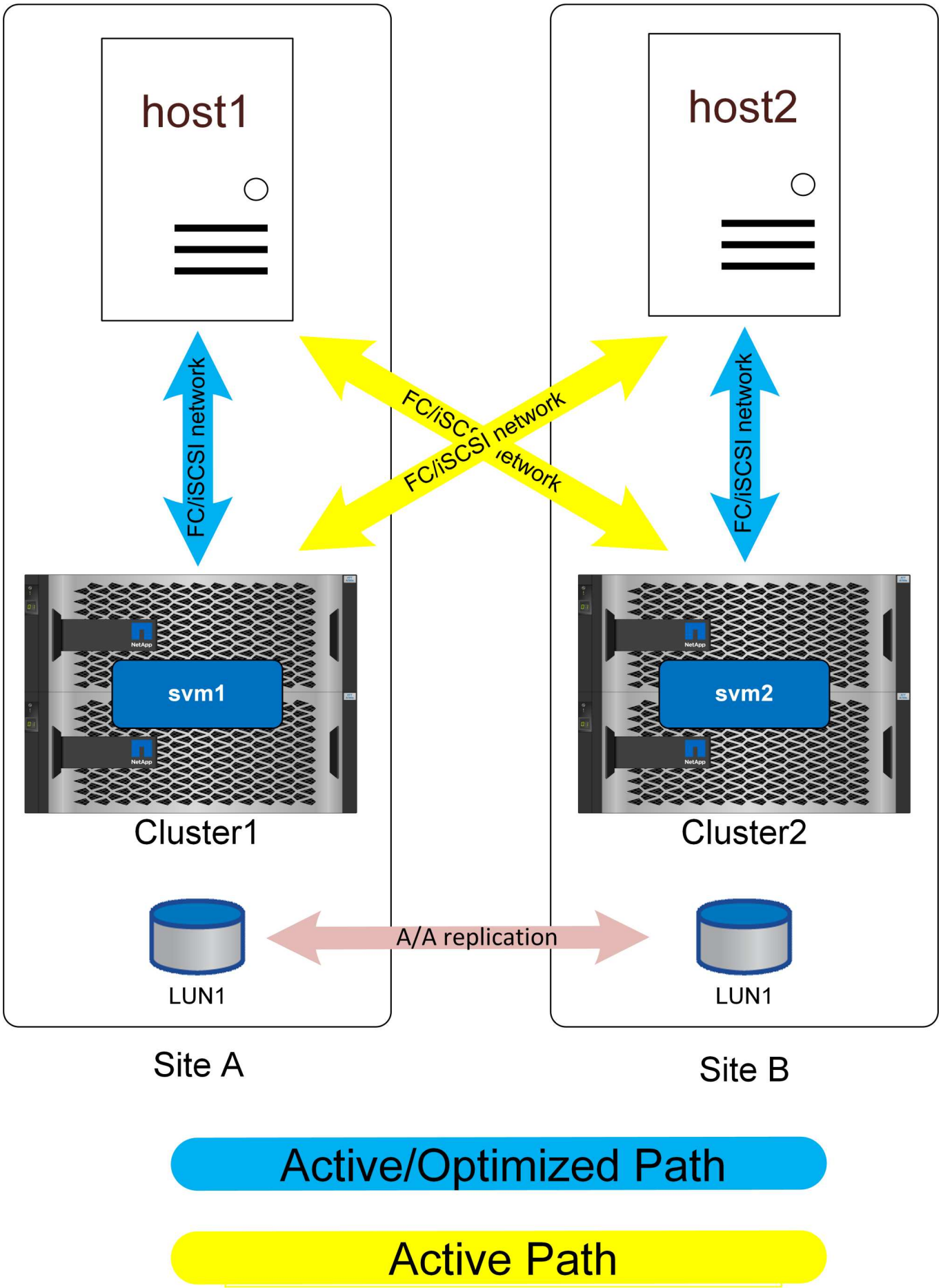
The second, optional step is to configure host proximity. This does not control access, it controls *priority*.

For example, a host at site A might be configured to access a LUN that is protected by SnapMirror active sync, and since the SAN is extended across sites, paths are available to that LUN using storage on site A or storage on site B.

Without proximity settings, that host will use both storage systems equally because both storage systems will advertise active/optimized paths. If the SAN latency and/or bandwidth between sites is limited, this may not be desireable, and you may wish to ensure that during normal operation each host preferentially uses paths to the local storage system. This is configured by adding the host WWN/iSCSI ID to the local cluster as a proximal host. This can be done at the CLI or SystemManager.

AFF

With an AFF system, the paths would appear as shown below when host proximity has been configured.


host1
host2
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
svm1
svm2
Cluster1
Cluster2
A/A replication
LUN1
LUN1
Site A
Site B
Active/Optimized Path
Active Path

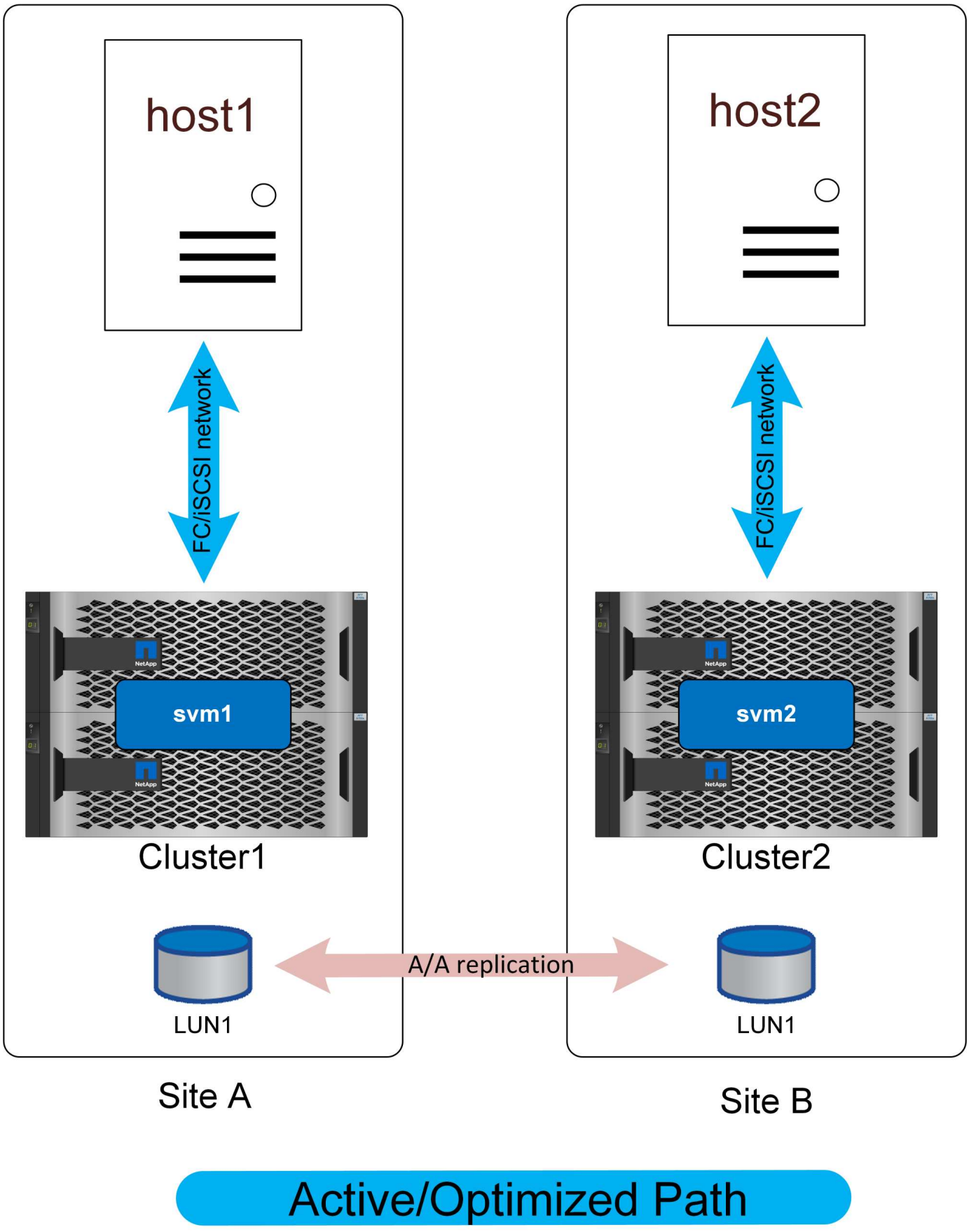
In normal operation, all IO is local IO. Reads and writes are serviced from the local storage array. Write IO will, of course, also need to be replicated by the local controller to the remote system before being acknowledged, but all read IO will be serviced locally and will not incur extra latency by traversing the SAN link between sites.

The only time the nonoptimized paths will be used is when all active/optimized paths are lost. For example, if the entire array on site A lost power, the hosts at site A would still be able to access paths to the array on site B and therefore remain operational, although they would be experiencing higher latency.

There are redundant paths through the local cluster that are not shown on these diagrams for the sake of simplicity. ONTAP storage systems are HA themselves, so a controller failure should not result in site failure. It should merely result in a change in which local paths are used on the affected site.

ASA

NetApp ASA systems offer active-active multipathing across all paths on a cluster. This also applies to SM-as configurations.



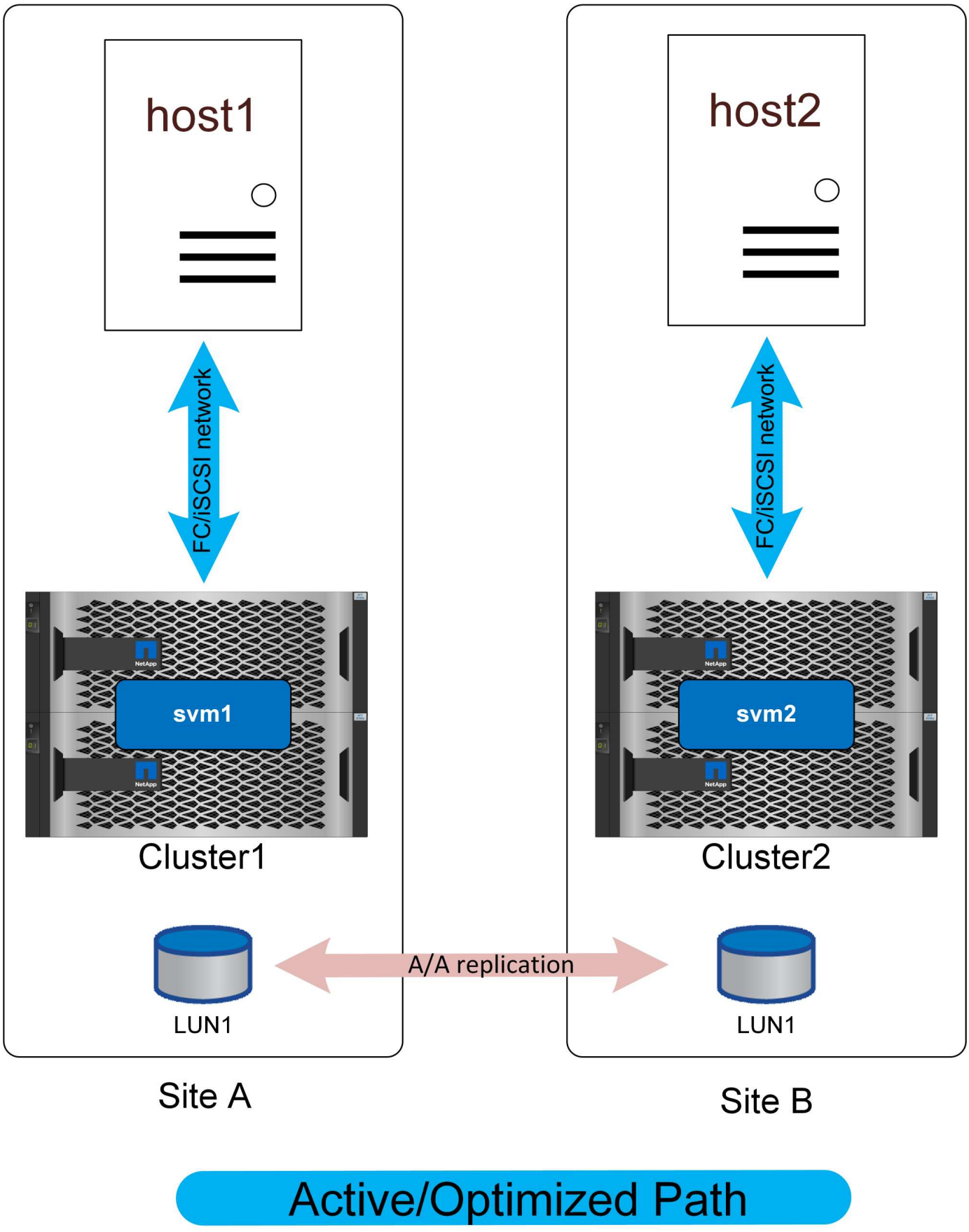
An ASA configuration with non-uniform access would work largely the same as it would with AFF. With uniform access, IO would be crossing the WAN. This may or may not be desirable.

If the two sites were 100 meters apart with fiber connectivity there should be no detectable additional latency crossing the WAN, but if the sites were a long distance apart then read performance would suffer on both sites. In contrast, with AFF those WAN-crossing paths would only be used if there were no local paths available and day-to-day performance would be better because all IO would be local IO. ASA with nonuniform access network would be an option to gain the cost and feature benefits of ASA without incurring a cross-site latency access penalty.

ASA with SM-as in a low-latency configuration offers two interesting benefits. First, it essentially **doubles** the performance for any single host because IO can be serviced by twice as many controllers using twice as many paths. Second, in a single-site environment it offers extreme availability because an entire storage system could be lost without interrupting host access.

Nonuniform access

Nonuniform access networking means each host only has access to ports on the local storage system. The SAN is not extended across sites (or failure domains within the same site).



The primary benefit to this approach is SAN simplicity - you remove the need to stretch a SAN over the network. Some customers don't have sufficiently low-latency connectivity between sites or lack the

infrastructure to tunnel FC SAN traffic over an intersite network.

The disadvantage to nonuniform access is that certain failure scenarios, including loss of the replication link, will result some hosts losing access to storage. Applications that run as single instances, such as a non-clustered database that is inherently only running on a single host at any given mount would fail if local storage connectivity was lost. The data would still be protected, but the database server would no longer have access. It would need to be restarted on a remote site, preferably through an automated process. For example, VMware HA can detect an all-paths-down situation on one server and restart a VM on another server where paths are available.

In contrast, a clustered application such as Oracle RAC can deliver a service that is simultaneously available at two different sites. Losing a site doesn't mean loss of the application service as a whole. Instances are still available and running at the surviving site.

In many cases, the additional latency overhead of an application accessing storage across a site-to-site link would be unacceptable. This means that the improved availability of uniform networking is minimal, since loss of storage on a site would lead to the need to shut down services on that failed site anyway.

There are redundant paths through the local cluster that are not shown on these diagrams for the sake of simplicity. ONTAP storage systems are HA themselves, so a controller failure should not result in site failure. It should merely result in a change in which local paths are used on the affected site.

Oracle Configurations

Overview

The use of SnapMirror active sync does not necessarily add to or change any best practices for operating a database.

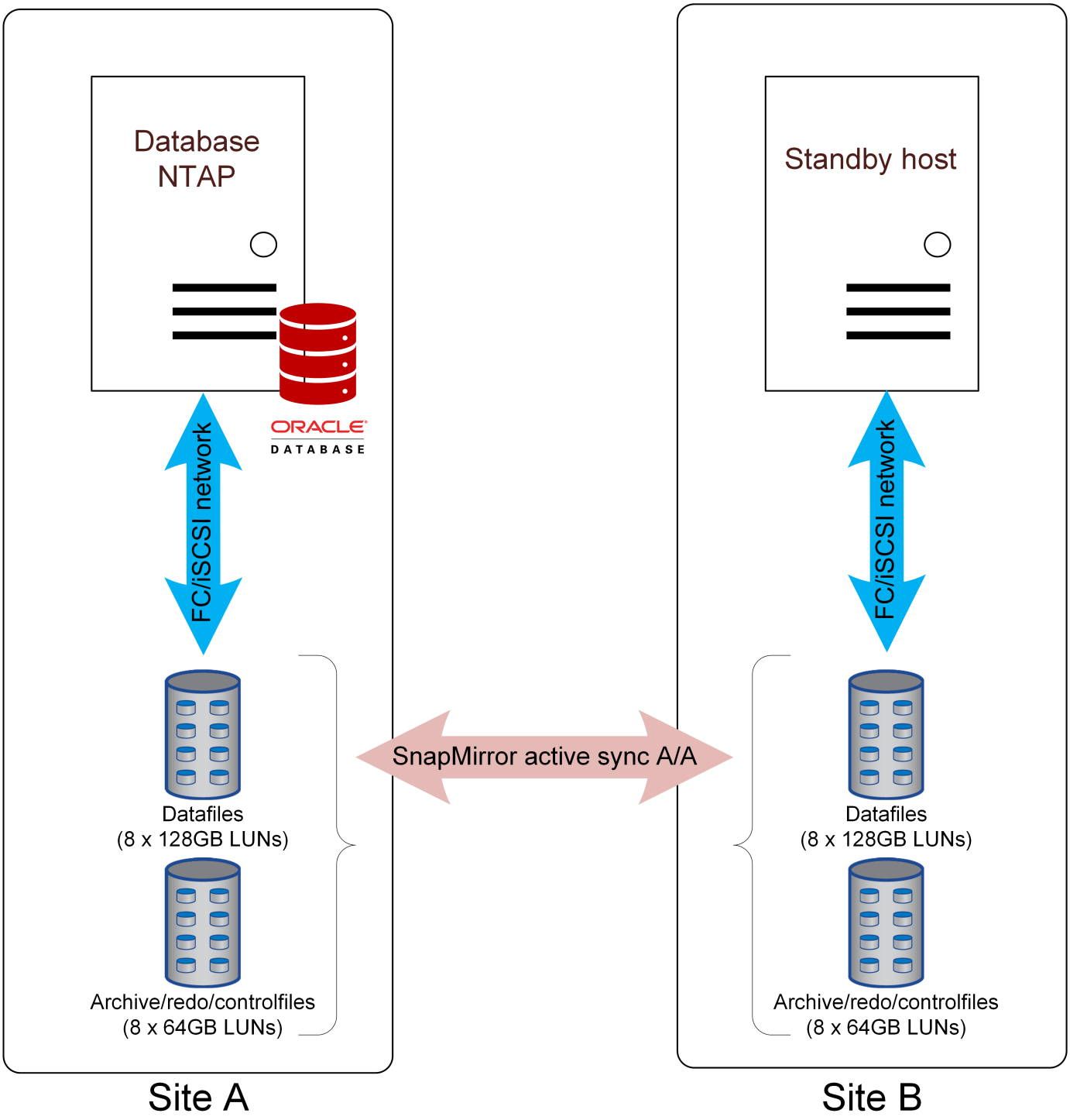
The best architecture depends on the business requirements. For example, if the goal is to have RPO=0 protection against data loss, but the RTO is relaxed, then using Oracle Single Instance databases and replicating the LUNs with SM-as might be sufficient as well as less expensive from an Oracle licensing standpoing. Failure of the remote site would not interrupt operations, and loss of the primary site would result in LUNs at the surviving site that are online and ready to be used.

If the RTO was more strict, basic active-passive automation through scripts or clusterware such as Pacemaker or Ansible would improve failover time. For example, VMware HA could be configured to detect VM failure on the primary site and active the VM on the remote site.

Finally, for extremely rapid failover, Oracle RAC could be deployed across sites. The RTO would essentially be zero because the database would be online and available on both sites at all times.

Oracle Single-Instance

The examples explained below show some of the many options for to deploying Oracle Single Instance databases with SnapMirror active sync replication.



Failover with a preconfigured OS

SnapMirror active sync 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 Pacemaker 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) will bring the databases online at the alternate site. One option is to create standby servers that are preconfigured for the 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. Detect failure of primary site
2. Perform discovery of FC or iSCSI LUNs
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.

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. Detect failure of primary site
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.

Storage failure protection

The diagram above shows the use of nonuniform access, where the SAN is not stretched across sites. This may be simpler to configure, and in some cases may be the only option given the current SAN capabilities, but it also means that failure of the primary storage system would cause a database outage until the application was failed over.

For additional resilience, the solution could be deployed with uniform access. This would allow the applications to continue operating using the paths advertized from the opposite site.

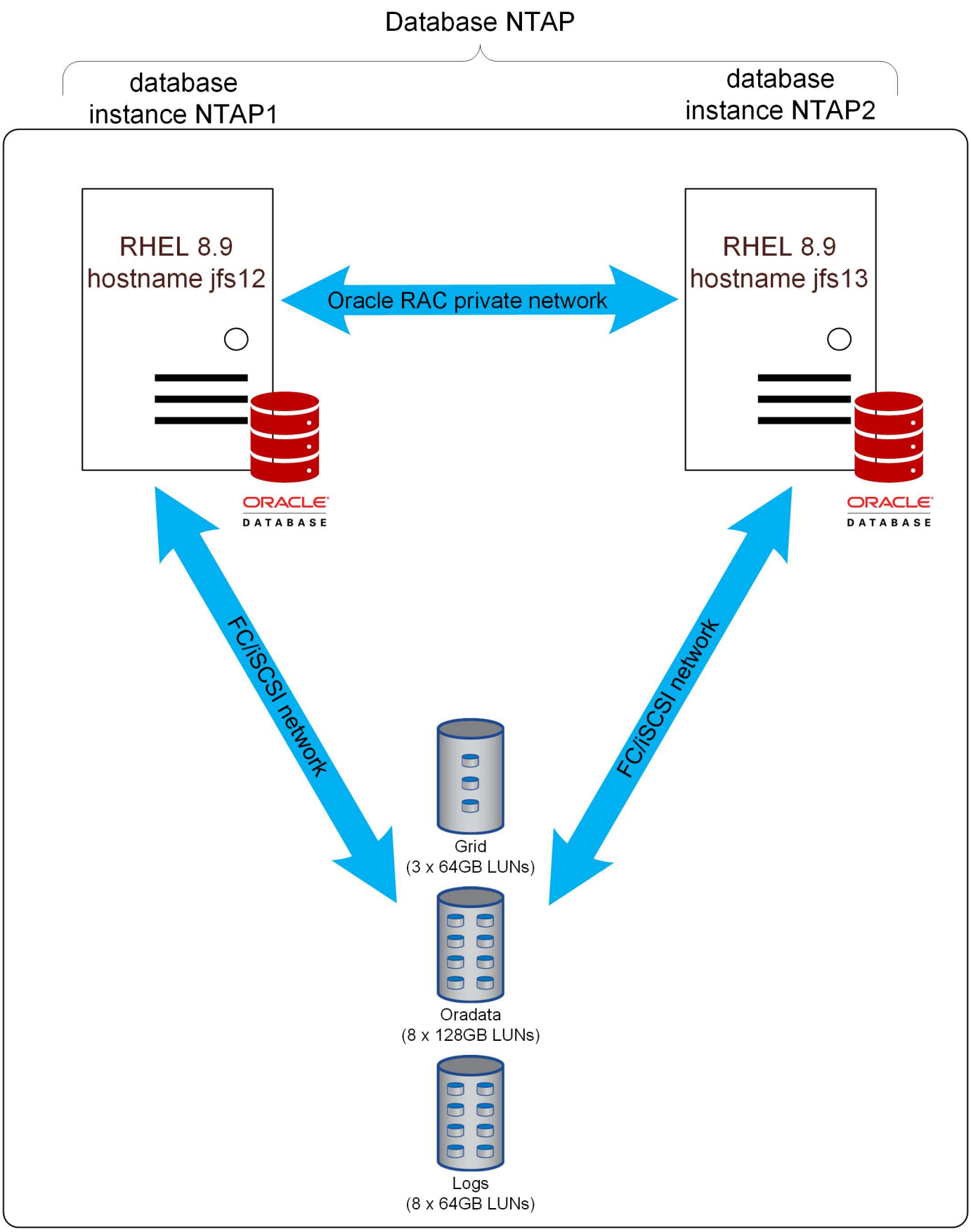
Oracle Extended RAC

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.

Traditional extended RAC clustered relied on ASM mirroring to provide data protection. This approach works, but it also requires a lot of manual configuration steps and imposes overhead on the network infrastructure. In contrast, allowing SnapMirror active sync to take responsibility for data replication dramatically simplifies the solution. Operations such as synchronization, resynchronization after disruptions, failovers, and quorum management are easier, plus the SAN does not need to be distributed across sites which simplifies SAN design and management.

Replication

They key to understanding RAC functionality on SnapMirror active sync is to view storage as a single set of LUNs which are hosted on mirrored storage. For example:



There is no primary copy or mirror copy. Logically, there is only a single copy of each LUN, and that LUN is available on SAN paths that are located on two different storage systems. From a host point of view, there are no storage failovers; instead there are path changes. Various failure events might lead to loss of certain paths

to the LUN while other paths remain online. SnapMirror active sync ensures the same data is available across all operational paths.

Storage configuration

In this example configuration, the ASM disks are configured the same as they would be in any single-site RAC configuration on enterprise storage. Since the storage system provides data protection, ASM external redundancy would be used.

Uniform vs nonuninform access

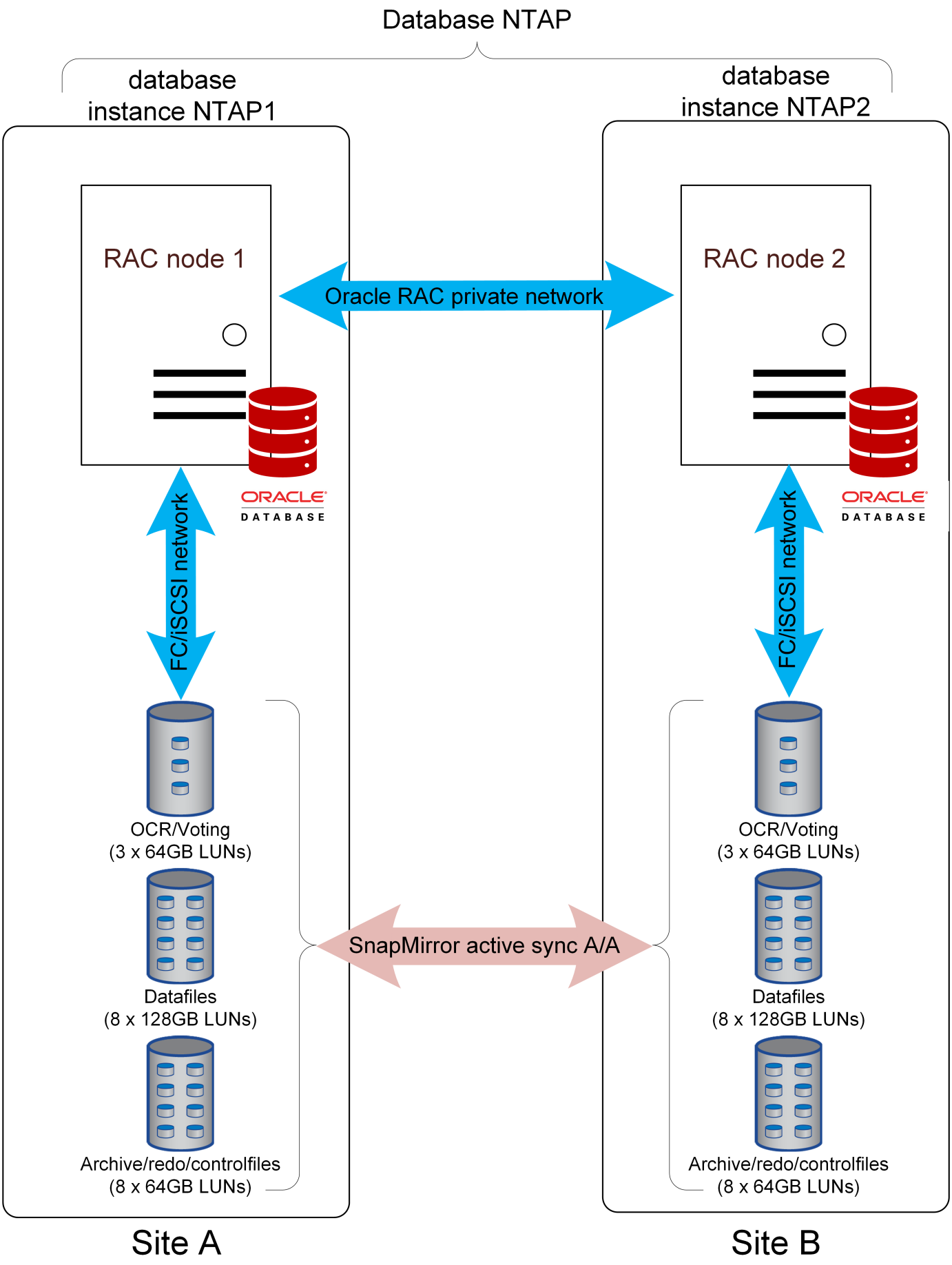
The most important consideration with Oracle RAC on SnapMirror active sync is whether to use uniform or nonuniform access.

Uniform access means each host can see paths on both clusters. Nonuniform access means hosts can only see paths to the local cluster.

Neither option is specifically recommended or discouraged. Some customers have dark fibre readily available to connect sites, others either do not have such connectivity or their SAN infrastructure doesn’t support a long-distance ISL.

Nonuniform access

Nonuniform access is simpler to configure from a SAN perspective.


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
SnapMirror active sync A/A
OCR/Voting (3 x 64GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B

The primary downside of the nonuniform access approach is that loss of site-to-site ONTAP connectivity or loss of a storage system will result in loss of the database instances at one site. This obviously is not desirable, but it may be an acceptable risk in exchange for a simpler SAN configuration.

Uniform access

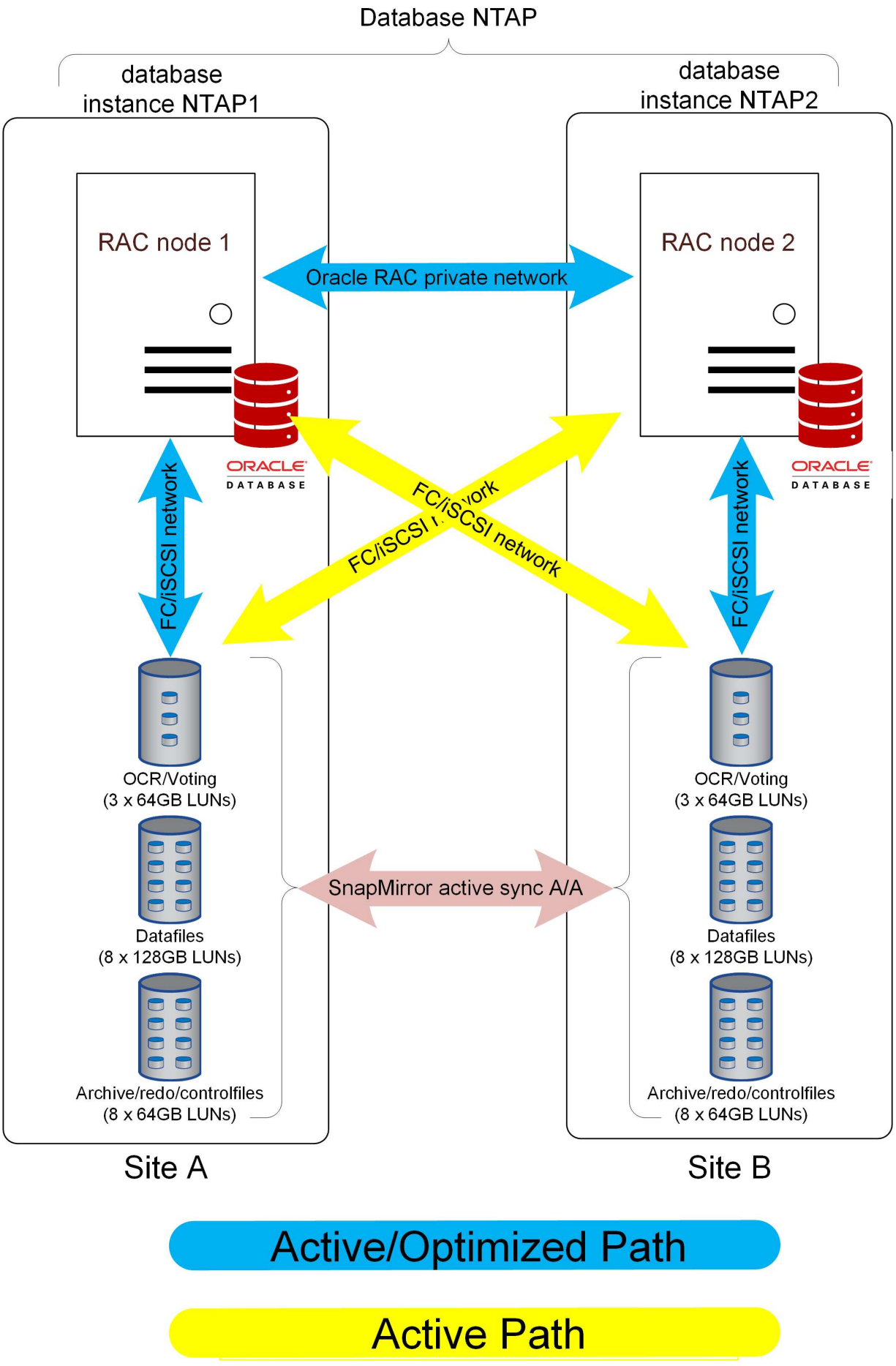
Uniform access requires extending the SAN across sites. The primary benefit is that loss of a storage system will not result in loss of a database instance. Instead, it would result in a multipathing change in which paths are currently in use.

There are several ways to configure nonuniform access.

> In the diagrams below, there are also active but nonoptimized paths present that would be used during simple controller failures, but those paths are not shown in the interest of simplifying the diagrams.

AFF with proximity settings

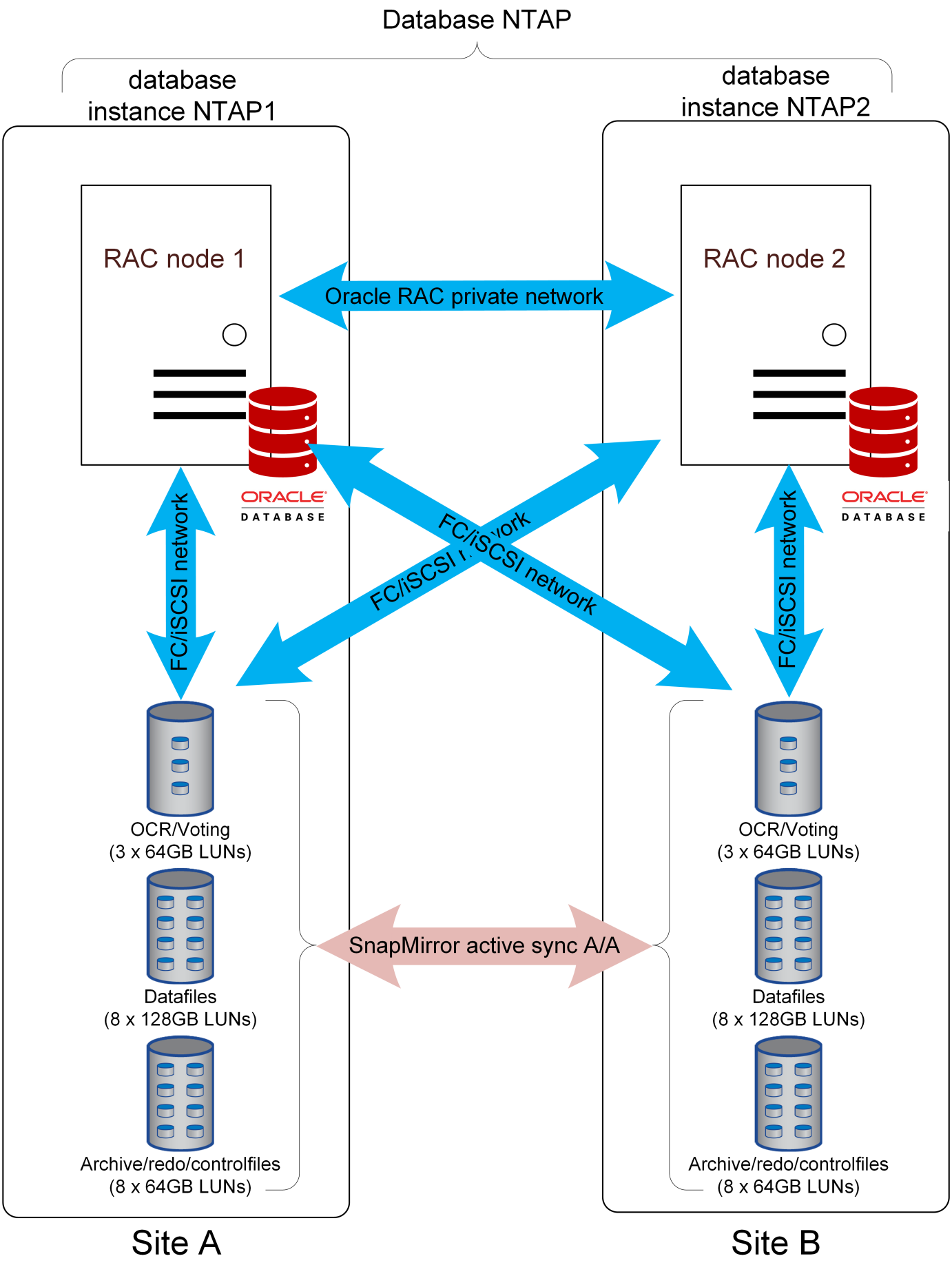
If there is significant latency between sites, then AFF systems can be configured with host proximity settings. This allows each storage system to be aware of which hosts are local and which are remote and assign path priorities appropriately.


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
OCR/Voting (3 x 64GB LUNs)
SnapMirror active sync A/A
Datafiles (8 x 128GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B
Active/Optimized Path
Active Path

In normal operation, each Oracle instance would preferentially use the local active/optimized paths. The result is that all reads would be serviced by the local copy of the blocks. This yields the lowest possible latency. Write IO is similarly sent down paths to the local controller. The IO must still be replicated before being acknowledged and therefor would still incur the additional latency of crossing the site-to-site network, but this cannot be avoided in a synchronous replication solution.

ASA / AFF without proximity settings

If there is no significant latency between sites, then AFF systems can be configured without host proximity settings, or ASA can be used.


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
OCR/Voting (3 x 64GB LUNs)
SnapMirror active sync A/A
Datafiles (8 x 128GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B

Each host will be able to use all operational paths on both storage systems. This potentially improves performance significantly by allowing each host to draw upon the performance potential of two clusters, not just one.

With ASA, not only would all paths to both clusters be considered active and optimized, but the paths on partner controllers would also be active. The result would be all-active SAN paths on the entire cluster, all the time.

ASA systems may also be used in a nonuniform access configuration. Since no cross-site paths exist, there would be no impact on performance resulting from IO crossing the ISL.

RAC tiebreaker

While extended RAC using SnapMirror active sync is a symmetric architecture with respect to IO, there is one exception that is connected to split-brain management.

What happens if the replication link is lost and neither site has quorum? What should happen? This question applies to both the Oracle RAC and the ONTAP behavior. If changes cannot be replicated across sites, and you want to resume operations, one of the sites will have to survive and the other site will have to become unavailable.

The ONTAP Mediator addresses this requirement at the ONTAP layer. There are multiple options for RAC tiebreaking.

Oracle tiebreakers

The best method to manage split-brain Oracle RAC risks is to use an odd number of RAC nodes, preferably by use of a 3rd site tiebreaker. If a 3rd site is unavailable, the tiebreaker instance could be placed on one site of the two sites, effectively designating it a preferred survivor site.

Oracle and css_critical

With an even number of nodes, the default Oracle RAC behavior is that one of the nodes in the cluster will be deemed more important than the other nodes. The site with that higher priority node will survive site isolation while the nodes on the other site will evict. The prioritization is based on multiple factors, but you can also control this behavior using the `css_critical` setting.

In the example architecture, the hostnames for the RAC nodes are jfs12 and jfs13. The current settings for `css_critical` are as follows:

```
[root@jfs12 ~]# /grid/bin/crsctl get server css_critical
CRS-5092: Current value of the server attribute CSS_CRITICAL is no.

[root@jfs13 trace]# /grid/bin/crsctl get server css_critical
CRS-5092: Current value of the server attribute CSS_CRITICAL is no.
```

If you want the site with jfs12 to be the preferred site, change this value to yes on a site A node and restart services.

```
[root@jfs12 ~]# /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.

[root@jfs12 ~]# /grid/bin/crsctl stop crs
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'jfs12'
CRS-2673: Attempting to stop 'ora.crsd' on 'jfs12'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
server 'jfs12'
CRS-2673: Attempting to stop 'ora.ntap.ntappdb1.pdb' on 'jfs12'
…
CRS-2673: Attempting to stop 'ora.gipcd' on 'jfs12'
CRS-2677: Stop of 'ora.gipcd' on 'jfs12' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'jfs12' has completed
CRS-4133: Oracle High Availability Services has been stopped.

[root@jfs12 ~]# /grid/bin/crsctl start crs
CRS-4123: Oracle High Availability Services has been started.
```

Failure scenarios

Overview

Planning a complete SnapMirror active sync application architecture requires understanding how SM-as will respond in various planned and unplanned failover scenarios.

For the following examples, assume that site A is configured as the preferred site.

Loss of replication connectivity

If SM-as replication is interrupted, write IO cannot be completed because it would be impossible for a cluster to replicate changes to the opposite site.

Site A (Preferred site)

The result of replication link failure on the preferred site will be an approximate 15 second pause in write IO processing as ONTAP retries replicated write operations before it determines that the replication link is genuinely unreachable. After the 15 seconds elapses, the site A system resumes read and write IO processing. The SAN paths will not change, and the LUNs will remain online.

Site B

Since site B is not the SnapMirror active sync preferred site, its LUN paths will become unavailable after about 15 seconds.

Storage system failure

The result of a storage system failure is nearly identical to the result of losing the replication link. The surviving site should experience a roughly 15 second IO pause. Once that 15 second period elapses, IO will resume on that site as usual.

Loss of the mediator

The mediator service does not directly control storage operations. It functions as an alternate control path between clusters. It exists primarily to automate failover without the risk of a split-brain scenario. In normal operation, each cluster is replicating changes to its partner, and each cluster therefore can verify that the partner cluster is online and serving data. If the replication link failed, replication would cease.

The reason a mediator is required for safe automated failover is because it would otherwise be impossible for a storage cluster to be able to determine whether loss of bidirectional communication was the result of a network outage or actual storage failure.

The mediator provides an alternate path for each cluster to verify the health of its partner. The scenarios are as follows:

- If a cluster can contact its partner directly, replication services are operational. No action required.
- If a preferred site cannot contact its partner directly or via the mediator, it will assume the partner is either actually unavailable or was isolated and has taken its LUN paths offline. The preferred site will then proceed to release the RPO=0 state and continue processing both read and write IO.
- If a non-preferred site cannot contact its partner directly, but can contact it via the mediator, it will take its paths offline and await the return of the replication connection.
- If a non-preferred site cannot contact its partner directly or via an operational mediator, it will assume the partner is either actually unavailable or was isolated and has taken its LUN paths offline. The non-preferred site will then proceed to release the RPO=0 state and continue processing both read and write IO. It will assume the role of the replication source and will become the new preferred site.

If the mediator is wholly unavailable:

- Failure of replication services for any reason, including failure of the nonpreferred site or storage system, will result in the preferred site releasing the RPO=0 state and resuming read and write IO processing. The non-preferred site will take its paths offline.
- Failure of the preferred site will result in an outage because the non-preferred site will be unable to verify that the opposite site is truly offline and therefore it would not be safe for the nonpreferred site to resume services.

Restoring services

After a failure is resolved, such as restoring site-to-site connectivity or powering on a failed system, the SnapMirror active sync endpoints will automatically detect the presence of a faulty replication relationship and bring it back to an RPO=0 state. Once synchronous replication is reestablished, the failed paths will come online again.

In many cases, clustered applications will automatically detect the return of failed paths, and those applications will also come back online. In other cases, a host-level SAN scan may be required, or applications may need to be brought back online manually. It depends on the application and how it is configured, and in general such tasks can be easily automated. ONTAP itself is self-healing and should not require any user intervention to resume RPO=0 storage operations.

Manual failover

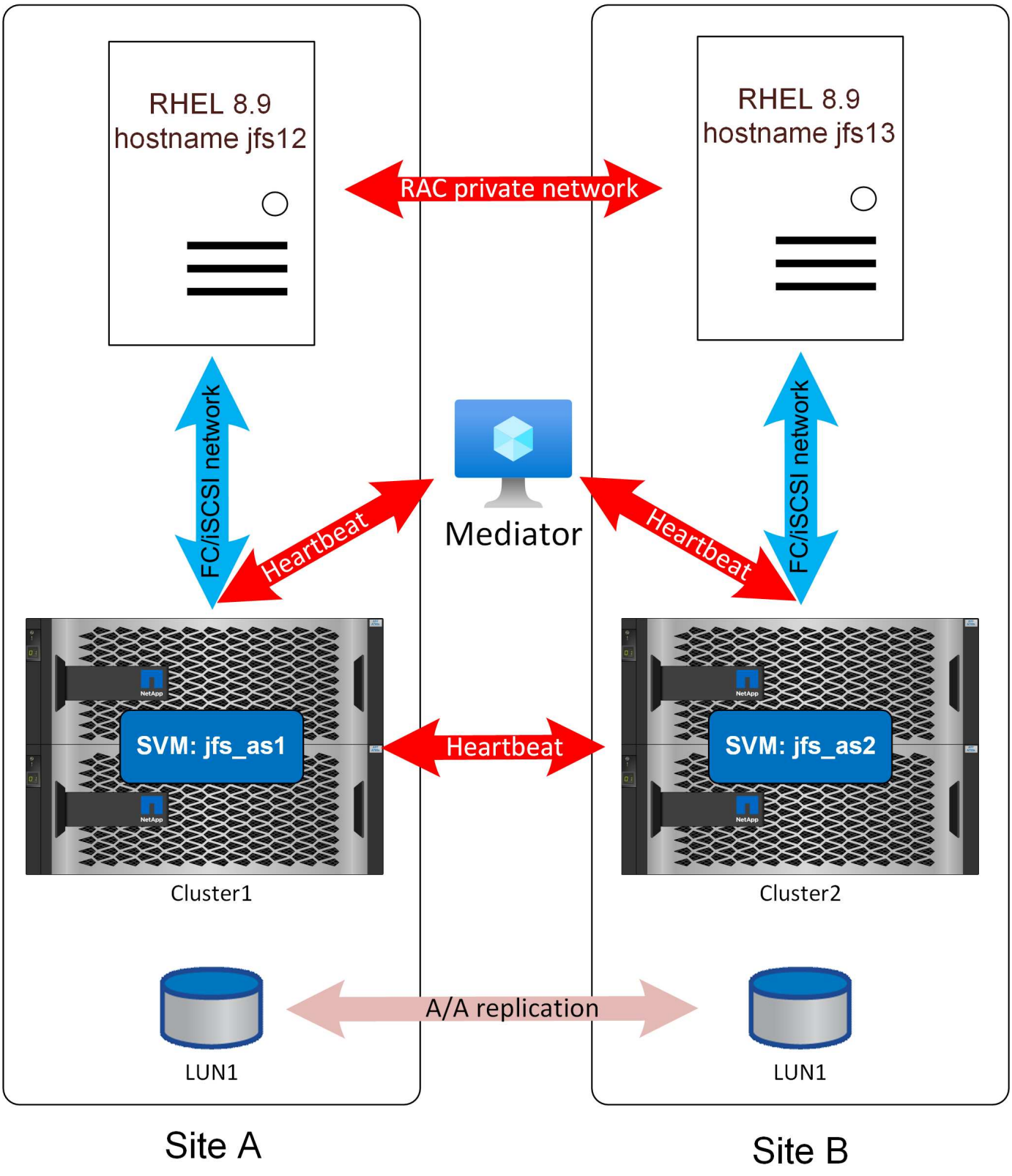
Changing the preferred site requires a simple operation. IO will pause for a second or two as authority over replication behavior switches between clusters, but IO is otherwise unaffected.

Sample architecture

The detailed failure examples shown in this sections are based on the architecture shown below.

> This is only one of many options for Oracle databases on SnapMirror active sync. This design was chosen because it illustrates some of the more complicated scenarios.

In this design, assume that site A is set at the preferred site.



RAC interconnect failure

Loss of the Oracle RAC replication link will produce a similar result to loss of SnapMirror connectivity, except the timeouts will be shorter by default. Under default settings, an Oracle RAC node will wait 200 seconds after loss of storage connectivity before evicting,

but it will only wait 30 seconds after loss of the RAC network heartbeat.

The CRS messages are similar to those shown below. You can see the 30 second timeout lapse. Since css_critical was set on jfs12, located on site A, that will be the site to survive and jfs13 on site B will be evicted.

```
2024-09-12 10:56:44.047 [ONMD(3528)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 6.980 seconds
2024-09-12 10:56:48.048 [ONMD(3528)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 2.980 seconds
2024-09-12 10:56:51.031 [ONMD(3528)]CRS-1607: Node jfs13 is being evicted
in cluster incarnation 621599354; details at (:CSSNM00007:) in
/gridbase/diag/crs/jfs12/crs/trace/onmd.trc.
2024-09-12 10:56:52.390 [CRSD(6668)]CRS-7503: The Oracle Grid
Infrastructure process 'crsd' observed communication issues between node
'jfs12' and node 'jfs13', interface list of local node 'jfs12' is
'192.168.30.1:33194;', interface list of remote node 'jfs13' is
'192.168.30.2:33621;'.
2024-09-12 10:56:55.683 [ONMD(3528)]CRS-1601: CSSD Reconfiguration
complete. Active nodes are jfs12 .
2024-09-12 10:56:55.722 [CRSD(6668)]CRS-5504: Node down event reported for
node 'jfs13'.
2024-09-12 10:56:57.222 [CRSD(6668)]CRS-2773: Server 'jfs13' has been
removed from pool 'Generic'.
2024-09-12 10:56:57.224 [CRSD(6668)]CRS-2773: Server 'jfs13' has been
removed from pool 'ora.NTAP'.
```

SnapMirror communication failure

If the SnapMirror active sync replication link, write IO cannot be completed because it would be impossible for a cluster to replicate changes to the opposite site.

Site A

The result on site A of a replication link failure will be an approximately 15 second pause in write IO processing as ONTAP attempts to replicate writes before it determines that the replication link is genuinely inoperable. After the 15 seconds elapses, the ONTAP cluster on site A resumes read and write IO processing. The SAN paths will not change, and the LUNs will remain online.

Site B

Since site B is not the SnapMirror active sync preferred site, its LUN paths will become unavailable after about 15 seconds.

The replication link was cut at the timestamp 15:19:44. The first warning from Oracle RAC arrives 100 seconds later as the 200 second timeout (controlled by the Oracle RAC parameter disktimeout) approaches.

```
2024-09-10 15:21:24.702 [ONMD(2792)]CRS-1615: No I/O has completed after
50% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 99340 milliseconds.
2024-09-10 15:22:14.706 [ONMD(2792)]CRS-1614: No I/O has completed after
75% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 49330 milliseconds.
2024-09-10 15:22:44.708 [ONMD(2792)]CRS-1613: No I/O has completed after
90% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 19330 milliseconds.
2024-09-10 15:23:04.710 [ONMD(2792)]CRS-1604: CSSD voting file is offline:
/dev/mapper/grid2; details at (:CSSNM00058:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc.
2024-09-10 15:23:04.710 [ONMD(2792)]CRS-1606: The number of voting files
available, 0, is less than the minimum number of voting files required, 1,
resulting in CSSD termination to ensure data integrity; details at
(:CSSNM00018:) in /gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-10 15:23:04.716 [ONMD(2792)]CRS-1699: The CSS daemon is
terminating due to a fatal error from thread:
clssnmvDiskPingMonitorThread; Details at (:CSSSC00012:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-10 15:23:04.731 [OCSSD(2794)]CRS-1652: Starting clean up of CRSD
resources.
```

Once the 200 second voting disk timeout has been reached, this Oracle RAC node will evict itself from the cluster and reboot.

Total network interconnectivity failure

If the replication link between sites is completely lost, both SnapMirror active sync and Oracle RAC connectivity will be interrupted.

Oracle RAC split-brain detection has a dependency on the Oracle RAC storage heartbeat. If loss of site-to-site connectivity results in simultaneous loss of both the RAC network heartbeat and storage replication services, the result is the RAC sites will not be able to communicate cross-site via either the RAC interconnect or the RAC voting disks. The result in an even-numbed set of nodes may be eviction of both sites under default settings. The exact behavior will depend on the sequence of events and the timing of the RAC network and disk heartbeat polls.

The risk of a 2-site outage can be addressed in two ways. First, a tiebreaker configuration can be used.

If a 3rd site is not available, this risk can be addressed by adjusting the misscount parameter on the RAC cluster. Under the defaults, the RAC network heartbeat timeout is 30 seconds. This normally is used by RAC to identify failed RAC nodes and remove them from the cluster. It also has a connection to the voting disk heartbeat.

If, for example, the conduit carrying intersite traffic for both Oracle RAC and storage replication services is cut by a backhoe, the 30 second misscount countdown will begin. If the RAC preferred site node cannot reestablish contact with the opposite site within 30 seconds, and it also cannot use the voting disks to confirm the opposite site is down within that same 30 second window, then the preferred site nodes will also evict. The

result is a full database outage.

Depending on when the misscount polling occurs, 30 seconds may not be enough time for SnapMirror active sync to time out and allow storage on the preferred site to resume services before the 30 second window expires. This 30 second window can be increased.

```
[root@jfs12 ~]# /grid/bin/crsctl set css misscount 100
CRS-4684: Successful set of parameter misscount to 100 for Cluster
Synchronization Services.
```

This value allows the storage system on the preferred site to resume operations before the miscount timeout expires. The result will then be eviction only of the nodes at the site where the LUN paths were removed. Example below:

```
2024-09-12 09:50:59.352 [ONMD(681360)]CRS-1612: Network communication with
node jfs13 (2) has been missing for 50% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 49.570 seconds
2024-09-12 09:51:10.082 [CRSD(682669)]CRS-7503: The Oracle Grid
Infrastructure process 'crsd' observed communication issues between node
'jfs12' and node 'jfs13', interface list of local node 'jfs12' is
'192.168.30.1:46039;', interface list of remote node 'jfs13' is
'192.168.30.2:42037;'.
2024-09-12 09:51:24.356 [ONMD(681360)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 24.560 seconds
2024-09-12 09:51:39.359 [ONMD(681360)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 9.560 seconds
2024-09-12 09:51:47.527 [OHASD(680884)]CRS-8011: reboot advisory message
from host: jfs13, component: cssagent, with time stamp: L-2024-09-12-
09:51:47.451
2024-09-12 09:51:47.527 [OHASD(680884)]CRS-8013: reboot advisory message
text: oracssdagent is about to reboot this node due to unknown reason as
it did not receive local heartbeats for 10470 ms amount of time
2024-09-12 09:51:48.925 [ONMD(681360)]CRS-1632: Node jfs13 is being
removed from the cluster in cluster incarnation 621596607
```

Oracle Support strongly discourages altering with the misscount or disktimeout parameters to solve configuration problems. Changing these parameters can, however, be warranted and unavoidable in many cases, including SAN booting, virtualized, and storage replication configurations. If, for example, you had stability problems with a SAN or IP network that was resulting in RAC evictions you should fix the underlying problem and not charge the values of the misscount or disktimeout. Changing timeouts to address configuration errors is masking a problem, not solving a problem. Changing these parameters to properly configure a RAC environment based on design aspects of the underlying infrastructure is different and is consistent with Oracle support statements. With SAN booting, it is common to adjust misscount all the way up to 200 to match disktimeout. See this link for additional information.

Site failure

The result of a storage system or site failure is nearly identical to the result of losing the replication link. The surviving site should experience a roughly 15 second IO pause on writes. Once that 15 second period elapses, IO will resume on that site as usual.

If only the storage system was affected, the Oracle RAC node on the failed site will lose storage services and enter the same 200 second disktimeout countdown before eviction and subsequent reboot.

```
2024-09-11 13:44:38.613 [ONMD(3629)]CRS-1615: No I/O has completed after
50% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 99750 milliseconds.
2024-09-11 13:44:51.202 [ORAAGENT(5437)]CRS-5011: Check of resource "NTAP"
failed: details at "(:CLSN00007:)" in
"/gridbase/diag/crs/jfs13/crs/trace/crsd_oraagent_oracle.trc"
2024-09-11 13:44:51.798 [ORAAGENT(75914)]CRS-8500: Oracle Clusterware
ORAAGENT process is starting with operating system process ID 75914
2024-09-11 13:45:28.626 [ONMD(3629)]CRS-1614: No I/O has completed after
75% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 49730 milliseconds.
2024-09-11 13:45:33.339 [ORAAGENT(76328)]CRS-8500: Oracle Clusterware
ORAAGENT process is starting with operating system process ID 76328
2024-09-11 13:45:58.629 [ONMD(3629)]CRS-1613: No I/O has completed after
90% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 19730 milliseconds.
2024-09-11 13:46:18.630 [ONMD(3629)]CRS-1604: CSSD voting file is offline:
/dev/mapper/grid2; details at (:CSSNM00058:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc.
2024-09-11 13:46:18.631 [ONMD(3629)]CRS-1606: The number of voting files
available, 0, is less than the minimum number of voting files required, 1,
resulting in CSSD termination to ensure data integrity; details at
(:CSSNM00018:) in /gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-11 13:46:18.638 [ONMD(3629)]CRS-1699: The CSS daemon is
terminating due to a fatal error from thread:
clssnmvDiskPingMonitorThread; Details at (:CSSSC00012:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-11 13:46:18.651 [OCSSD(3631)]CRS-1652: Starting clean up of CRSD
resources.
```

The SAN path state on the RAC node that has lost storage services looks like this:

```
oradata7 (3600a0980383041334a3f55676c697347) dm-20 NETAPP,LUN C-Mode
size=128G features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
alua' wp=rw
|-+- policy='service-time 0' prio=0 status=enabled
| `- 34:0:0:18 sdam 66:96  failed faulty running
`-+- policy='service-time 0' prio=0 status=enabled
  `- 33:0:0:18 sdaj 66:48  failed faulty running
```

The linux host detected the loss of the paths much quicker than 200 seconds, but from a database perspective the client connections to the host on the failed site will still be frozen for 200 seconds under the default Oracle RAC settings. Full database operations will only resume after the eviction is completed.

Meanwhile, the Oracle RAC node on the opposite site will record the loss of the other RAC node. It otherwise continues to operate as usual.

```
2024-09-11 13:46:34.152 [ONMD(3547)]CRS-1612: Network communication with
node jfs13 (2) has been missing for 50% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 14.020 seconds
2024-09-11 13:46:41.154 [ONMD(3547)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 7.010 seconds
2024-09-11 13:46:46.155 [ONMD(3547)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 2.010 seconds
2024-09-11 13:46:46.470 [OHASD(1705)]CRS-8011: reboot advisory message
from host: jfs13, component: cssmonit, with time stamp: L-2024-09-11-
13:46:46.404
2024-09-11 13:46:46.471 [OHASD(1705)]CRS-8013: reboot advisory message
text: At this point node has lost voting file majority access and
oracssdmonitor is rebooting the node due to unknown reason as it did not
receive local hearbeats for 28180 ms amount of time
2024-09-11 13:46:48.173 [ONMD(3547)]CRS-1632: Node jfs13 is being removed
from the cluster in cluster incarnation 621516934
```

Mediator failure

The mediator service does not directly control storage operations. It functions as an alternate control path between clusters. It exists primarily to automate failover without the risk of a split-brain scenario.

In normal operation, each cluster is replicating changes to its partner, and each cluster therefore can verify that the partner cluster is online and serving data. If the replication link failed, replication would cease.

The reason a mediator is required for safe automated operations is because it would otherwise be impossible for a storage clusters to be able to determine whether loss of bidirectional communication was the result of a network outage or actual storage failure.

The mediator provides an alternate path for each cluster to verify the health of its partner. The scenarios are as follows:

- If a cluster can contact its partner directly, replication services are operational. No action required.
- If a preferred site cannot contact its partner directly or via the mediator, it will assume the partner is either actually unavailable or was isolated and has taken its LUN paths offline. The preferred site will then proceed to release the RPO=0 state and continue processing both read and write IO.
- If a non-preferred site cannot contact its partner directly, but can contact it via the mediator, it will take its paths offline and await the return of the replication connection.
- If a non-preferred site cannot contact its partner directly or via an operational mediator, it will assume the partner is either actually unavailable or was isolated and has taken its LUN paths offline. The non-preferred site will then proceed to release the RPO=0 state and continue processing both read and write IO. It will assume the role of the replication source and will become the new preferred site.

If the mediator is wholly unavailable:

- Failure of replication services for any reason will result in the preferred site releasing the RPO=0 state and resuming read and write IO processing. The non-preferred site will take its paths offline.
- Failure of the preferred site will result in an outage because the non-preferred site will be unable to verify that the opposite site is truly offline and therefore it would not be safe for the nonpreferred site to resume services.

Service restoration

SnapMirror is self-healing. SnapMirror active sync will automatically detect the presence of a faulty replication relationship and bring it back to an RPO=0 state. Once synchronous replication is reestablished, the paths will come online again.

In many cases, clustered applications will automatically detect the return of failed paths, and those applications will also come back online. In other cases, a host-level SAN scan may be required, or applications may need to be brought back online manually.

It depends on the application and how it’s configured, and in general such tasks can be easily automated. SnapMirror active sync itself is self-fixing and should not require any user intervention to resume RPO=0 storage operations once power and connectivity is restored.

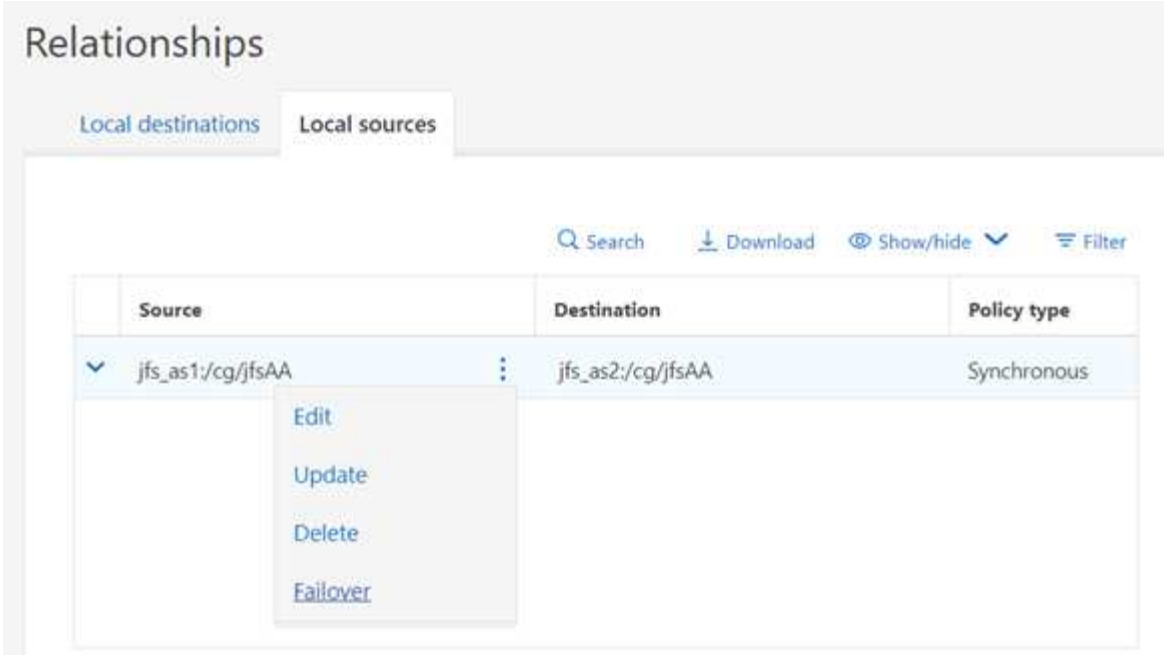
Manual failover

The term "failover" does not refer to the direction of replication with SnapMirror active sync because it is a bidirectional replication technology. Instead, 'failover' refers to which storage system will be the preferred site in the event of failure.

For example, you may want to perform a failover to change the preferred site before you shut down a site for maintenance, or before performing a DR test.

Changing the preferred site requires a simple operation. IO will pause for a second or two as authority over replication behavior switches between clusters, but IO is otherwise unaffected.

GUI example:



Example of changing it back via the CLI:

```
Cluster2::> snapmirror failover start -destination-path jfs_as2:/cg/jfsAA
[Job 9575] Job is queued: SnapMirror failover for destination
"jfs_as2:/cg/jfsAA                      ".

Cluster2::> snapmirror failover show

Source    Destination                                          Error
Path      Path        Type     Status    start-time end-time   Reason
--------- ----------  -------- --------- ---------- ---------- ----------
jfs_as1:/cg/jfsAA
         jfs_as2:/cg/jfsAA
                     planned  completed 9/11/2024  9/11/2024
                                        09:29:22   09:29:32

The new destination path can be verified as follows:


Cluster1::> snapmirror show -destination-path jfs_as1:/cg/jfsAA

                           Source Path: jfs_as2:/cg/jfsAA
                        Destination Path: jfs_as1:/cg/jfsAA
                       Relationship Type: XDP
                 Relationship Group Type: consistencygroup
                  SnapMirror Policy Type: automated-failover-duplex
                       SnapMirror Policy: AutomatedFailOverDuplex
                             Tries Limit: -
                           Mirror State: Snapmirrored
                    Relationship Status: InSync
```

Oracle database migration

Overview

Leveraging the capabilities of a new storage platform has one unavoidable requirement; data must be placed on the new storage system. ONTAP makes the migration process simple, including both ONTAP to ONTAP migrations and upgrades, foreign LUN imports, and procedures for using the host operating system or Oracle database software directly.

This documentation replaces previously published technical report *TR-4534: Migration of Oracle Databases to NetApp Storage Systems*

In the case of a new database project, this is not a concern because the database and application environments are constructed in place. Migration, however, poses special challenges regarding business disruption, the time required for the completion of migration, needed skill sets, and risk minimization.

Scripts

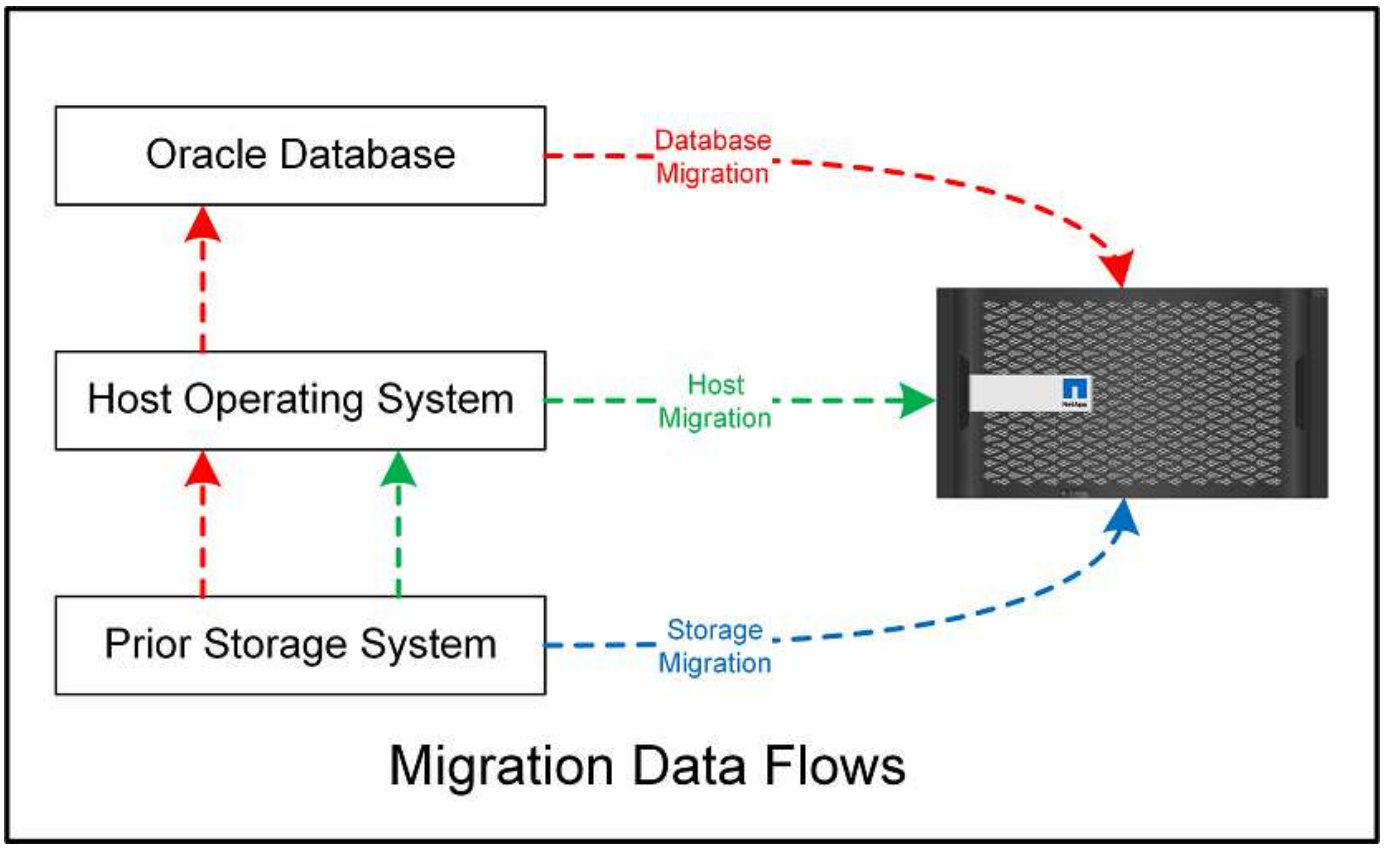
Sample scripts are provided in this documentation. These scripts provide sample methods of automating various aspects of migration to reduce the chance of user errors. The scripts can reduce the overall demands on the IT staff responsible for a migration and they can speed up the overall process. These scripts are all drawn from actual migration projects performed by NetApp Professional Services and NetApp partners. Examples of their use are shown throughout this documentation.

Migration planning

Oracle data migration can occur at one of three levels: the database, the host, or the storage array.

The differences lie in which component of the overall solution is responsible for moving data: the database, the host operating system, or the storage system.

The figure below shows an example of the migration levels and the flow of data. In the case of database-level migration, the data is moved from the original storage system through the host and database layers into the new environment. Host-level migration is similar, but data does not pass through the application layer and is instead written to the new location by using host processes. Finally, with storage-level migration, an array such as a NetApp FAS system is responsible for data movement.



A database-level migration generally refers to the use of Oracle log shipping through a standby database to complete a migration at the Oracle layer. Host-level migrations are performed by using the native capability of the host operating system configuration. This configuration includes file copy operations using commands such as cp, tar, and Oracle Recovery Manager (RMAN) or using a logical volume manager (LVM) to relocate the underlying bytes of a file system. Oracle Automatic Storage Management (ASM) is categorized as a host- level capability because it runs below the level of the database application. ASM takes the place of the usual logical volume manager on a host. Finally, data can be migrated at the storage- array level, which means beneath the

level of the operating system.

Planning considerations

The best option for migration depends on a combination of factors, including the scale of the environment to be migrated, the need to avoid downtime, and the overall effort required to perform the migration. Large databases obviously require more time and effort for migration, but the complexity of such a migration is minimal. Small databases can be migrated quickly, but, if there are thousands to be migrated, the scale of the effort can create complications. Finally, the larger the database, the more likely it is to be business-critical, which gives rise to a need to minimize downtime while preserving a back- out path.

Some of the considerations for planning a migration strategy are discussed here.

Data size

The sizes of the databases to be migrated obviously affect migration planning, although size does not necessarily affect the cutover time. When a large amount of data must be migrated, the primary consideration is bandwidth. Copy operations are usually performed with efficient sequential I/O. As a conservative estimate, assume 50% utilization of the available network bandwidth for copy operations. For example, an 8GB FC port can transfer about 800MBps in theory. Assuming 50% utilization, a database can be copied at a rate of about 400MBps. Therefore, a 10TB database can be copied in about seven hours at this rate.

Migration over longer distances usually requires a more creative approach, such as the log shipping process explained in Online datafile move. Long-distance IP networks rarely have bandwidth anywhere close to LAN or SAN speeds. In one case, NetApp assisted with the long-distance migration of a 220TB database with very high archive- log generation rates. The chosen approach for data transfer was daily shipment of tapes, because this method offered the maximum possible bandwidth.

Database count

In many cases, the problem with moving a large amount of data is not the data size, but rather it is the complexity of the configuration that supports the database. Simply knowing that 50TB of databases must be migrated is not sufficient information. It could be a single 50TB mission-critical database, a collection of 4, 000 legacy databases, or a mix of production and nonproduction data. In some cases, much of the data consists of clones of a source database. These clones do not need to be migrated at all because they can be easily recreated, especially when the new architecture is designed to leverage NetApp FlexClone volumes.

For migration planning, you must understand how many databases are in scope and how they must be prioritized. As the number of databases increases, the preferred migration option tends to be lower and lower in the stack. For example, copying a single database might be easily performed with RMAN and a short outage. This is host-level replication.

If there are 50 databases, it might be easier to avoid setting up a new file system structure to receive an RMAN copy and instead move the data in place. This process can be done by leveraging host-based LVM migration to relocate data from old LUNs to new LUNs. Doing so moves responsibility from the database administrator (DBA) team to the OS team, and, as a result, data is migrated transparently with respect to the database. The file system configuration is unchanged.

Finally, if 500 databases across 200 servers must be migrated, storage-based options such as the ONTAP Foreign LUN Import (FLI) capability can be used to perform a direct migration of the LUNs.

Rearchitecture requirements

Typically, a database file layout must be altered to leverage the features of the new storage array; however, this is not always the case. For example, the features of EF-Series all-flash arrays are directed primarily at

SAN performance and SAN reliability. In most cases, databases can be migrated to an EF-Series array with no special considerations for data layout. The only requirements are high IOPS, low latency, and robust reliability. Although there are best practices relating to such factors as RAID configuration or Dynamic Disk Pools, EF-Series projects rarely require any significant changes to the overall storage architecture to leverage such features.

In contrast, migration to ONTAP generally requires more consideration of the database layout to make sure that the final configuration delivers maximum value. By itself, ONTAP offers many features for a database environment, even without any specific architecture effort. Most importantly, it delivers the ability to nondisruptively migrate to new hardware when the current hardware reaches its end of life. Generally speaking, a migration to ONTAP is the last migration that you would need to perform. Subsequent hardware is upgraded in place and data is nondisruptively migrated to new media.

With some planning, even more benefits are available. The most important considerations surround the use of snapshots. Snapshots are the basis for performing near-instantaneous backups, restores, and cloning operations. As an example of the power of snapshots, the largest known use is with a single database of 996TB running on about 250 LUNs on 6 controllers. This database can be backed up in 2 minutes, restored in 2 minutes, and cloned in 15 minutes. Additional benefits include the ability to move data around the cluster in response to changes in workload and the application of quality of service (QoS) controls to provide good, consistent performance in a multidatabase environment.

Technologies such as QoS controls, data relocation, snapshots, and cloning work in nearly any configuration. However, some thought is generally required to maximize benefits. In some cases, database storage layouts can require design changes to maximize the investment in the new storage array. Such design changes can affect the migration strategy because host-based or storage-based migrations replicate the original data layout. Additional steps might be required to complete the migration and deliver a data layout optimized for ONTAP. The procedures shown in Oracle migration procedures overview and later demonstrate some of the methods to not just migrate a database, but to migrate it into the optimal final layout with minimal effort.

Cutover time

The maximum allowable service outage during cutover should be determined. It is a common mistake to assume that the entire migration process causes disruption. Many tasks can be completed before any service interruption begins, and many options enable the completion of migration without disruption or outage. Even when disruption is unavoidable, you must still define the maximum allowable service outage because the duration of the cutover time varies from procedure to procedure.

For example, copying a 10TB database typically requires approximately seven hours to complete. If business needs allow a seven- hour outage, file copying is an easy and safe option for migration. If five hours is unacceptable, a simple log- shipping process (see Oracle log shipping) can be set up with minimal effort to reduce the cutover time to approximately 15 minutes. During this time, a database administrator can complete the process. If 15 minutes is unacceptable, the final cutover process can be automated through scripting to reduce the cutover time to just a few minutes. You can always speed up a migration, but doing so comes at the cost of time and effort. The cutover time targets should be based on what is acceptable to the business.

Back-out path

No migration is completely risk free. Even if technology operates perfectly, there is always a possibility of user error. The risk associated with a chosen migration path must be considered alongside the consequences of a failed migration. For example, the transparent online storage migration capability of Oracle ASM is one of its key features, and this method is one of the most reliable known. However, data is being irreversibly copied with this method. In the highly unlikely event that a problem occurs with ASM, there is no easy back- out path. The only option is to either restore the original environment or use ASM to reverse the migration back to the original LUNs. The risk can be minimized, but not eliminated, by performing a snapshot-type backup on the original storage system, assuming the system is capable of performing such an operation.

Rehearsal

Some migration procedures must be fully verified before execution. A need for migration and rehearsal of the cutover process is a common request with mission-critical databases for which migration must be successful and downtime must be minimized. In addition, user- acceptance tests are frequently included as part of the postmigration work, and the overall system can be returned to production only after these tests are complete.

If there is a need for rehearsal, several ONTAP capabilities can make the process much easier. In particular, snapshots can reset a test environment and quickly create multiple space-efficient copies of a database environment.

Procedures

Overview

Many procedures are available for Oracle migration database. The right one depends on your business needs.

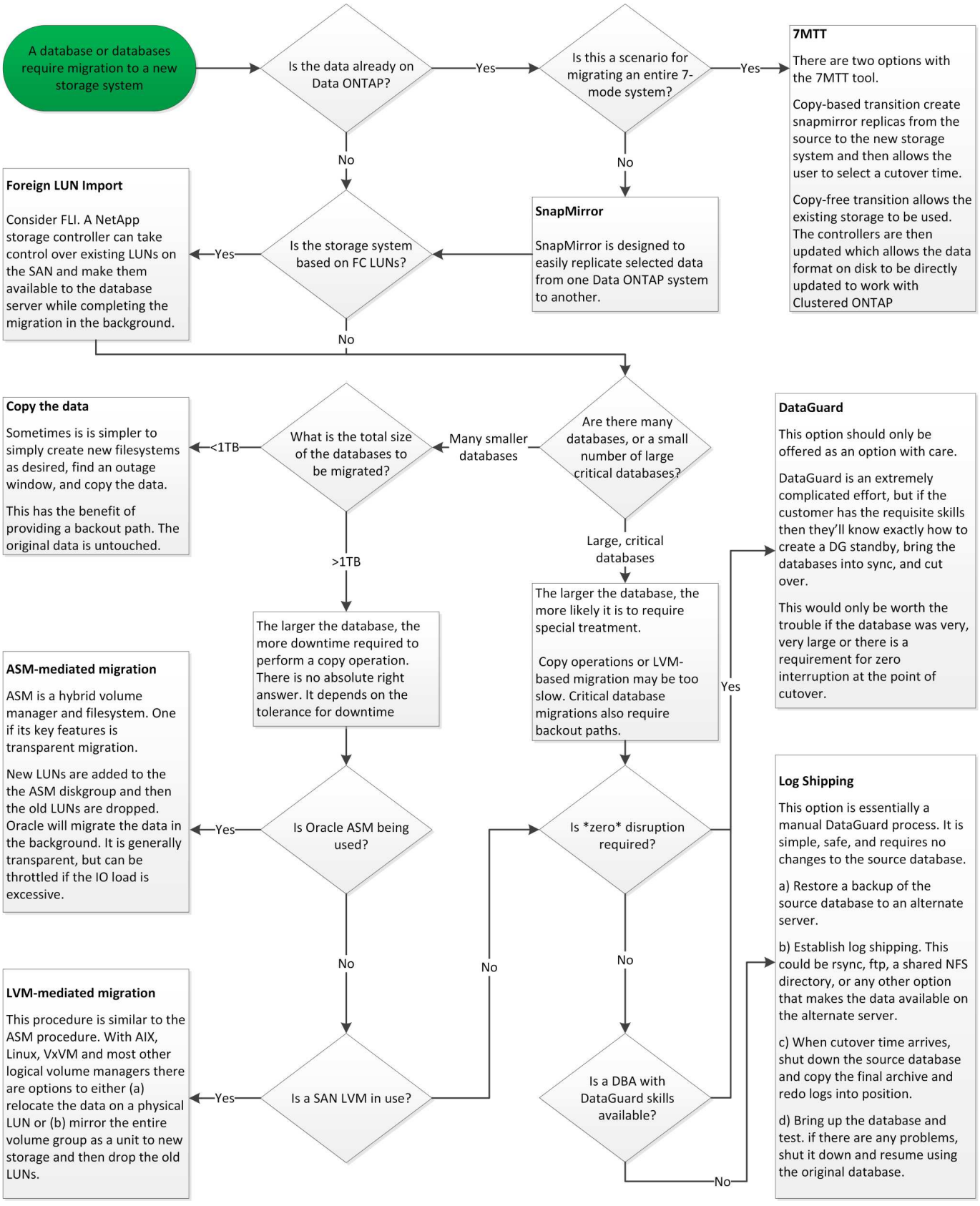
In many cases, system administrators and DBAs have their own preferred methods of relocating physical volume data, mirroring and demirroring, or leveraging Oracle RMAN to copy data.

These procedures are provided primarily as guidance for IT staff less familiar with some of the available options. In addition, the procedures illustrate the tasks, time requirements, and skillset demands for each migration approach. This allows other parties such as NetApp and partner professional services or IT management to more fully appreciate the requirements for each procedure.

There is no single best practice for creating a migration strategy. Creating a plan requires first understanding the availability options and then selecting the method that best suits the needs of the business. The figure below illustrates the basic considerations and typical conclusions made by customers, but it is not universally applicable to all situations.

For example, one step raises the issue of the total database size. The next step depends on whether the database is more or less than 1TB. The recommended steps are just that—recommendations based on typical customer practices. Most customers would not use DataGuard to copy a small database, but some might. Most customers would not attempt to copy a 50TB database because of the time required, but some might have a sufficiently large maintenance window to permit such an operation.

The flowchart below shows the types of considerations on which migration path is best. You can right-click on the image and open it in a new tab to improve readability.



.

Online datafile move

Oracle 12cR1 and higher include the ability to move a datafile while the database remains online. It furthermore works between different filesystem types. For example, a datafile can be relocated from an xfs

filesystem to ASM. This method is not generally used at scale because of the number of individual datafile move operations that would be required, but it is an option worth considering with smaller databases with fewer datafiles.

In addition, simply moving a datafile is a good option for migrating parts of existing databases. For example, less-active datafiles could be relocated to more cost-efficient storage, such as a FabricPool volume which can store idle blocks in Object Store.

Database-level migration

Migration at the database level means allowing the database to relocate data. Specifically, this means log shipping. Technologies such as RMAN and ASM are Oracle products, but, for the purposes of migration, they operate at the host level where they copy files and manage volumes.

Log shipping

The foundation for database- level migration is the Oracle archive log, which contains a log of changes to the database. Most of the time, an archive log is part of a backup and recovery strategy. The recovery process begins with the restoration of a database and then the replaying of one or more archive logs to bring the database to the desired state. This same basic technology can be used to perform a migration with little to no interruption of operations. More importantly, this technology enables migration while leaving the original database untouched, preserving a back- out path.

The migration process begins with restoration of a database backup to a secondary server. You can do so in a variety of ways, but most customers use their normal backup application to restore the data files. After the data files are restored, users establish a method for log shipping. The goal is to create a constant feed of archive logs generated by the primary database and replay them on the restored database to keep them both close to the same state. When the cutover time arrives, the source database is completely shut down and the final archive logs, and in some cases the redo logs, are copied over and replayed. It is critical that the redo logs are also considered because they might contain some of the final transactions committed.

After these logs have been transferred and replayed, both databases are consistent with one another. At this point, most customers perform some basic testing. If any errors are made during the migration process, then the log replay should report errors and fail. It is still advisable to perform some quick tests based on known queries or application-driven activities to verify that the configuration is optimal. It is also a common practice to create one final test table before shutting down the original database to verify whether it is present in the migrated database. This step makes sure that no errors were made during the final log synchronization.

A simple log- shipping migration can be configured out of band with respect to the original database, which makes it particularly useful for mission-critical databases. No configuration changes are required for the source database, and the restoration and initial configuration of the migration environment have no effect on production operations. After log shipping is configured, it places some I/O demands on the production servers. However, log shipping consists of simple sequential reads of the archive logs, which is unlikely to have any effect on production database performance.

Log shipping has proven to be particularly useful for long-distance, high- change-rate migration projects. In one instance, a single 220TB database was migrated to a new location approximately 500 miles away. The change rate was extremely high and security restrictions prevented the use of a network connection. Log shipping was performed by using tape and courier. A copy of the source database was initially restored by using procedures outlined below. The logs were then shipped on a weekly basis by courier until the time of cutover when the final set of tapes was delivered and the logs were applied to the replica database.

Oracle DataGuard

In some cases, a complete DataGuard environment is warranted. It is incorrect to use the term DataGuard to

refer to any log shipping or standby database configuration. Oracle DataGuard is a comprehensive framework for managing database replication, but it is not a replication technology. The primary benefit of a complete DataGuard environment in a migration effort is the transparent switchover from one database to another. DataGuard also enables a transparent switchover back to the original database if a problem is discovered, such as a performance or network connectivity issue with the new environment. A fully configured DataGuard environment requires configuration of not only the database layer but also applications so that applications are able to detect a change in the primary database location. In general, it is not necessary to use DataGuard to complete a migration, but some customers have extensive DataGuard expertise in-house and already rely on it for migration work.

Rearchitecture

As discussed before, leveraging the advanced features of storage arrays sometimes requires changing the database layout. Furthermore, a change in storage protocol such as moving from ASM to an NFS file system necessarily alters the file system layout.

One of the principal advantages of log shipping methods, including DataGuard, is that the replication destination does not have to match the source. There are no issues with using a log-shipping approach to migrate from ASM to a regular file system or vice versa. The precise layout of data files can be changed at the destination to optimize the use of Pluggable Database (PDB) technology or to set QoS controls selectively on certain files. In other words, a migration process based on log shipping allows you to optimize the database storage layout easily and safely.

Server resources

One limitation to database-level migration is the need for a second server. There are two ways this second server can be used:

1. You can use the second server as a permanent new home for the database.
2. You can use the second server as a temporary staging server. After data migration to the new storage array is complete and tested, the LUN or NFS file systems are disconnected from the staging server and reconnected to the original server.

The first option is the easiest, but using it might not be feasible in very large environments requiring very powerful servers. The second option requires extra work to relocate the file systems back to the original location. This can be a simple operation in which NFS is used as the storage protocol because the file systems can be unmounted from the staging server and remounted on the original server.

Block-based file systems require extra work to update FC zoning or iSCSI initiators. With most logical volume managers (including ASM), the LUNs are automatically detected and brought online after they are made available on the original server. However, some file system and LVM implementations might require more work to export and import the data. The precise procedure might vary, but it is generally easy to establish a simple, repeatable procedure to complete the migration and rehome the data on the original server.

Although it is possible to set up log shipping and replicate a database within a single server environment, the new instance must have a different process SID to replay the logs. It is possible to temporarily bring up the database under a different set of process IDs with a different SID and change it later. However, doing so can lead to a lot of complicated management activities, and it puts the database environment at risk of user error.

Host-level migration

Migrating data at the host level means using the host operating system and associated utilities to complete the migration. This process includes any utility that copies data, including Oracle RMAN and Oracle ASM.

Data copying

The value of a simple copy operation should not be underestimated. Modern network infrastructures can move data at rates measured in gigabytes per second, and file copy operations are based on efficient sequential read and write I/O. More disruption is unavoidable with a host copy operation when compared to log shipping, but a migration is more than just the data movement. It generally includes changes to networking, the database restart time, and postmigration testing.

The actual time required to copy data might not be significant. Furthermore, a copy operation preserves a guaranteed back- out path because the original data remains untouched. If any problems are encountered during the migration process, the original file systems with the original data can be reactivated.

Replatforming

Replatforming refers to a change in the CPU type. When a database is migrated from a traditional Solaris, AIX, or HP-UX platform to x86 Linux, the data must be reformatted because of changes in the CPU architecture. SPARC, IA64, and POWER CPUs are known as big endian processors, while the x86 and x86_64 architectures are known as little endian. As a result, some data within Oracle data files is ordered differently depending on the processor in use.

Traditionally, customers have used DataPump to replicate data across platforms. DataPump is a utility that creates a special type of logical data export that can be more rapidly imported at the destination database. Because it creates a logical copy of the data, DataPump leaves the dependencies of processor endianness behind. DataPump is still used by some customers for replatforming, but a faster option has become available with Oracle 11g: cross-platform transportable tablespaces. This advance allows a tablespace to be converted to a different endian format in place. This is a physical transformation that offers better performance than a DataPump export, which must convert physical bytes to logical data and then convert back to physical bytes.

A complete discussion of DataPump and transportable tablespaces is beyond the scope NetApp documentation, but NetApp has some recommendations based on our experience assisting customers during migration to a new storage array log with a new CPU architecture:

- If DataPump is being used, the time required to complete the migration should be measured in a test environment. Customers are sometimes surprised at the time required to complete the migration. This unexpected additional downtime can cause disruption.
- Many customers mistakenly believe that cross-platform transportable tablespaces do not require data conversion. When a CPU with a different endian is used, an RMAN `convert` operation must be performed on the data files beforehand. This is not an instantaneous operation. In some cases, the conversion process can be sped up by having multiple threads operating on different data files, but the conversion process cannot be avoided.

Logical volume manager-driven migration

LVMs work by taking a group of one or more LUNs and breaking them into small units generally referred to as extents. The pool of extents is then used as a source to create logical volumes that are essentially virtualized. This virtualization layer delivers value in various ways:

- Logical volumes can use extents drawn from multiple LUNs. When a file system is created on a logical volume, it can use the full performance capabilities of all LUNs. It also promotes the even loading of all LUNs in the volume group, delivering more predictable performance.
- Logical volumes can be resized by adding and, in some cases, removing extents. Resizing a file system on a logical volume is generally nondisruptive.
- Logical volumes can be nondisruptively migrated by moving the underlying extents.

Migration using an LVM works in one of two ways: moving an extent or mirroring/demirroring an extent. LVM migration uses efficient large-block sequential I/O and only rarely creates any performance concerns. If this does become an issue, there are usually options for throttling the I/O rate. Doing so increases the time required to complete the migration and yet reduces the I/O burden on the host and storage systems.

Mirror and demirror

Some volume managers, such as AIX LVM, allow the user to specify the number of copies for each extent and to control which devices host each copy. Migration is accomplished by taking an existing logical volume, mirroring the underlying extents to the new volumes, waiting for the copies to synchronize, and then dropping the old copy. If a back- out path is desired, a snapshot of the original data can be created before the point at which the mirror copy is dropped. Alternatively, the server can be shut down briefly to mask original LUNs before forcibly deleting the contained mirror copies. Doing so preserves a recoverable copy of the data in its original location.

Extent migration

Almost all volume managers allow extents to be migrated, and sometimes multiple options exist. For example, some volume managers allow an administrator to relocate the individual extents for a specific logical volume from old to new storage. Volume managers such as Linux LVM2 offer the `pvmove` command, which relocates all extents on the specified LUN device to a new LUN. After the old LUN is evacuated, it can be removed.

The primary risk to operations is the removal of old, unused LUNs from the configuration. Great care must be taken when changing FC zoning and removing stale LUN devices.

Oracle Automatic Storage Management

Oracle ASM is a combined logical volume manager and file system. At a high level, Oracle ASM takes a collection of LUNs, breaks them into small units of allocation, and presents them as a single volume known as an ASM disk group. ASM also includes the ability to mirror the disk group by setting the redundancy level. A volume can be unmirrored (external redundancy), mirrored (normal redundancy), or three-way mirrored (high redundancy). Care must be taken when configuring the redundancy level because it cannot be changed after creation.

ASM also provides file system functionality. Although the file system is not visible directly from the host, the Oracle database can create, move, and delete files and directories on an ASM disk group. Also, the structure can be navigated by using the asmcmd utility.

As with other LVM implementations, Oracle ASM optimizes I/O performance by striping and load-balancing the I/O of each file across all available LUNs. Second, the underlying extents can be relocated to enable both resizing of the ASM disk group as well as migration. Oracle ASM automates the process through the rebalancing operation. New LUNs are added to an ASM disk group and old LUNs are dropped, which triggers extent relocation and subsequent drop of the evacuated LUN from the disk group. This process is one of the most proven methods of migration, and the reliability of ASM at delivering transparent migration is possibly its most important feature.

Because the mirroring level of Oracle ASM is fixed, it cannot be used with the mirror and demirror method of migration.

Storage-level migration

Storage-level migration means performing the migration below both the application and operating system level. In the past, this sometimes meant using specialized devices that would copy LUNs at the network level, but these capabilities are now found natively in ONTAP.

SnapMirror

Migration of databases from between NetApp systems is almost universally performed with the NetApp SnapMirror data replication software. The process involves setting up a mirror relationship for the volumes to be migrated, allowing them to synchronize, and then waiting for the cutover window. When it arrives, the source database is shut down, one final mirror update is performed, and the mirror is broken. The replica volumes are then ready for use, either by mounting a contained NFS file system directory or by discovering the contained LUNs and starting the database.

Relocating volumes within a single ONTAP cluster is not considered migration, but rather a routine `volume move` operation. SnapMirror is used as the data replication engine within the cluster. This process is fully automated. There are no additional migration steps to be performed when attributes of the volume, such as LUN mapping or the NFS export permissions, are moved with the volume itself. The relocation is nondisruptive to host operations. In some cases, network access must be updated to make sure that the newly relocated data is accessed in the most efficient way possible, but these tasks are also nondisruptive.

Foreign LUN Import (FLI)

FLI is a feature that allows a Data ONTAP system running 8.3 or higher to migrate an existing LUN from another storage array. The procedure is simple: The ONTAP system is zoned to the existing storage array as if it was any other SAN host. Data ONTAP then takes control of the desired legacy LUNs and migrates the underlying data. In addition, the import process uses the efficiency settings of the new volume as data is migrated, meaning that data can be compressed and deduplicated inline during the migration process.

The first implementation of FLI in Data ONTAP 8.3 permitted only offline migration. This was an extremely fast transfer, but it still meant that the LUN data was unavailable until the migration was complete. Online migration was introduced in Data ONTAP 8.3.1. This kind of migration minimizes disruption by allowing ONTAP to serve LUN data during the transfer process. There is a brief disruption while the host is rezoned to use the LUNs through ONTAP. However, as soon as those changes are made, the data is once again accessible and remains accessible throughout the migration process.

Read I/O is proxied through ONTAP until the copy operation is complete, while write I/O is synchronously written to both the foreign and ONTAP LUN. The two LUN copies are kept in sync in this manner until the administrator executes a complete cutover that releases the foreign LUN and no longer replicates writes.

FLI is designed to work with FC, but if there is a desire to change to iSCSI, then the migrated LUN can easily be remapped as an iSCSI LUN after migration is completed.

Among the features of FLI is automatic alignment detection and adjustment. In this context, the term alignment refers to a partition on a LUN device. Optimum performance requires that I/O be aligned to 4K blocks. If a partition is placed at an offset that is not a multiple of 4K, performance suffers.

There is a second aspect of alignment that cannot be corrected by adjusting a partition offset—the file system block size. For example, a ZFS file system generally defaults to an internal block size of 512 bytes. Other customers using AIX have occasionally created jfs2 file systems with a 512- or 1, 024- byte block size. Although the file system might be aligned to a 4K boundary, the files created within that file system are not and performance suffers.

FLI should not be used in these circumstances. Although the data is accessible after migration, the result is file systems with serious performance limitations. As a general principle, any file system supporting a random overwrite workload on ONTAP should use a 4K block size. This is primarily applicable to workloads such as database data files and VDI deployments. The block size can be identified using the relevant host operating system commands.

For example, on AIX, the block size can be viewed with `lsfs -q`. With Linux, `xfs_info` and `tune2fs` can

be used for `xfs` and `ext3/ext4`, respectively. With `zfs`, the command is `zdb -C`.

The parameter that controls the block size is `ashift` and generally defaults to a value of 9, which means 2^9, or 512 bytes. For optimum performance, the `ashift` value must be 12 (2^12=4K). This value is set at the time the zpool is created and cannot be changed, which means that data zpools with an `ashift` other than 12 should be migrated by copying data to a newly created zpool.

Oracle ASM does not have a fundamental block size. The only requirement is that the partition on which the ASM disk is built must be properly aligned.

7-Mode Transition Tool

The 7-Mode Transition Tool (7MTT) is an automation utility used to migrate large 7- Mode configurations to ONTAP. Most database customers find other methods easier, in part because they usually migrate their environments database by database rather than relocating the entire storage footprint. Additionally, databases are frequently only a part of a larger storage environment. Therefore, databases are often migrated individually, and then the remaining environment can be moved with 7MTT.

There is a small but significant number of customers who have storage systems that are dedicated to complicated database environments. These environments might contain many volumes, snapshots, and numerous configuration details such as export permissions, LUN initiator groups, user permissions, and Lightweight Directory Access Protocol configuration. In such cases, the automation abilities of 7MTT can simplify a migration.

7MTT can operate in one of two modes:

- **Copy- based transition (CBT).** 7MTT with CBT sets up SnapMirror volumes from an existing 7- Mode system in the new environment. After the data is in sync, 7MTT orchestrates the cutover process.
- **Copy- free transition (CFT).** 7MTT with CFT is based on the in-place conversion of existing 7- Mode disk shelves. No data is copied, and the existing disk shelves can be reused. The existing data protection and storage efficiency configuration is preserved.

The primary difference between these two options is that copy-free transition is a big- bang approach in which all disk shelves attached to the original 7- Mode HA pair must be relocated to the new environment. There is no option to move a subset of shelves. The copy-based approach allows selected volumes to be moved. There is also potentially a longer cutover window with copy-free transition because of the tie required to recable disk shelves and convert metadata. Based on field experience, NetApp recommends allowing 1 hour for relocating and recabling disk shelves and between 15 minutes and 2 hours for metadata conversion.

Datafile migration

Individual Oracle datafiles can be moved with a single command.

For example, the following command moves the datafile IOPST.dbf from filesystem `/oradata2` to filesystem `/oradata3`.

```
SQL> alter database move datafile  '/oradata2/NTAP/IOPS002.dbf' to
'/oradata3/NTAP/IOPS002.dbf';
Database altered.
```

Moving a datafile with this method can be slow, but it normally should not produce enough I/O that it interferes with the day-to-day database workloads. In contrast, migration via ASM rebalancing can run much faster but at

the expense of slowing down the overall database while the data is being moved.

The time required to move datafiles can easily be measured by creating a test datafile and then moving it. The elapsed time for the operation is recorded in the v$session data:

```
SQL> set linesize 300;
SQL> select elapsed_seconds||':'||message from v$session_longops;
ELAPSED_SECONDS||':'||MESSAGE
------------------------------------------------------------------------------
---------------
351:Online data file move: data file 8: 22548578304 out of 22548578304
bytes done
SQL> select bytes / 1024 / 1024 /1024 as GB from dba_data_files where
FILE_ID = 8;
        GB
----------
        21
```

In this example, the file that was moved was datafile 8, which was 21GB in size and required about 6 minutes to migrate. The time required obviously depends on the capabilities of the storage system, the storage network, and the overall database activity occurring at the time of migration.

Log shipping

The goal of a migration using log shipping is to create a copy of the original data files at a new location and then establish a method of shipping changes into the new environment.

Once established, log shipment and replay can be automated to keep the replica database largely in sync with the source. For example, a cron job can be scheduled to (a) copy the most recent logs to the new location and (b) replay them every 15 minutes. Doing so provides minimal disruption at the time of cutover because no more than 15 minutes of archive logs must be replayed.

The procedure shown below is also essentially a database clone operation. The logic shown is similar to the engine within NetApp SnapManager for Oracle (SMO) and the NetApp SnapCenter Oracle Plug- in. Some customers have used the procedure shown within scripts or WFA workflows for custom cloning operations. Although this procedure is more manual than using either SMO or SnapCenter, it is still readily scripted and the data management APIs within ONTAP further simplify the process.

Log shipping - file system to file system

This example demonstrates the migration of a database called WAFFLE from an ordinary file system to another ordinary file system located on a different server. It also illustrates the use of SnapMirror to make a rapid copy of data files, but this is not an integral part of the overall procedure.

Create database backup

The first step is to create a database backup. Specifically, this procedure requires a set of data files that can be used for archive log replay.

Environment

In this example, the source database is on an ONTAP system. The simplest method to create a backup of a database is by using a snapshot. The database is placed in hot backup mode for a few seconds while a `snapshot create` operation is executed on the volume hosting the data files.

```
SQL> alter database begin backup;
Database altered.
```

```
Cluster01::*> snapshot create -vserver vserver1 -volume jfsc1_oradata
hotbackup
Cluster01::*>
```

```
SQL> alter database end backup;
Database altered.
```

The result is a snapshot on disk called `hotbackup` that contains an image of the data files while in hot backup mode. When combined with the appropriate archive logs to make the data files consistent, the data in this snapshot can be used as the basis of a restore or a clone. In this case, it is replicated to the new server.

Restore to new environment

The backup must now be restored in the new environment. This can be done in a number of ways, including Oracle RMAN, restoration from a backup application like NetBackup, or a simple copy operation of data files that were placed in hot backup mode.

In this example, SnapMirror is used to replicate the snapshot hotbackup to a new location.

1. Create a new volume to receive the snapshot data. Initialize the mirroring from `jfsc1_oradata` to `vol_oradata`.

   ```
   Cluster01::*> volume create -vserver vserver1 -volume vol_oradata
   -aggregate data_01 -size 20g -state online -type DP -snapshot-policy
   none -policy jfsc3
   [Job 833] Job succeeded: Successful
   ```

   ```
   Cluster01::*> snapmirror initialize -source-path vserver1:jfsc1_oradata
   -destination-path vserver1:vol_oradata
   Operation is queued: snapmirror initialize of destination
   "vserver1:vol_oradata".
   Cluster01::*> volume mount -vserver vserver1 -volume vol_oradata
   -junction-path /vol_oradata
   Cluster01::*>
   ```

2. After the state is set by SnapMirror, indicating that synchronization is complete, update the mirror based specifically on the desired snapshot.

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields state
source-path             destination-path        state
----------------------- ----------------------- ------------
vserver1:jfsc1_oradata vserver1:vol_oradata SnapMirrored
```

```
Cluster01::*> snapmirror update -destination-path vserver1:vol_oradata
-source-snapshot hotbackup
Operation is queued: snapmirror update of destination
"vserver1:vol_oradata".
```

3. Successful synchronization can be verified by viewing the `newest-snapshot` field on the mirror volume.

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields newest-snapshot
source-path             destination-path        newest-snapshot
----------------------- ----------------------- ---------------
vserver1:jfsc1_oradata vserver1:vol_oradata hotbackup
```

4. The mirror can then be broken.

```
Cluster01::> snapmirror break -destination-path vserver1:vol_oradata
Operation succeeded: snapmirror break for destination
"vserver1:vol_oradata".
Cluster01::>
```

5. Mount the new file system.With block-based file systems, the precise procedures vary based on the LVM in use. FC zoning or iSCSI connections must be configured. After connectivity to the LUNs is established, commands such as Linux `pvscan` might be needed to discover which volume groups or LUNs need to be properly configured to be discoverable by ASM.

   In this example, a simple NFS file system is used. This file system can be mounted directly.

```
fas8060-nfs1:/vol_oradata        19922944   1639360   18283584   9%
/oradata
fas8060-nfs1:/vol_logs            9961472       128    9961344   1%
/logs
```

Create controlfile creation template

You must next create a controlfile template. The `backup controlfile to trace` command creates text commands to recreate a controlfile. This function can be useful for restoring a database from backup under some circumstances, and it is often used with scripts that perform tasks such as database cloning.

1. The output of the following command is used to recreate the controlfiles for the migrated database.

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctrl';
Database altered.
```

2. After the controlfiles have been created, copy the file to the new server.

```
[oracle@jfsc3 tmp]$ scp oracle@jfsc1:/tmp/waffle.ctrl /tmp/
oracle@jfsc1's password:
waffle.ctrl                                           100% 5199
5.1KB/s   00:00
```

Backup parameter file

A parameter file is also required in the new environment. The simplest method is to create a pfile from the current spfile or pfile. In this example, the source database is using an spfile.

```
SQL> create pfile='/tmp/waffle.tmp.pfile' from spfile;
File created.
```

Create oratab entry

The creation of an oratab entry is required for the proper functioning of utilities such as oraenv. To create an oratab entry, complete the following step.

```
WAFFLE:/orabin/product/12.1.0/dbhome_1:N
```

Prepare directory structure

If the required directories were not already present, you must create them or the database startup procedure fails. To prepare the directory structure, complete the following minimum requirements.

```
[oracle@jfsc3 ~]$ . oraenv
ORACLE_SID = [oracle] ? WAFFLE
The Oracle base has been set to /orabin
[oracle@jfsc3 ~]$ cd $ORACLE_BASE
[oracle@jfsc3 orabin]$ cd admin
[oracle@jfsc3 admin]$ mkdir WAFFLE
[oracle@jfsc3 admin]$ cd WAFFLE
[oracle@jfsc3 WAFFLE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

Parameter file updates

1. To copy the parameter file to the new server, run the following commands. The default location is the `$ORACLE_HOME/dbs` directory. In this case, the pfile can be placed anywhere. It is only being used as an intermediate step in the migration process.

```
[oracle@jfsc3 admin]$ scp oracle@jfsc1:/tmp/waffle.tmp.pfile
$ORACLE_HOME/dbs/waffle.tmp.pfile
oracle@jfsc1's password:
waffle.pfile                                          100%  916
0.9KB/s   00:00
```

1. Edit the file as required. For example, if the archive log location has changed, the pfile must be altered to reflect the new location. In this example, only the controlfiles are being relocated, in part to distribute them between the log and data file systems.

```
[root@jfsc1 tmp]# cat waffle.pfile
WAFFLE.__data_transfer_cache_size=0
WAFFLE.__db_cache_size=507510784
WAFFLE.__java_pool_size=4194304
WAFFLE.__large_pool_size=20971520
WAFFLE.__oracle_base='/orabin'#ORACLE_BASE set from environment
WAFFLE.__pga_aggregate_target=268435456
WAFFLE.__sga_target=805306368
WAFFLE.__shared_io_pool_size=29360128
WAFFLE.__shared_pool_size=234881024
WAFFLE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/WAFFLE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/oradata//WAFFLE/control01.ctl','/oradata//WAFFLE/contr
ol02.ctl'
*.control_files='/oradata/WAFFLE/control01.ctl','/logs/WAFFLE/control02.
ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='WAFFLE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=WAFFLEXDB)'
*.log_archive_dest_1='LOCATION=/logs/WAFFLE/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

2. After the edits are complete, create an spfile based on this pfile.

```
SQL> create spfile from pfile='waffle.tmp.pfile';
File created.
```

Recreate controlfiles

In a previous step, the output of `backup controlfile to trace` was copied to the new server. The specific portion of the output required is the `controlfile recreation` command. This information can be found in the file under the section marked `Set #1. NORESETLOGS`. It starts with the line `create controlfile reuse database` and should include the word `noresetlogs`. It ends with the semicolon (;) character.

1. In this example procedure, the file reads as follows.

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/logs/WAFFLE/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/logs/WAFFLE/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/logs/WAFFLE/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

2. Edit this script as desired to reflect the new location of the various files. For example, certain data files known to support high I/O might be redirected to a file system on a high- performance storage tier. In other cases, the changes might be purely for administrator reasons, such as isolating the data files of a given PDB in dedicated volumes.
3. In this example, the `DATAFILE` stanza is left unchanged, but the redo logs are moved to a new location in `/redo` rather than sharing space with archive logs in `/logs`.

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

```
SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             331353200 bytes
Database Buffers          465567744 bytes
Redo Buffers                5455872 bytes
SQL> CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
  2      MAXLOGFILES 16
  3      MAXLOGMEMBERS 3
  4      MAXDATAFILES 100
  5      MAXINSTANCES 8
  6      MAXLOGHISTORY 292
  7  LOGFILE
  8    GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  9    GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
 10    GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
 11  -- STANDBY LOGFILE
 12  DATAFILE
 13    '/oradata/WAFFLE/system01.dbf',
 14    '/oradata/WAFFLE/sysaux01.dbf',
 15    '/oradata/WAFFLE/undotbs01.dbf',
 16    '/oradata/WAFFLE/users01.dbf'
 17  CHARACTER SET WE8MSWIN1252
 18  ;
Control file created.
SQL>
```

If any files are misplaced or parameters are misconfigured, errors are generated that indicate what must be fixed. The database is mounted, but it is not yet open and cannot be opened because the data files in use are still marked as being in hot backup mode. Archive logs must first be applied to make the database consistent.

Initial log replication

At least one log reply operation is required to make the data files consistent. Many options are available to replay logs. In some cases, the original archive log location on the original server can be shared through NFS, and log reply can be done directly. In other cases, the archive logs must be copied.

For example, a simple `scp` operation can copy all current logs from the source server to the migration server:

```
[oracle@jfsc3 arch]$ scp jfsc1:/logs/WAFFLE/arch/* ./
oracle@jfsc1's password:
1_22_912662036.dbf                                          100%   47MB
47.0MB/s   00:01
1_23_912662036.dbf                                          100%   40MB
40.4MB/s   00:00
1_24_912662036.dbf                                          100%   45MB
45.4MB/s   00:00
1_25_912662036.dbf                                          100%   41MB
40.9MB/s   00:01
1_26_912662036.dbf                                          100%   39MB
39.4MB/s   00:00
1_27_912662036.dbf                                          100%   39MB
38.7MB/s   00:00
1_28_912662036.dbf                                          100%   40MB
40.1MB/s   00:01
1_29_912662036.dbf                                          100%   17MB
16.9MB/s   00:00
1_30_912662036.dbf                                          100%  636KB
636.0KB/s   00:00
```

Initial log replay

After the files are in the archive log location, they can be replayed by issuing the command `recover database until cancel` followed by the response `AUTO` to automatically replay all available logs.

```
SQL> recover database until cancel;
ORA-00279: change 382713 generated at 05/24/2016 09:00:54 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_23_912662036.dbf
ORA-00280: change 382713 for thread 1 is in sequence #23
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 405712 generated at 05/24/2016 15:01:05 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_24_912662036.dbf
ORA-00280: change 405712 for thread 1 is in sequence #24
ORA-00278: log file '/logs/WAFFLE/arch/1_23_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
ORA-00278: log file '/logs/WAFFLE/arch/1_30_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_31_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

The final archive log reply reports an error, but this is normal. The log indicates that `sqlplus` was seeking a particular log file and did not find it. The reason is, most likely, that the log file does not exist yet.

If the source database can be shut down before copying archive logs, this step must be performed only once. The archive logs are copied and replayed, and then the process can continue directly to the cutover process that replicates the critical redo logs.

Incremental log replication and replay

In most cases, migration is not performed right away. It could be days or even weeks before the migration process is completed, which means that the logs must be continuously shipped to the replica database and replayed. Therefore, when cutover arrives, minimal data must be transferred and replayed.

Doing so can be scripted in many ways, but one of the more popular methods is using rsync, a common file replication utility. The safest way to use this utility is to configure it as a daemon. For example, the `rsyncd.conf` file that follows shows how to create a resource called `waffle.arch` that is accessed with Oracle user credentials and is mapped to `/logs/WAFFLE/arch`. Most importantly, the resource is set to read-only, which allows the production data to be read but not altered.

```
[root@jfsc1 arch]# cat /etc/rsyncd.conf
[waffle.arch]
   uid=oracle
   gid=dba
   path=/logs/WAFFLE/arch
   read only = true
[root@jfsc1 arch]# rsync --daemon
```

The following command synchronizes the new server's archive log destination against the rsync resource `waffle.arch` on the original server. The `t` argument in `rsync - potg` causes the file list to be compared based on timestamp, and only new files are copied. This process provides an incremental update of the new server. This command can also be scheduled in cron to run on a regular basis.

```
[oracle@jfsc3 arch]$ rsync -potg --stats --progress jfsc1::waffle.arch/*
/logs/WAFFLE/arch/
1_31_912662036.dbf
      650240 100%  124.02MB/s    0:00:00 (xfer#1, to-check=8/18)
1_32_912662036.dbf
     4873728 100%  110.67MB/s    0:00:00 (xfer#2, to-check=7/18)
1_33_912662036.dbf
     4088832 100%   50.64MB/s    0:00:00 (xfer#3, to-check=6/18)
1_34_912662036.dbf
     8196096 100%   54.66MB/s    0:00:00 (xfer#4, to-check=5/18)
1_35_912662036.dbf
    19376128 100%   57.75MB/s    0:00:00 (xfer#5, to-check=4/18)
1_36_912662036.dbf
       71680 100%  201.15kB/s    0:00:00 (xfer#6, to-check=3/18)
1_37_912662036.dbf
     1144320 100%    3.06MB/s    0:00:00 (xfer#7, to-check=2/18)
1_38_912662036.dbf
    35757568 100%   63.74MB/s    0:00:00 (xfer#8, to-check=1/18)
1_39_912662036.dbf
      984576 100%    1.63MB/s    0:00:00 (xfer#9, to-check=0/18)
Number of files: 18
Number of files transferred: 9
Total file size: 399653376 bytes
Total transferred file size: 75143168 bytes
Literal data: 75143168 bytes
Matched data: 0 bytes
File list size: 474
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 204
Total bytes received: 75153219
sent 204 bytes  received 75153219 bytes  150306846.00 bytes/sec
total size is 399653376  speedup is 5.32
```

After the logs have been received, they must be replayed. Previous examples show the use of sqlplus to manually run `recover database until cancel`, a process that can easily be automated. The example shown here uses the script described in Replay Logs on Database. The scripts accept an argument that specifies the database requiring a replay operation. This permits the same script to be used in a multidatabase migration effort.

```
[oracle@jfsc3 logs]$ ./replay.logs.pl WAFFLE
ORACLE_SID = [WAFFLE] ? The Oracle base remains unchanged with value
/orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu May 26 10:47:16 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 814256 generated at 05/26/2016 04:52:30 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_32_912662036.dbf
ORA-00280: change 814256 for thread 1 is in sequence #32
ORA-00278: log file '/logs/WAFFLE/arch/1_31_912662036.dbf' no longer
needed for
this recovery
ORA-00279: change 814780 generated at 05/26/2016 04:53:04 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_33_912662036.dbf
ORA-00280: change 814780 for thread 1 is in sequence #33
ORA-00278: log file '/logs/WAFFLE/arch/1_32_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
ORA-00278: log file '/logs/WAFFLE/arch/1_39_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

Cutover

When you are ready to cut over to the new environment, you must perform one final synchronization that includes both archive logs and the redo logs. If the original redo log location is not already known, it can be identified as follows:

```
SQL> select member from v$logfile;
MEMBER
--------------------------------------------------------------------------------
------
/logs/WAFFLE/redo/redo01.log
/logs/WAFFLE/redo/redo02.log
/logs/WAFFLE/redo/redo03.log
```

1. Shut down the source database.
2. Perform one final synchronization of the archive logs on the new server with the desired method.
3. The source redo logs must be copied to the new server. In this example, the redo logs were relocated to a new directory at `/redo`.

   ```
   [oracle@jfsc3 logs]$ scp jfsc1:/logs/WAFFLE/redo/* /redo/
   oracle@jfsc1's password:
   redo01.log
   100%   50MB  50.0MB/s   00:01
   redo02.log
   100%   50MB  50.0MB/s   00:00
   redo03.log
   100%   50MB  50.0MB/s   00:00
   ```

4. At this stage, the new database environment contains all of the files required to bring it to the exact same state as the source. The archive logs must be replayed one final time.

```
SQL> recover database until cancel;
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

5. Once complete, the redo logs must be replayed. If the message `Media recovery complete` is returned, the process is successful and the databases are synchronized and can be opened.

```
SQL> recover database;
Media recovery complete.
SQL> alter database open;
Database altered.
```

Log shipping - ASM to file system

This example demonstrates the use of Oracle RMAN to migrate a database. It is very similar to the prior example of file system to file system log shipping, but the files on ASM are not visible to the host. The only options for migrating data located on ASM devices is either by relocating the ASM LUN or by using Oracle RMAN to perform the copy operations.

Although RMAN is a requirement for copying files from Oracle ASM, the use of RMAN is not limited to ASM. RMAN can be used to migrate from any type of storage to any other type.

This example shows the relocation of a database called PANCAKE from ASM storage to a regular file system located on a different server at paths `/oradata` and `/logs`.

Create database backup

The first step is to create a backup of the database to be migrated to an alternate server. Because the source uses Oracle ASM, RMAN must be used. A simple RMAN backup can be performed as follows. This method creates a tagged backup that can be easily identified by RMAN later in the procedure.

The first command defines the type of destination for the backup and the location to be used. The second initiates the backup of the data files only.

```
RMAN> configure channel device type disk format '/rman/pancake/%U';
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT   '/rman/pancake/%U';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT   '/rman/pancake/%U';
new RMAN configuration parameters are successfully stored
RMAN> backup database tag 'ONTAP_MIGRATION';
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=251 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/PANCAKE/system01.dbf
input datafile file number=00002 name=+ASM0/PANCAKE/sysaux01.dbf
input datafile file number=00003 name=+ASM0/PANCAKE/undotbs101.dbf
input datafile file number=00004 name=+ASM0/PANCAKE/users01.dbf
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/1gr6c161_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/1hr6c164_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

Backup controlfile

A backup controlfile is required later in the procedure for the `duplicate database` operation.

```
RMAN> backup current controlfile format '/rman/pancake/ctrl.bkp';
Starting backup at 24-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/ctrl.bkp tag=TAG20160524T032651 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

Backup parameter file

A parameter file is also required in the new environment. The simplest method is to create a pfile from the current spfile or pfile. In this example, the source database uses an spfile.

```
RMAN> create pfile='/rman/pancake/pfile' from spfile;
Statement processed
```

ASM file rename script

Several file locations currently defined in the controlfiles change when the database is moved. The following script creates an RMAN script to make the process easier. This example shows a database with a very small number of data files, but typically databases contain hundreds or even thousands of data files.

This script can be found in ASM to File System Name Conversion and it does two things.

First, it creates a parameter to redefine the redo log locations called `log_file_name_convert`. It is essentially a list of alternating fields. The first field is the location of a current redo log, and the second field is the location on the new server. The pattern is then repeated.

The second function is to supply a template for data file renaming. The script loops through the data files, pulls the name and file number information, and formats it as an RMAN script. Then it does the same with the temp files. The result is a simple rman script that can be edited as desired to make sure that the files are restored to the desired location.

```
SQL> @/rman/mk.rename.scripts.sql
Parameters for log file conversion:
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/NEW_PATH/redo01.log','+ASM0/PANCAKE/redo02.log',
'/NEW_PATH/redo02.log','+ASM0/PANCAKE/redo03.log', '/NEW_PATH/redo03.log'
rman duplication script:
run
{
set newname for datafile 1 to '+ASM0/PANCAKE/system01.dbf';
set newname for datafile 2 to '+ASM0/PANCAKE/sysaux01.dbf';
set newname for datafile 3 to '+ASM0/PANCAKE/undotbs101.dbf';
set newname for datafile 4 to '+ASM0/PANCAKE/users01.dbf';
set newname for tempfile 1 to '+ASM0/PANCAKE/temp01.dbf';
duplicate target database for standby backup location INSERT_PATH_HERE;
}
PL/SQL procedure successfully completed.
```

Capture the output of this screen. The `log_file_name_convert` parameter is placed in the pfile as described below. The RMAN data file rename and duplicate script must be edited accordingly to place the data files in the desired locations. In this example, they are all placed in `/oradata/pancake`.

```
run
{
set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
set newname for datafile 4 to '/oradata/pancake/users.dbf';
set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
duplicate target database for standby backup location '/rman/pancake';
}
```

Prepare directory structure

The scripts are almost ready to execute, but first the directory structure must be in place. If the required directories are not already present, they must be created or the database startup procedure fails. The example below reflects the minimum requirements.

```
[oracle@jfsc2 ~]$ mkdir /oradata/pancake
[oracle@jfsc2 ~]$ mkdir /logs/pancake
[oracle@jfsc2 ~]$ cd /orabin/admin
[oracle@jfsc2 admin]$ mkdir PANCAKE
[oracle@jfsc2 admin]$ cd PANCAKE
[oracle@jfsc2 PANCAKE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

Create oratab entry

The following command is required for utilities such as oraenv to work properly.

```
PANCAKE:/orabin/product/12.1.0/dbhome_1:N
```

Parameter updates

The saved pfile must be updated to reflect any path changes on the new server. The data file path changes are changed by the RMAN duplication script, and nearly all databases require changes to the `control_files` and `log_archive_dest` parameters. There might also be audit file locations that must be changed, and parameters such as `db_create_file_dest` might not be relevant outside of ASM. An experienced DBA should carefully review the proposed changes before proceeding.

In this example, the key changes are the controlfile locations, the log archive destination, and the addition of the `log_file_name_convert` parameter.

```
PANCAKE.__data_transfer_cache_size=0
PANCAKE.__db_cache_size=545259520
PANCAKE.__java_pool_size=4194304
PANCAKE.__large_pool_size=25165824
PANCAKE.__oracle_base='/orabin'#ORACLE_BASE set from environment
PANCAKE.__pga_aggregate_target=268435456
PANCAKE.__sga_target=805306368
PANCAKE.__shared_io_pool_size=29360128
PANCAKE.__shared_pool_size=192937984
PANCAKE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/PANCAKE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='+ASM0/PANCAKE/control01.ctl','+ASM0/PANCAKE/control02.ctl
'
*.control_files='/oradata/pancake/control01.ctl','/logs/pancake/control02.
ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='PANCAKE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PANCAKEXDB)'
*.log_archive_dest_1='LOCATION=+ASM1'
*.log_archive_dest_1='LOCATION=/logs/pancake'
*.log_archive_format='%t_%s_%r.dbf'
'/logs/path/redo02.log'
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/logs/pancake/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/logs/pancake/redo02.log', '+ASM0/PANCAKE/redo03.log',
'/logs/pancake/redo03.log'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

After the new parameters are confirmed, the parameters must be put into effect. Multiple options exist, but most customers create an spfile based on the text pfile.

```
bash-4.1$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Fri Jan 8 11:17:40 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> create spfile from pfile='/rman/pancake/pfile';
File created.
```

Startup nomount

The final step before replicating the database is to bring up the database processes but not mount the files. In this step, problems with the spfile might become evident. If the `startup nomount` command fails because of a parameter error, it is simple to shut down, correct the pfile template, reload it as an spfile, and try again.

```
SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             373296240 bytes
Database Buffers          423624704 bytes
Redo Buffers                5455872 bytes
```

Duplicate the database

Restoring the prior RMAN backup to the new location consumes more time than other steps in this process. The database must be duplicated without a change to the database ID (DBID) or resetting the logs. This prevents logs from being applied, which is a required step to fully synchronize the copies.

Connect to the database with RMAN as aux and issue the duplicate database command by using the script created in a previous step.

```
[oracle@jfsc2 pancake]$ rman auxiliary /
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 03:04:56
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to auxiliary database: PANCAKE (not mounted)
RMAN> run
2> {
3> set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
4> set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
5> set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
6> set newname for datafile 4 to '/oradata/pancake/users.dbf';
7> set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
8> duplicate target database for standby backup location '/rman/pancake';
9> }
```

```
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting Duplicate Db at 24-MAY-16
contents of Memory Script:
{
   restore clone standby controlfile from  '/rman/pancake/ctrl.bkp';
}
executing Memory Script
Starting restore at 24-MAY-16
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file name=/oradata/pancake/control01.ctl
output file name=/logs/pancake/control02.ctl
Finished restore at 24-MAY-16
contents of Memory Script:
{
   sql clone 'alter database mount standby database';
}
executing Memory Script
sql statement: alter database mount standby database
released channel: ORA_AUX_DISK_1
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
contents of Memory Script:
{
   set newname for tempfile  1 to
 "/oradata/pancake/temp.dbf";
   switch clone tempfile all;
   set newname for datafile  1 to
 "/oradata/pancake/pancake.dbf";
   set newname for datafile  2 to
 "/oradata/pancake/sysaux.dbf";
   set newname for datafile  3 to
 "/oradata/pancake/undotbs1.dbf";
   set newname for datafile  4 to
 "/oradata/pancake/users.dbf";
   restore
   clone database
   ;
}
executing Memory Script
```

```
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/pancake/temp.dbf in control file
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting restore at 24-MAY-16
using channel ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/oradata/pancake/pancake.dbf
channel ORA_AUX_DISK_1: restoring datafile 00002 to
/oradata/pancake/sysaux.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/oradata/pancake/undotbs1.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/oradata/pancake/users.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/rman/pancake/1gr6c161_1_1
channel ORA_AUX_DISK_1: piece handle=/rman/pancake/1gr6c161_1_1
tag=ONTAP_MIGRATION
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:07
Finished restore at 24-MAY-16
contents of Memory Script:
{
   switch clone datafile all;
}
executing Memory Script
datafile 1 switched to datafile copy
input datafile copy RECID=5 STAMP=912655725 file
name=/oradata/pancake/pancake.dbf
datafile 2 switched to datafile copy
input datafile copy RECID=6 STAMP=912655725 file
name=/oradata/pancake/sysaux.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=7 STAMP=912655725 file
name=/oradata/pancake/undotbs1.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=8 STAMP=912655725 file
name=/oradata/pancake/users.dbf
Finished Duplicate Db at 24-MAY-16
```

Initial log replication

You must now ship the changes from the source database to a new location. Doing so might require a combination of steps. The simplest method would be to have RMAN on the source database write out archive logs to a shared network connection. If a shared location is not available, an alternative method is using RMAN to write to a local file system and then using rcp or rsync to copy the files.

In this example, the `/rman` directory is an NFS share that is available to both the original and migrated database.

One important issue here is the `disk format` clause. The disk format of the backup is `%h_%e_%a.dbf`, which means that you must use the format of thread number, sequence number, and activation ID for the database. Although the letters are different, this matches the `log_archive_format='%t_%s_%r.dbf` parameter in the pfile. This parameter also specifies archive logs in the format of thread number, sequence number, and activation ID. The end result is that the log file backups on the source use a naming convention that is expected by the database. Doing so makes operations such as `recover database` much simpler because sqlplus correctly anticipates the names of the archive logs to be replayed.

```
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/arch/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN> backup as copy archivelog from time 'sysdate-2';
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=373 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=70 STAMP=912658508
output file name=/rman/pancake/logship/1_54_912576125.dbf RECID=123
STAMP=912659482
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=29 STAMP=912654101
output file name=/rman/pancake/logship/1_41_912576125.dbf RECID=124
STAMP=912659483
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=33 STAMP=912654688
output file name=/rman/pancake/logship/1_45_912576125.dbf RECID=152
STAMP=912659514
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=36 STAMP=912654809
output file name=/rman/pancake/logship/1_47_912576125.dbf RECID=153
STAMP=912659515
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

Initial log replay

After the files are in the archive log location, they can be replayed by issuing the command `recover database until cancel` followed by the response `AUTO` to automatically replay all available logs. The parameter file is currently directing archive logs to `/logs/archive`, but this does not match the location where RMAN was used to save logs. The location can be temporarily redirected as follows before recovering the database.

```
SQL> alter system set log_archive_dest_1='LOCATION=/rman/pancake/logship'
scope=memory;
System altered.
SQL> recover standby database until cancel;
ORA-00279: change 560224 generated at 05/24/2016 03:25:53 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_49_912576125.dbf
ORA-00280: change 560224 for thread 1 is in sequence #49
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 560353 generated at 05/24/2016 03:29:17 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_50_912576125.dbf
ORA-00280: change 560353 for thread 1 is in sequence #50
ORA-00278: log file '/rman/pancake/logship/1_49_912576125.dbf' no longer
needed
for this recovery
...
ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
ORA-00278: log file '/rman/pancake/logship/1_53_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_54_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

The final archive log reply reports an error, but this is normal. The error indicates that sqlplus was seeking a particular log file and did not find it. The reason is most likely that the log file does not yet exist.

If the source database can be shut down before copying archive logs, this step must be performed only once. The archive logs are copied and replayed, and then the process can continue directly to the cutover process that replicates the critical redo logs.

Incremental log replication and replay

In most cases, migration is not performed right away. It could be days or even weeks before the migration process is complete, which means that the logs must be continuously shipped to the replica database and replayed. Doing so makes sure that minimal data must be transferred and replayed when the cutover arrives.

This process can easily be scripted. For example, the following command can be scheduled on the original database to make sure that the location used for log shipping is continuously updated.

```
[oracle@jfsc1 pancake]$ cat copylogs.rman
configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
backup as copy archivelog from time 'sysdate-2';
```

```
[oracle@jfsc1 pancake]$ rman target / cmdfile=copylogs.rman
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 04:36:19
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to target database: PANCAKE (DBID=3574534589)
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=369 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:22
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=124 STAMP=912659483
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:23
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_41_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
```

```
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:55
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:57
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
Recovery Manager complete.
```

After the logs have been received, they must be replayed. Previous examples showed the use of sqlplus to manually run `recover database until cancel`, which can be easily automated. The example shown here uses the script described in Replay Logs on Standby Database. The script accepts an argument that specifies the database requiring a replay operation. This process permits the same script to be used in a multidatabase migration effort.

```
[root@jfsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 04:47:10 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 562219 generated at 05/24/2016 04:15:08 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_55_912576125.dbf
ORA-00280: change 562219 for thread 1 is in sequence #55
ORA-00278: log file '/rman/pancake/logship/1_54_912576125.dbf' no longer
needed for this recovery
ORA-00279: change 562370 generated at 05/24/2016 04:19:18 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_56_912576125.dbf
ORA-00280: change 562370 for thread 1 is in sequence #56
ORA-00278: log file '/rman/pancake/logship/1_55_912576125.dbf' no longer
needed for this recovery
...
ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
ORA-00278: log file '/rman/pancake/logship/1_64_912576125.dbf' no longer
needed for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_65_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

Cutover

When you are ready to cut over to the new environment, you must perform one final synchronization. When working with regular file systems, it is easy to make sure that the migrated database is 100% synchronized against the original because the original redo logs are copied and replayed. There is no good way to do this with ASM. Only the archive logs can be easily recopied. To make sure that no data is lost, the final shutdown of the original database must be performed carefully.

1. First, the database must be quiesced, ensuring that no changes are being made. This quiescing might include disabling scheduled operations, shutting down listeners, and/or shutting down applications.
2. After this step is taken, most DBAs create a dummy table to serve as a marker of the shutdown.
3. Force a log archiving to make sure that the creation of the dummy table is recorded within the archive logs. To do so, run the following commands:

```
SQL> create table cutovercheck as select * from dba_users;
Table created.
SQL> alter system archive log current;
System altered.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

4. To copy the last of the archive logs, run the following commands. The database must be available but not open.

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             331353200 bytes
Database Buffers          465567744 bytes
Redo Buffers                5455872 bytes
Database mounted.
```

5. To copy the archive logs, run the following commands:

```
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=8 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:24
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:58
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:59:00
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
```

6. Finally, replay the remaining archive logs on the new server.

```
[root@jfsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 05:00:53 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed
for thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 563629 generated at 05/24/2016 04:55:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_66_912576125.dbf
ORA-00280: change 563629 for thread 1 is in sequence #66
ORA-00278: log file '/rman/pancake/logship/1_65_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_66_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

7. At this stage, replicate all data. The database is ready to be converted from a standby database to an active operational database and then opened.

```
SQL> alter database activate standby database;
Database altered.
SQL> alter database open;
Database altered.
```

8. Confirm the presence of the dummy table and then drop it.

```
SQL> desc cutovercheck
 Name                                      Null?    Type
 ----------------------------------------- --------
----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 USER_ID                                   NOT NULL NUMBER
 PASSWORD                                           VARCHAR2(4000)
 ACCOUNT_STATUS                            NOT NULL VARCHAR2(32)
 LOCK_DATE                                          DATE
 EXPIRY_DATE                                        DATE
 DEFAULT_TABLESPACE                        NOT NULL VARCHAR2(30)
 TEMPORARY_TABLESPACE                      NOT NULL VARCHAR2(30)
 CREATED                                   NOT NULL DATE
 PROFILE                                   NOT NULL VARCHAR2(128)
 INITIAL_RSRC_CONSUMER_GROUP                        VARCHAR2(128)
 EXTERNAL_NAME                                      VARCHAR2(4000)
 PASSWORD_VERSIONS                                  VARCHAR2(12)
 EDITIONS_ENABLED                                   VARCHAR2(1)
 AUTHENTICATION_TYPE                                VARCHAR2(8)
 PROXY_ONLY_CONNECT                                 VARCHAR2(1)
 COMMON                                             VARCHAR2(3)
 LAST_LOGIN                                         TIMESTAMP(9) WITH
TIME ZONE
 ORACLE_MAINTAINED                                  VARCHAR2(1)
SQL> drop table cutovercheck;
Table dropped.
```

Nondisruptive redo log migration

There are times when a database is correctly organized overall with the exception of the redo logs. This can happen for many reasons, the most common of which is related to snapshots. Products such as SnapManager for Oracle, SnapCenter, and the NetApp Snap Creator storage management framework enable near-instantaneous recovery of a database, but only if you revert the state of the data file volumes. If redo logs share space with the data files, reversion cannot be performed safely because it would result in destruction of the redo logs, likely meaning data loss. Therefore, the redo logs must be relocated.

This procedure is simple and can be performed nondisruptively.

Current redo log configuration

1. Identify the number of redo log groups and their respective group numbers.

```
SQL> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
--------------------------------------------------------------------------------
--------
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
rows selected.
```

2. Enter the size of the redo logs.

```
SQL> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
--------------------------------------------------------------------------------
--------
1 524288000
2 524288000
3 524288000
```

Create new logs

1. For each redo log, create a new group with a matching size and number of members.

```
SQL> alter database add logfile ('/newredo0/redo01a.log',
'/newredo1/redo01b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo02a.log',
'/newredo1/redo02b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo03a.log',
'/newredo1/redo03b.log') size 500M;
Database altered.
SQL>
```

2. Verify the new configuration.

```
SQL> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
--------------------------------------------------------------------------------
--------
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
4 /newredo0/redo01a.log
4 /newredo1/redo01b.log
5 /newredo0/redo02a.log
5 /newredo1/redo02b.log
6 /newredo0/redo03a.log
6 /newredo1/redo03b.log
12 rows selected.
```

Drop old logs

1. Drop the old logs (groups 1, 2, and 3).

```
SQL> alter database drop logfile group 1;
Database altered.
SQL> alter database drop logfile group 2;
Database altered.
SQL> alter database drop logfile group 3;
Database altered.
```

2. If you encounter an error that prevents you from dropping an active log, force a switch to the next log to release the lock and force a global checkpoint. See the following example of this process. The attempt to drop logfile group 2, which was located on the old location, was denied because there was still active data in this logfile.

```
SQL> alter database drop logfile group 2;
alter database drop logfile group 2
*
ERROR at line 1:
ORA-01623: log 2 is current log for instance NTAP (thread 1) - cannot
drop
ORA-00312: online log 2 thread 1: '/redo0/NTAP/redo02a.log'
ORA-00312: online log 2 thread 1: '/redo1/NTAP/redo02b.log'
```

3. A log archiving followed by a checkpoint enables you to drop the logfile.

```
SQL> alter system archive log current;
System altered.
SQL> alter system checkpoint;
System altered.
SQL> alter database drop logfile group 2;
Database altered.
```

4. Then delete the logs from the file system. You should perform this process with extreme care.

Host data copying

As with database-level migration, migration at the host layer provides a storage vendor–independent approach.

In other words, sometime "just copy the files" is the best option.

Although this low-tech approach might seem too basic, it does offer significant benefits because no special software is required and the original data remains safely untouched during the process. The primary limitation is the fact that a file-copy data migration is a disruptive process, because the database must be shut down before the copy operation begins. There is no good way to synchronize changes within a file, so the files must be completely quiesced before copying begins.

If the shutdown required by a copy operation is not desirable, the next best host-based option is leveraging a logical volume manager (LVM). Many LVM options exist, including Oracle ASM, all with similar capabilities, but also with some limitations that must be taken into account. In most cases, the migration can be accomplished without downtime and disruption.

Filesystem to filesystem copying

The usefulness of a simple copy operation should not be underestimated. This operation requires downtime during the copy process, but it is a highly reliable process and requires no special expertise with operating systems, databases, or storage systems. Furthermore, it is very safe because it does not affect the original data. Typically, a system administrator changes the source file systems to be mounted as read-only and then reboots a server to guarantee that nothing can damage the current data. The copy process can be scripted to make sure that it runs as quickly as possible without risk of user error. Because the type of I/O is a simple sequential transfer of data, it is highly bandwidth efficient.

The following example demonstrates one option for a safe and rapid migration.

Environment

The environment to be migrated is as follows:

- Current file systems

```
ontap-nfs1:/host1_oradata       52428800  16196928  36231872  31%
/oradata
ontap-nfs1:/host1_logs          49807360    548032  49259328   2% /logs
```

- New file systems

```
ontap-nfs1:/host1_logs_new      49807360       128  49807232   1%
/new/logs
ontap-nfs1:/host1_oradata_new   49807360       128  49807232   1%
/new/oradata
```

Overview

The database can be migrated by a DBA by simply shutting down the database and copying the files, but the process is easily scripted if many databases must be migrated or minimizing downtime is critical. The use of scripts also reduces the chance for user error.

The example scripts shown automate the following operations:

- Shutting down the database
- Converting the existing file systems to a read-only state
- Copying all data from the source to target file systems, which preserves all file permissions
- Unmounting the old and new file systems
- Remounting the new file systems at the same paths as the prior file systems

Procedure

1. Shut down the database.

```
[root@host1 current]# ./dbshut.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 15:58:48 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
```

2. Convert the file systems to read-only. This can be done more quickly by using a script, as shown in Convert File System to Read Only.

```
[root@host1 current]# ./mk.fs.readonly.pl /oradata
/oradata unmounted
/oradata mounted read-only
[root@host1 current]# ./mk.fs.readonly.pl /logs
/logs unmounted
/logs mounted read-only
```

3. Confirm that the file systems are now read-only.

```
ontap-nfs1:/host1_oradata on /oradata type nfs
(ro,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
```

4. Synchronize file system contents with the `rsync` command.

```
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /oradata/ /new/oradata/
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
```

```
 10737426432 100%  153.50MB/s    0:01:06 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
    22823573 100%   12.09MB/s    0:00:01 (xfer#2, to-check=9/13)
...
NTAP/undotbs02.dbf
  1073750016 100%  131.60MB/s    0:00:07 (xfer#10, to-check=1/13)
NTAP/users01.dbf
     5251072 100%    3.95MB/s    0:00:01 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes  received 228 bytes  162204017.96 bytes/sec
total size is 18570092218  speedup is 1.00
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /logs/ /new/logs/
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%   95.98MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%   49.45MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%   44.68MB/s    0:00:01 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%   68.03MB/s    0:00:00 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
```

```
sent 527098156 bytes  received 278 bytes  95836078.91 bytes/sec
total size is 527032832  speedup is 1.00
```

5. Unmount the old file systems and relocate the copied data. This can be done more quickly by using a script, as shown in Replace File System.

```
[root@host1 current]# ./swap.fs.pl /logs,/new/logs
/new/logs unmounted
/logs unmounted
Updated /logs mounted
[root@host1 current]# ./swap.fs.pl /oradata,/new/oradata
/new/oradata unmounted
/oradata unmounted
Updated /oradata mounted
```

6. Confirm that the new file systems are in position.

```
ontap-nfs1:/host1_logs_new on /logs type nfs
(rw,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
```

7. Start the database.

```
[root@host1 current]# ./dbstart.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 16:10:07 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             390073456 bytes
Database Buffers          406847488 bytes
Redo Buffers                5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
```

Fully automated cutover

This sample script accepts arguments of the database SID followed by common-delimited pairs of file systems. For the example shown above, the command is issued as follows:

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
```

When executed, the example script attempts to perform the following sequence. It terminates if it encounters an error in any step:

1. Shut down the database.
2. Convert the current file systems to read-only status.
3. Use each comma-delimited pair of file system arguments and synchronize the first file system to the second.
4. Dismount the prior file systems.
5. Update the `/etc/fstab` file as follows:
    a. Create a backup at `/etc/fstab.bak`.
    b. Comment out the prior entries for the prior and new file systems.
    c. Create a new entry for the new file system that uses the old mountpoint.
6. Mount the file systems.
7. Start the database.

The following text provides an execution example for this script:

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:05:50 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
sending incremental file list
```

```
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%  185.40MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%   81.34MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%   70.42MB/s    0:00:00 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%   47.08MB/s    0:00:01 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes  received 278 bytes  150599552.57 bytes/sec
total size is 527032832  speedup is 1.00
Succesfully replicated filesystem /logs to /new/logs
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
 10737426432 100%  176.55MB/s    0:00:58 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
    22823573 100%    9.48MB/s    0:00:02 (xfer#2, to-check=9/13)
... NTAP/undotbs01.dbf
   309338112 100%   70.76MB/s    0:00:04 (xfer#9, to-check=2/13)
NTAP/undotbs02.dbf
  1073750016 100%  187.65MB/s    0:00:05 (xfer#10, to-check=1/13)
NTAP/users01.dbf
     5251072 100%    5.09MB/s    0:00:00 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
```

```
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes  received 228 bytes  177725933.55 bytes/sec
total size is 18570092218  speedup is 1.00
Succesfully replicated filesystem /oradata to /new/oradata
swap 0 /logs /new/logs
/new/logs unmounted
/logs unmounted
Mounted updated /logs
Swapped filesystem /logs for /new/logs
swap 1 /oradata /new/oradata
/new/oradata unmounted
/oradata unmounted
Mounted updated /oradata
Swapped filesystem /oradata for /new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:08:59 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             390073456 bytes
Database Buffers          406847488 bytes
Redo Buffers                5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
[root@host1 current]#
```

Oracle ASM spfile and passwd migration

One difficulty in completing migration involving ASM is the ASM-specific spfile and the password file. By default, these critical metadata files are created on the first ASM disk group defined. If a particular ASM disk group must be evacuated and removed, the spfile and password file that govern that ASM instance must be relocated.

Another use case in which these files might need to be relocated is during a deployment of database management software such as SnapManager for Oracle or the SnapCenter Oracle plug- in. One of the features of these products is to rapidly restore a database by reverting the state of the ASM LUNs hosting the data files. Doing so requires taking the ASM disk group offline before performing a restore. This is not a problem as long as a given database's data files are isolated in a dedicated ASM disk group.

When that disk group also contains the ASM spfile/passwd file, the only way the disk group can be brought offline is to shut down the entire ASM instance. This is a disruptive process, which means that the spfile/passwd file would need to be relocated.

Environment

1. Database SID = TOAST
2. Current data files on `+DATA`
3. Current logfiles and controlfiles on `+LOGS`
4. New ASM disk groups established as `+NEWDATA` and `+NEWLOGS`

ASM spfile/passwd file locations

Relocating these files can be done nondisruptively. However, for safety, NetApp recommends shutting down the database environment so that you can be certain that the files have been relocated and the configuration is properly updated. This procedure must be repeated if multiple ASM instances are present on a server.

Identify ASM instances

Identify the ASM instances based on the data recorded in the `oratab` file. The ASM instances are denoted by a + symbol.

```
-bash-4.1$ cat /etc/oratab | grep '^+'
+ASM:/orabin/grid:N             # line added by Agent
```

There is one ASM instance called +ASM on this server.

Make sure all databases are shut down

The only smon process visible should be the smon for the ASM instance in use. The presence of another smon process indicates that a database is still running.

```
-bash-4.1$ ps -ef | grep smon
oracle     857     1   0 18:26 ?        00:00:00 asm_smon_+ASM
```

The only smon process is the ASM instance itself. This means that no other databases are running, and it is safe to proceed without risk of disrupting database operations.

Locate files

Identify the current location of the ASM spfile and password file by using the `spget` and `pwget` commands.

```
bash-4.1$ asmcmd
ASMCMD> spget
+DATA/spfile.ora
```

```
ASMCMD> pwget --asm
+DATA/orapwasm
```

The files are both located at the base of the `+DATA` disk group.

Copy files

Copy the files to the new ASM disk group with the `spcopy` and `pwcopy` commands. If the new disk group was recently created and is currently empty, it might need to be mounted first.

```
ASMCMD> mount NEWDATA
```

```
ASMCMD> spcopy +DATA/spfile.ora +NEWDATA/spfile.ora
copying +DATA/spfile.ora -> +NEWDATA/spfilea.ora
```

```
ASMCMD> pwcopy +DATA/orapwasm +NEWDATA/orapwasm
copying +DATA/orapwasm -> +NEWDATA/orapwasm
```

The files have now been copied from `+DATA` to `+NEWDATA`.

Update ASM instance

The ASM instance must now be updated to reflect the change in location. The `spset` and `pwset` commands update the ASM metadata required for starting the ASM disk group.

```
ASMCMD> spset +NEWDATA/spfile.ora
ASMCMD> pwset --asm +NEWDATA/orapwasm
```

Activate ASM using updated files

At this point, the ASM instance still uses the prior locations of these files. The instance must be restarted to force a reread of the files from their new locations and to release locks on the prior files.

```
-bash-4.1$ sqlplus / as sysasm
SQL> shutdown immediate;
ASM diskgroups volume disabled
ASM diskgroups dismounted
ASM instance shutdown
```

```
SQL> startup
ASM instance started
Total System Global Area 1140850688 bytes
Fixed Size                  2933400 bytes
Variable Size            1112751464 bytes
ASM Cache                  25165824 bytes
ORA-15032: not all alterations performed
ORA-15017: diskgroup "NEWDATA" cannot be mounted
ORA-15013: diskgroup "NEWDATA" is already mounted
```

Remove old spfile and password files

If the procedure has been performed successfully, the prior files are no longer locked and can now be removed.

```
-bash-4.1$ asmcmd
ASMCMD> rm +DATA/spfile.ora
ASMCMD> rm +DATA/orapwasm
```

Oracle ASM to ASM copy

Oracle ASM is essentially a lightweight combined volume manager and file system. Because the file system is not readily visible, RMAN must be used to perform copy operations. Although a copy-based migration process is safe and simple, it results in some disruption. The disruption can be minimized, but not fully eliminated.

If you want nondisruptive migration of an ASM-based database, the best option is to leverage ASM's capability to rebalance ASM extents to new LUNs while dropping the old LUNs. Doing so is generally safe and nondisruptive to operations, but it offers no back- out path. If functional or performance problems are encountered, the only option is to migrate the data back to the source.

This risk can be avoided by copying the database to the new location rather than moving data, so that the original data is untouched. The database can be fully tested in its new location before going live, and the original database is available as a fall- back option if problems are found.

This procedure is one of many options involving RMAN. It is designed to allow a two-step process in which the initial backup is created and then later synchronized through log replay. This process is desirable to minimize downtime because it allows the database to remain operational and serving data during the initial baseline copy.

Copy database

Oracle RMAN creates a level 0 (complete) copy of the source database currently located on the ASM disk group `+DATA` to the new location on `+NEWDATA`.

```
-bash-4.1$ rman target /
Recovery Manager: Release 12.1.0.2.0 - Production on Sun Dec 6 17:40:03
2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to target database: TOAST (DBID=2084313411)
RMAN> backup as copy incremental level 0 database format '+NEWDATA' tag
'ONTAP_MIGRATION';
Starting backup at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=302 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
...
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
output file name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
tag=ONTAP_MIGRATION RECID=5 STAMP=897759622
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWDATA/TOAST/BACKUPSET/2015_12_06/nnsnn0_ontap_migration_0.262.89
7759623 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

Force archive log switch

You must force an archive log switch to make sure that the archive logs contain all data required to make the copy fully consistent. Without this command, key data might still be present in the redo logs.

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

Shut down source database

Disruption begins in this step because the database is shut down and placed in a limited-access, read-only mode. To shut down the source database, run the following commands:

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                390073456 bytes
Database Buffers             406847488 bytes
Redo Buffers                   5455872 bytes
```

Controlfile backup

You must back up the controlfile in case you must abort the migration and revert to the original storage location. A copy of the backup controlfile isn't 100% required, but it does make the process of resetting the database file locations back to the original location easier.

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=358 device type=DISK
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151206T174753 RECID=6
STAMP=897760073
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

Parameter updates

The current spfile contains references to the controlfiles on their current locations within the old ASM disk group. It must be edited, which is easily done by editing an intermediate pfile version.

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

Update pfile

Update any parameters referring to old ASM disk groups to reflect the new ASM disk group names. Then save the updated pfile. Make sure that the `db_create` parameters are present.

In the example below, the references to `+DATA` that were changed to `+NEWDATA` are highlighted in yellow. Two key parameters are the `db_create` parameters that create any new files at the correct location.

```
*.compatible='12.1.0.2.0'
*.control_files='+NEWLOGS/TOAST/CONTROLFILE/current.258.897683139'
*.db_block_size=8192
*. db_create_file_dest='+NEWDATA'
*. db_create_online_log_dest_1='+NEWLOGS'
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB)'
*.log_archive_dest_1='LOCATION=+NEWLOGS'
*.log_archive_format='%t_%s_%r.dbf'
```

Update init.ora file

Most ASM-based databases use an `init.ora` file located in the `$ORACLE_HOME/dbs` directory, which is a point to the spfile on the ASM disk group. This file must be redirected to a location on the new ASM disk group.

```
-bash-4.1$ cd $ORACLE_HOME/dbs
-bash-4.1$ cat initTOAST.ora
SPFILE='+DATA/TOAST/spfileTOAST.ora'
```

Change this file as follows:

```
SPFILE=+NEWLOGS/TOAST/spfileTOAST.ora
```

Parameter file recreation

The spfile is now ready to be populated by the data in the edited pfile.

```
RMAN> create spfile from pfile='/tmp/pfile';
Statement processed
```

Start database to start using new spfile

Start the database to make sure that it now uses the newly created spfile and that any further changes to system parameters are correctly recorded.

```
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                373296240 bytes
Database Buffers             423624704 bytes
Redo Buffers                   5455872 bytes
```

Restore controlfile

The backup controlfile created by RMAN can also be restored by RMAN directly to the location specified in the new spfile.

```
RMAN> restore controlfile from
'+DATA/TOAST/CONTROLFILE/current.258.897683139';
Starting restore at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=417 device type=DISK
channel ORA_DISK_1: copied control file copy
output file name=+NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
Finished restore at 06-DEC-15
```

Mount the database and verify the use of the new controlfile.

```
RMAN> alter database mount;
using target database control file instead of recovery catalog
Statement processed
```

```
SQL> show parameter control_files;
NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
control_files                        string
+NEWLOGS/TOAST/CONTROLFILE/cur
                                                 rent.273.897761061
```

Log replay

The database currently uses the data files in the old location. Before the copy can be used, they must be synchronized. Time has passed during the initial copy process, and the changes have been logged primarily in the archive logs. These changes are replicated as follows:

1. Perform an RMAN incremental backup, which contains the archive logs.

```
RMAN> backup incremental level 1 format '+NEWLOGS' for recover of copy
with tag 'ONTAP_MIGRATION' database;
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=62 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
input datafile file number=00002
name=+DATA/TOAST/DATAFILE/sysaux.260.897683143
input datafile file number=00003
name=+DATA/TOAST/DATAFILE/undotbs1.257.897683145
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/ncsnn1_ontap_migration_0.267.
897762697 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

2. Replay the log.

```
RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 06-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001
name=+NEWDATA/TOAST/DATAFILE/system.259.897759609
recovering datafile copy file number=00002
name=+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
recovering datafile copy file number=00003
name=+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
recovering datafile copy file number=00004
name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
channel ORA_DISK_1: reading from backup piece
+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.8977626
93
channel ORA_DISK_1: piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15
```

Activation

The controlfile that was restored still references the data files at the original location, and it also contains the path information for the copied data files.

1. To change the active data files, run the `switch database to copy` command.

```
RMAN> switch database to copy;
datafile 1 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/system.259.897759609"
datafile 2 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615"
datafile 3 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619"
datafile 4 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/users.258.897759623"
```

   The active data files are now the copied data files, but there still might be changes contained within the final redo logs.

2. To replay all of the remaining logs, run the `recover database` command. If the message `media recovery complete` appears, the process was successful.

```
RMAN> recover database;
Starting recover at 06-DEC-15
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15
```

This process only changed the location of the normal data files. The temporary data files must be renamed, but they do not need to be copied because they are only temporary. The database is currently down, so there is no active data in the temporary data files.

3. To relocate the temporary data files, first identify their location.

```
RMAN> select file#||' '||name from v$tempfile;
FILE#||''||NAME
-------------------------------------------------------------------------------
--------
1 +DATA/TOAST/TEMPFILE/temp.263.897683145
```

4. Relocate temporary data files by using an RMAN command that sets the new name for each data file. With Oracle Managed Files (OMF), the complete name is not necessary; the ASM disk group is sufficient. When the database is opened, OMF links to the appropriate location on the ASM disk group. To relocate files, run the following commands:

```
run {
set newname for tempfile 1 to '+NEWDATA';
switch tempfile all;
}
```

```
RMAN> run {
2> set newname for tempfile 1 to '+NEWDATA';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to +NEWDATA in control file
```

Redo log migration

The migration process is nearly complete, but the redo logs are still located on the original ASM disk group. Redo logs cannot be directly relocated. Instead, a new set of redo logs is created and added to the configuration, followed by a drop of the old logs.

1. Identify the number of redo log groups and their respective group numbers.

```
RMAN> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
------------------------------------------------------------------------------
--------
1 +DATA/TOAST/ONLINELOG/group_1.261.897683139
2 +DATA/TOAST/ONLINELOG/group_2.259.897683139
3 +DATA/TOAST/ONLINELOG/group_3.256.897683139
```

2. Enter the size of the redo logs.

```
RMAN> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
------------------------------------------------------------------------------
--------
1 52428800
2 52428800
3 52428800
```

3. For each redo log, create a new group with a matching configuration. If you are not using OMF, you must specify the full path. This is also an example that uses the `db_create_online_log` parameters. As was shown previously, this parameter was set to +NEWLOGS. This configuration allows you to use the following commands to create new online logs without the need to specify a file location or even a specific ASM disk group.

```
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
```

4. Open the database.

```
SQL> alter database open;
Database altered.
```

5. Drop the old logs.

```
RMAN> alter database drop logfile group 1;
Statement processed
```

6. If you encounter an error that prevents you from dropping an active log, force a switch to the next log to

release the lock and force a global checkpoint. An example is shown below. The attempt to drop logfile group 3, which was located on the old location, was denied because there was still active data in this logfile. A log archiving following a checkpoint allows you to delete the logfile.

```
RMAN> alter database drop logfile group 3;
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 3 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 3 thread 1:
'+LOGS/TOAST/ONLINELOG/group_3.259.897563549'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 3;
Statement processed
```

7. Review the environment to make sure that all location-based parameters are updated.

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

8. The following script demonstrates how to simplify this process:

```
[root@host1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
+NEWDATA/TOAST/DATAFILE/system.259.897759609
+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
+NEWDATA/TOAST/DATAFILE/users.258.897759623
TOAST redo logs:
+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123
+NEWLOGS/TOAST/ONLINELOG/group_5.265.897763125
+NEWLOGS/TOAST/ONLINELOG/group_6.264.897763125
TOAST temp datafiles:
+NEWDATA/TOAST/TEMPFILE/temp.260.897763165
TOAST spfile
spfile                                string
+NEWDATA/spfiletoast.ora
TOAST key parameters
control_files +NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
log_archive_dest_1 LOCATION=+NEWLOGS
db_create_file_dest +NEWDATA
db_create_online_log_dest_1 +NEWLOGS
```

9. If the ASM disk groups were completely evacuated, they can now be unmounted with `asmcmd`. However, in many cases the files belonging to other databases or the ASM spfile/passwd file might still be present.

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>
```

Oracle ASM to file system copy

The Oracle ASM to file system copy procedure is very similar to the ASM to ASM copy procedure, with similar benefits and restrictions. The primary difference is the syntax of the various commands and configuration parameters when using a visible file system as opposed to an ASM disk group.

Copy database

Oracle RMAN is used to create a level 0 (complete) copy of the source database currently located on the ASM disk group `+DATA` to the new location on `/oradata`.

```
RMAN> backup as copy incremental level 0 database format
'/oradata/TOAST/%U' tag 'ONTAP_MIGRATION';
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=377 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-
1_01r5fhjg tag=ONTAP_MIGRATION RECID=1 STAMP=911722099
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-
2_02r5fhjo tag=ONTAP_MIGRATION RECID=2 STAMP=911722106
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt tag=ONTAP_MIGRATION RECID=3 STAMP=911722113
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/oradata/TOAST/cf_D-TOAST_id-2098173325_04r5fhk5
tag=ONTAP_MIGRATION RECID=4 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-
4_05r5fhk6 tag=ONTAP_MIGRATION RECID=5 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/oradata/TOAST/06r5fhk7_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
```

Force archive log switch

Forcing the archive log switch is required to make sure that the archive logs contain all of the data required to make the copy fully consistent. Without this command, key data might still be present in the redo logs. To force an archive log switch, run the following command:

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

Shut down source database

Disruption begins in this step because the database is shut down and placed in a limited-access read-only mode. To shut down the source database, run the following commands:

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                331353200 bytes
Database Buffers             465567744 bytes
Redo Buffers                   5455872 bytes
```

Controlfile backup

Back up controlfiles in case you must abort the migration and revert to the original storage location. A copy of the backup controlfile isn't 100% required, but it does make the process of resetting the database file locations back to the original location easier.

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 08-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151208T194540 RECID=30
STAMP=897939940
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 08-DEC-15
```

Parameter updates

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

Update pfile

Any parameters referring to old ASM disk groups should be updated and, in some cases, deleted when they are no longer relevant. Update them to reflect the new file system paths and save the updated pfile. Make sure that the complete target path is listed. To update these parameters, run the following commands:

```
*.audit_file_dest='/orabin/admin/TOAST/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/logs/TOAST/arch/control01.ctl','/logs/TOAST/redo/control
02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB)'
*.log_archive_dest_1='LOCATION=/logs/TOAST/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

Disable the original init.ora file

This file is located in the `$ORACLE_HOME/dbs` directory and is usually in a pfile that serves as a pointer to the spfile on the ASM disk group. To make sure that the original spfile is no longer used, rename it. Do not delete it, however, because this file is needed if the migration must be aborted.

```
[oracle@jfsc1 ~]$ cd $ORACLE_HOME/dbs
[oracle@jfsc1 dbs]$ cat initTOAST.ora
SPFILE='+ASM0/TOAST/spfileTOAST.ora'
[oracle@jfsc1 dbs]$ mv initTOAST.ora initTOAST.ora.prev
[oracle@jfsc1 dbs]$
```

Parameter file recreation

This is the final step in spfile relocation. The original spfile is no longer used and the database is currently started (but not mounted) using the intermediate file. The contents of this file can be written out to the new spfile location as follows:

```
RMAN> create spfile from pfile='/tmp/pfile';
Statement processed
```

Start database to start using new spfile

You must start the database to release the locks on the intermediate file and start the database by using only the new spfile file. Starting the database also proves that the new spfile location is correct and its data is valid.

```
RMAN> shutdown immediate;
Oracle instance shut down
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                331353200 bytes
Database Buffers             465567744 bytes
Redo Buffers                   5455872 bytes
```

Restore controlfile

A backup controlfile was created at the path `/tmp/TOAST.ctrl` earlier in the procedure. The new spfile defines the controlfile locations as /`logfs/TOAST/ctrl/ctrlfile1.ctrl` and `/logfs/TOAST/redo/ctrlfile2.ctrl`. However, those files do not yet exist.

1. This command restores the controlfile data to the paths defined in the spfile.

   ```
   RMAN> restore controlfile from '/tmp/TOAST.ctrl';
   Starting restore at 13-MAY-16
   using channel ORA_DISK_1
   channel ORA_DISK_1: copied control file copy
   output file name=/logs/TOAST/arch/control01.ctl
   output file name=/logs/TOAST/redo/control02.ctl
   Finished restore at 13-MAY-16
   ```

2. Issue the mount command so that the controlfiles are discovered correctly and contain valid data.

   ```
   RMAN> alter database mount;
   Statement processed
   released channel: ORA_DISK_1
   ```

   To validate the `control_files` parameter, run the following command:

```
SQL> show parameter control_files;
NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
control_files                        string
/logs/TOAST/arch/control01.ctl
                                                 ,
/logs/TOAST/redo/control02.c
                                                 tl
```

Log replay

The database is currently using the data files in the old location. Before the copy can be used, the data files must be synchronized. Time has passed during the initial copy process, and changes were logged primarily in the archive logs. These changes are replicated in the following two steps.

1. Perform an RMAN incremental backup, which contains the archive logs.

```
RMAN>  backup incremental level 1 format '/logs/TOAST/arch/%U' for
recover of copy with tag 'ONTAP_MIGRATION' database;
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=124 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/logs/TOAST/arch/09r5fj8i_1_1 tag=ONTAP_MIGRATION
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

2. Replay the logs.

```
RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 13-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
recovering datafile copy file number=00002 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
recovering datafile copy file number=00003 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
recovering datafile copy file number=00004 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
channel ORA_DISK_1: reading from backup piece
/logs/TOAST/arch/09r5fj8i_1_1
channel ORA_DISK_1: piece handle=/logs/TOAST/arch/09r5fj8i_1_1
tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

Activation

The controlfile that was restored still references the data files at the original location, and it also contains the path information for the copied data files.

1. To change the active data files, run the `switch database to copy` command:

```
RMAN> switch database to copy;
datafile 1 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSTEM_FNO-1_01r5fhjg"
datafile 2 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSAUX_FNO-2_02r5fhjo"
datafile 3 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt"
datafile 4 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-USERS_FNO-4_05r5fhk6"
```

2. Although the data files should be fully consistent, one final step is required to replay the remaining changes recorded in the online redo logs. Use the `recover database` command to replay these changes and make the copy 100% identical to the original. The copy is not yet open, however.

```
RMAN> recover database;
Starting recover at 13-MAY-16
using channel ORA_DISK_1
starting media recovery
archived log for thread 1 with sequence 28 is already on disk as file
+ASM0/TOAST/redo01.log
archived log file name=+ASM0/TOAST/redo01.log thread=1 sequence=28
media recovery complete, elapsed time: 00:00:00
Finished recover at 13-MAY-16
```

Relocate Temporary Data Files

1. Identify the location of temporary data files still in use on the original disk group.

```
RMAN> select file#||' '||name from v$tempfile;
FILE#||''||NAME
-------------------------------------------------------------------------
--------
1 +ASM0/TOAST/temp01.dbf
```

2. To relocate the data files, run the following commands. If there are many tempfiles, use a text editor to create the RMAN command and then cut and paste it.

```
RMAN> run {
2> set newname for tempfile 1 to '/oradata/TOAST/temp01.dbf';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/TOAST/temp01.dbf in control file
```

Redo log migration

The migration process is nearly complete, but the redo logs are still located on the original ASM disk group. Redo logs cannot be directly relocated. Instead, a new set of redo logs is created and added to the configuration, following by a drop of the old logs.

1. Identify the number of redo log groups and their respective group numbers.

```
RMAN> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
------------------------------------------------------------------------------
--------
1 +ASM0/TOAST/redo01.log
2 +ASM0/TOAST/redo02.log
3 +ASM0/TOAST/redo03.log
```

2. Enter the size of the redo logs.

```
RMAN> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
------------------------------------------------------------------------------
--------
1 52428800
2 52428800
3 52428800
```

3. For each redo log, create a new group by using the same size as the current redo log group using the new file system location.

```
RMAN> alter database add logfile '/logs/TOAST/redo/log00.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log01.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log02.rdo' size
52428800;
Statement processed
```

4. Remove the old logfile groups that are still located on the prior storage.

```
RMAN> alter database drop logfile group 4;
Statement processed
RMAN> alter database drop logfile group 5;
Statement processed
RMAN> alter database drop logfile group 6;
Statement processed
```

5. If an error is encountered that blocks dropping an active log, force a switch to the next log to release the lock and force a global checkpoint. An example is shown below. The attempt to drop logfile group 3, which was located on the old location, was denied because there was still active data in this logfile. A log

archiving followed by a checkpoint enables logfile deletion.

```
RMAN> alter database drop logfile group 4;
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 4 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 4 thread 1:
'+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 4;
Statement processed
```

6. Review the environment to make sure that all location-based parameters are updated.

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

7. The following script demonstrates how to make this process easier.

```
[root@jfsc1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
TOAST redo logs:
/logs/TOAST/redo/log00.rdo
/logs/TOAST/redo/log01.rdo
/logs/TOAST/redo/log02.rdo
TOAST temp datafiles:
/oradata/TOAST/temp01.dbf
TOAST spfile
spfile                                  string
/orabin/product/12.1.0/dbhome_
                                                        1/dbs/spfileTOAST.ora
TOAST key parameters
control_files /logs/TOAST/arch/control01.ctl,
/logs/TOAST/redo/control02.ctl
log_archive_dest_1 LOCATION=/logs/TOAST/arch
```

8. If the ASM disk groups were completely evacuated, they can now be unmounted with `asmcmd`. In many cases, files belonging to other databases or the ASM spfile/passwd file can still be present.

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>
```

Data file cleanup procedure

The migration process might result in data files with long or cryptic syntax, depending on how Oracle RMAN was used. In the example shown here, the backup was performed with the file format of `/oradata/TOAST/%U`. `%U` indicates that RMAN should create a default unique name for each data file. The result is similar to what is shown in the following text. The traditional names for the data files are embedded within the names. This can be cleaned up by using the scripted approach shown in ASM Migration Cleanup.

```
[root@jfsc1 current]# ./fixuniquenames.pl TOAST
#sqlplus Commands
shutdown immediate;
startup mount;
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/system.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/sysaux.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt /oradata/TOAST/undotbs1.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
/oradata/TOAST/users.dbf
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSTEM_FNO-1_01r5fhjg' to '/oradata/TOAST/system.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSAUX_FNO-2_02r5fhjo' to '/oradata/TOAST/sysaux.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
UNDOTBS1_FNO-3_03r5fhjt' to '/oradata/TOAST/undotbs1.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
USERS_FNO-4_05r5fhk6' to '/oradata/TOAST/users.dbf';
alter database open;
```

Oracle ASM rebalance

As discussed previously, an Oracle ASM disk group can be transparently migrated to a new storage system by using the rebalancing process. In summary, the rebalancing process requires the addition of equal-sized LUNs to the existing group of LUNs followed by a drop operation of the prior LUN. Oracle ASM automatically relocates the underlying data to new storage in an optimal layout and then releases the old LUNs when complete.

The migration process uses efficient sequential I/O and does not generally cause any performance disruption, but the migration rate can be throttled when needed.

Identify data to be migrated

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
SQL> select group_number||' '||name from v$asm_diskgroup;
1 NEWDATA
```

Create new LUNs

Create new LUNs of the same size, and set user and group membership as required. The LUNs should appear as `CANDIDATE` disks.

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
 0 0 /dev/mapper/3600a098038303537762b47594c31586b CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c315869 CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c315858 CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c31586a CANDIDATE
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
```

Add new LUNS

While the add and drop operations can be performed together, it is generally easier to add new LUNs in two steps. First, add the new LUNs to the disk group. This step results in half of the extents being migrated from the current ASM LUNs to the new LUNs.

The rebalance power indicates the rate at which data is being transferred. The higher the number, the higher the parallelism of the data transfer. The migration is performed with efficient sequential I/O operations that are unlikely to cause performance problems. However, if desired, the rebalance power of an ongoing migration can be adjusted with the `alter diskgroup [name] rebalance power [level]` command. Typical migrations use a value of 5.

```
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586b' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315869' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315858' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586a' rebalance power 5;
Diskgroup altered.
```

Monitor operation

A rebalancing operation can be monitored and managed in multiple ways. We used the following command for this example.

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
------------ ----- ----
           1 REBAL RUN
           1 REBAL WAIT
```

When migration is complete, no rebalancing operations are reported.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

Drop old LUNs

The migration is now halfway complete. It might be desirable to perform some basic performance tests to make sure that the environment is healthy. After confirmation, the remaining data can be relocated by dropping the old LUNs. Note that this does not result in immediate release of the LUNs. The drop operation signals Oracle ASM to relocate the extents first and then release the LUN.

```
sqlplus / as sysasm
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0000 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0001 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0002 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0003 rebalance power 5;
Diskgroup altered.
```

Monitor operation

The rebalancing operation can be monitored and managed in multiple ways. We used the following command for this example:

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
------------ ----- ----
           1 REBAL RUN
           1 REBAL WAIT
```

When migration is complete, no rebalancing operations are reported.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

Remove old LUNs

Before you remove the old LUNs from the disk group, you should perform one final check on the header status. After a LUN is released from ASM, it no longer has a name listed and the header status is listed as `FORMER`. This indicates that these LUNs can safely be removed from the system.

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NAME||''||GROUP_NUMBER||''||TOTAL_MB||''||PATH||''||HEADER_STATUS
--------------------------------------------------------------------------
------
 0 0 /dev/mapper/3600a098038303537762b47594c315863 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315864 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315861 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315862 FORMER
NEWDATA_0005 1 10240 /dev/mapper/3600a098038303537762b47594c315869 MEMBER
NEWDATA_0007 1 10240 /dev/mapper/3600a098038303537762b47594c31586a MEMBER
NEWDATA_0004 1 10240 /dev/mapper/3600a098038303537762b47594c31586b MEMBER
NEWDATA_0006 1 10240 /dev/mapper/3600a098038303537762b47594c315858 MEMBER
8 rows selected.
```

LVM migration

The procedure presented here shows the principles of an LVM-based migration of a volume group called `datavg`. The examples are drawn from the Linux LVM, but the principles apply equally to AIX, HP-UX, and VxVM. The precise commands might vary.

1. Identify the LUNs currently in the `datavg` volume group.

```
[root@host1 ~]# pvdisplay -C | grep datavg
  /dev/mapper/3600a098038303537762b47594c31582f datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c31585a datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c315859 datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c31586c datavg lvm2 a--  10.00g
10.00g
```

2. Create new LUNs of the same or slightly larger physical size and define them as physical volumes.

```
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315864
  Physical volume "/dev/mapper/3600a098038303537762b47594c315864"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315863
  Physical volume "/dev/mapper/3600a098038303537762b47594c315863"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315862
  Physical volume "/dev/mapper/3600a098038303537762b47594c315862"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315861
  Physical volume "/dev/mapper/3600a098038303537762b47594c315861"
successfully created
```

3. Add the new volumes to the volume group.

```
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315864
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315863
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315862
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315861
  Volume group "datavg" successfully extended
```

4. Issue the `pvmove` command to relocate the extents of each current LUN to the new LUN. The `-i [seconds]` argument monitors the progress of the operation.

```
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31582f
/dev/mapper/3600a098038303537762b47594c315864
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 14.2%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 28.4%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 42.5%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 57.1%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 72.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 87.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31585a
/dev/mapper/3600a098038303537762b47594c315863
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 14.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 29.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 44.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 60.1%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 75.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 90.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c315859
/dev/mapper/3600a098038303537762b47594c315862
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 14.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 29.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 45.5%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 61.1%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 76.6%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 91.7%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31586c
/dev/mapper/3600a098038303537762b47594c315861
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 15.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 30.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 46.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 61.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 77.2%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 92.3%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 100.0%
```

5. When this process is complete, drop the old LUNs from the volume group by using the `vgreduce` command. If successful, the LUN can now be safely removed from the system.

```
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31582f
Removed "/dev/mapper/3600a098038303537762b47594c31582f" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31585a
  Removed "/dev/mapper/3600a098038303537762b47594c31585a" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c315859
  Removed "/dev/mapper/3600a098038303537762b47594c315859" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31586c
  Removed "/dev/mapper/3600a098038303537762b47594c31586c" from volume
group "datavg"
```

Foreign LUN import

Planning

The procedures to migrate SAN resources using FLI are documented in NetApp ONTAP Foreign LUN Import Documentation.

From a database and host point of view, no special steps are required. After the FC zones are updated and the LUNs become available on ONTAP, the LVM should be able to read the LVM metadata from the LUNs. Also, the volume groups are ready for use with no further configuration steps. In rare cases, environments might include configuration files that were hard-coded with references to the prior storage array. For example, a Linux system that included `/etc/multipath.conf` rules that referenced a WWN of a given device must be updated to reflect the changes introduced by FLI.

Reference the NetApp Compatibility Matrix for information on supported configurations. If your environment is not included, contact your NetApp representative for assistance.

This example shows the migration of both ASM and LVM LUNs hosted on a Linux server. FLI is supported on other operating systems, and, although the host-side commands might differ, the principles are the same, and the ONTAP procedures are identical.

Identify LVM LUNs

The first step in preparation is to identify the LUNs to be migrated. In the example shown here, two SAN-based file systems are mounted at `/orabin` and `/backups`.

```
[root@host1 ~]# df -k
Filesystem                          1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root                52403200   8811464  43591736  17% /
devtmpfs                             65882776         0  65882776   0% /dev
...
fas8060-nfs-public:/install         199229440 119368128  79861312  60%
/install
/dev/mapper/sanvg-lvorabin           20961280  12348476   8612804  59%
/orabin
/dev/mapper/sanvg-lvbackups          73364480  62947536  10416944  86%
/backups
```

The name of the volume group can be extracted from the device name, which uses the format (volume group name)-(logical volume name). In this case, the volume group is called `sanvg`.

The `pvdisplay` command can be used as follows to identify the LUNs that support this volume group. In this case, there are 10 LUNs that make up the `sanvg` volume group.

```
[root@host1 ~]# pvdisplay -C -o pv_name,pv_size,pv_fmt,vg_name
  PV                                                PSize   VG
  /dev/mapper/3600a0980383030445424487556574266  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574267  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574268  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574269  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426a  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426b  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426c  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426d  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426e  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426f  10.00g sanvg
  /dev/sda2                                       278.38g rhel
```

Identify ASM LUNs

ASM LUNs must also be migrated. To obtain the number of LUNs and LUN paths from sqlplus as the sysasm user, run the following command:

```
SQL> select path||' '||os_mb from v$asm_disk;
PATH||''||OS_MB
--------------------------------------------------------------------------------
------
/dev/oracleasm/disks/ASM0 10240
/dev/oracleasm/disks/ASM9 10240
/dev/oracleasm/disks/ASM8 10240
/dev/oracleasm/disks/ASM7 10240
/dev/oracleasm/disks/ASM6 10240
/dev/oracleasm/disks/ASM5 10240
/dev/oracleasm/disks/ASM4 10240
/dev/oracleasm/disks/ASM1 10240
/dev/oracleasm/disks/ASM3 10240
/dev/oracleasm/disks/ASM2 10240
10 rows selected.
SQL>
```

FC network changes

The current environment contains 20 LUNs to be migrated. Update the current SAN so that ONTAP can access the current LUNs. Data is not migrated yet, but ONTAP must read configuration information from the current LUNs to create the new home for that data.

At a minimum, at least one HBA port on the AFF/FAS system must be configured as an initiator port. In addition, the FC zones must be updated so that ONTAP can access the LUNs on the foreign storage array. Some storage arrays have LUN masking configured, which limits which WWNs can access a given LUN. In such cases, LUN masking must also be updated to grant access to the ONTAP WWNs.

After this step is completed, ONTAP should be able to view the foreign storage array with the `storage array show` command. The key field it returns is the prefix that is used to identify the foreign LUN on the system. In the example below, the LUNs on the foreign array `FOREIGN_1` appear within ONTAP using the prefix of `FOR-1`.

Identify foreign array

```
Cluster01::> storage array show -fields name,prefix
name          prefix
------------- ------
FOREIGN_1     FOR-1
Cluster01::>
```

Identify foreign LUNs

The LUNs can be listed by passing the `array-name` to the `storage disk show` command. The data returned is referenced multiple times during the migration procedure.

```
Cluster01::> storage disk show -array-name FOREIGN_1 -fields disk,serial
disk     serial-number
-------- -------------
FOR-1.1  800DT$HuVWBX
FOR-1.2  800DT$HuVWBZ
FOR-1.3  800DT$HuVWBW
FOR-1.4  800DT$HuVWBY
FOR-1.5  800DT$HuVWB/
FOR-1.6  800DT$HuVWBa
FOR-1.7  800DT$HuVWBd
FOR-1.8  800DT$HuVWBb
FOR-1.9  800DT$HuVWBc
FOR-1.10 800DT$HuVWBe
FOR-1.11 800DT$HuVWBf
FOR-1.12 800DT$HuVWBg
FOR-1.13 800DT$HuVWBi
FOR-1.14 800DT$HuVWBh
FOR-1.15 800DT$HuVWBj
FOR-1.16 800DT$HuVWBk
FOR-1.17 800DT$HuVWBm
FOR-1.18 800DT$HuVWBl
FOR-1.19 800DT$HuVWBo
FOR-1.20 800DT$HuVWBn
20 entries were displayed.
Cluster01::>
```

Register foreign array LUNs as import candidates

The foreign LUNs are initially classified as any particular LUN type. Before data can be imported, the LUNs must be tagged as foreign and therefore a candidate for the import process. This step is completed by passing the serial number to the `storage disk modify` command, as shown in the following example. Note that this process tags only the LUN as foreign within ONTAP. No data is written to the foreign LUN itself.

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign true
Cluster01::*>
```

Create volumes to host migrated LUNs

A volume is needed to host the migrated LUNs. The exact volume configuration depends on the overall plan to leverage ONTAP features. In this example, the ASM LUNs are placed into one volume and the LVM LUNs are placed in a second volume. Doing so allows you to manage the LUNs as independent groups for purposes such as tiering, creation of snapshots, or setting QoS controls.

Set the `snapshot-policy `to `none`. The migration process can include a great deal of data turnover. Therefore, there might be a large increase in space consumption if snapshots are created by accident because unwanted data is captured in the snapshots.

```
Cluster01::> volume create -volume new_asm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1152] Job succeeded: Successful
Cluster01::> volume create -volume new_lvm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1153] Job succeeded: Successful
Cluster01::>
```

Create ONTAP LUNs

After the volumes are created, the new LUNs must be created. Normally, the creation of a LUN requires the user to specify such information as the LUN size, but in this case the foreign-disk argument is passed to the command. As a result, ONTAP replicates the current LUN configuration data from the specified serial number. It also uses the LUN geometry and partition table data to adjust LUN alignment and establish optimum performance.

In this step, serial numbers must be cross-referenced against the foreign array to make sure that the correct foreign LUN is matched to the correct new LUN.

```
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN0 -ostype
linux -foreign-disk 800DT$HuVWBW
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN1 -ostype
linux -foreign-disk 800DT$HuVWBX
Created a LUN of size 10g (10737418240)
...
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN8 -ostype
linux -foreign-disk 800DT$HuVWBn
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN9 -ostype
linux -foreign-disk 800DT$HuVWBo
Created a LUN of size 10g (10737418240)
```

Create import relationships

The LUNs have now been created but are not configured as a replication destination. Before this step can be taken, the LUNs must first be placed offline. This extra step is designed to protect data from user errors. If ONTAP allowed a migration to be performed on an online LUN, it would create a risk that a typographical error could result in overwriting active data. The extra step of forcing the user to first take a LUN offline helps verify that the correct target LUN is used as a migration destination.

```
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN0
Warning: This command will take LUN "/vol/new_asm/LUN0" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN1
Warning: This command will take LUN "/vol/new_asm/LUN1" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
...
Warning: This command will take LUN "/vol/new_lvm/LUN8" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_lvm/LUN9
Warning: This command will take LUN "/vol/new_lvm/LUN9" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
```

After the LUNs are offline, you can establish the import relationship by passing the foreign LUN serial number to the `lun import create` command.

```
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN0
-foreign-disk 800DT$HuVWBW
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN1
-foreign-disk 800DT$HuVWBX
...
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN8
-foreign-disk 800DT$HuVWBn
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN9
-foreign-disk 800DT$HuVWBo
Cluster01::*>
```

After all import relationships are established, the LUNs can be placed back online.

```
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN9
Cluster01::*>
```

Create initiator group

An initiator group (igroup) is part of the ONTAP LUN masking architecture. A newly created LUN is not accessible unless a host is first granted access. This is done by creating an igroup that lists either the FC WWNs or iSCSI initiator names that should be granted access. At the time this report was written, FLI was supported only for FC LUNs. However, converting to iSCSI postmigration is a simple task, as shown in Protocol Conversion.

In this example, an igroup is created that contains two WWNs that correspond to the two ports available on the host's HBA.

```
Cluster01::*> igroup create linuxhost -protocol fcp -ostype linux
-initiator 21:00:00:0e:1e:16:63:50 21:00:00:0e:1e:16:63:51
```

Map new LUNs to host

Following igroup creation, the LUNs are then mapped to the defined igroup. These LUNs are available only to the WWNs included in this igroup. NetApp assumes at this stage in the migration process that the host has not been zoned to ONTAP. This is important because if the host is simultaneously zoned to the foreign array and the new ONTAP system, then there is a risk that LUNs bearing the same serial number could be discovered on each array. This situation could lead to multipath malfunctions or damage to data.

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
Cluster01::*>
```

Cutover

Some disruption during a foreign LUN import is unavoidable because of the need to change the FC network configuration. However, the disruption does not have to last much longer than the time required to restart the database environment and update FC zoning

to switch the host FC connectivity from the foreign LUN to ONTAP.

This process can be summarized as follows:

1. Quiesce all LUN activity on the foreign LUNs.
2. Redirect host FC connections to the new ONTAP system.
3. Trigger the import process.
4. Rediscover the LUNs.
5. Restart the database.

You do not need to wait for the migration process to complete. As soon as the migration for a given LUN begins, it is available on ONTAP and can serve data while the data copy process continues. All reads are passed through to the foreign LUN, and all writes are synchronously written to both arrays. The copy operation is very fast and the overhead of redirecting FC traffic is minimal, so any impact on performance should be transient and minimal. If there is concern, you can delay restarting the environment until after the migration process is complete and the import relationships have been deleted.

Shut down database

The first step in quiescing the environment in this example is to shut down the database.

```
[oracle@host1 bin]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base remains unchanged with value /orabin
[oracle@host1 bin]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

Shut down grid services

One of the SAN-based file systems being migrated also includes the Oracle ASM services. Quiescing the underlying LUNs requires dismounting the file systems, which in turn means stopping any processes with open files on this file system.

```
[oracle@host1 bin]$ ./crsctl stop has -f
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'host1'
CRS-2673: Attempting to stop 'ora.evmd' on 'host1'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host1'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host1'
CRS-2677: Stop of 'ora.DATA.dg' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host1'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host1' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host1' succeeded
CRS-2677: Stop of 'ora.asm' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host1'
CRS-2677: Stop of 'ora.cssd' on 'host1' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'host1' has completed
CRS-4133: Oracle High Availability Services has been stopped.
[oracle@host1 bin]$
```

Dismount file systems

If all the processes are shut down, the umount operation succeeds. If permission is denied, there must be a process with a lock on the file system. The `fuser` command can help identify these processes.

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

Deactivate volume groups

After all file systems in a given volume group are dismounted, the volume group can be deactivated.

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

FC network changes

The FC zones can now be updated to remove all access from the host to the foreign array and establish access to ONTAP.

Start import process

To start the LUN import processes, run the `lun import start` command.

```
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN0
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN1
...
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN8
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN9
Cluster01::lun import*>
```

Monitor import progress

The import operation can be monitored with the `lun import show` command. As shown below, the import of all 20 LUNs is underway, which means that data is now accessible through ONTAP even though the data copy operation still progresses.

```
Cluster01::lun import*> lun import show -fields path,percent-complete
vserver   foreign-disk path                 percent-complete
--------- ------------ ------------------ ----------------
vserver1  800DT$HuVWB/ /vol/new_asm/LUN4 5
vserver1  800DT$HuVWBW /vol/new_asm/LUN0 5
vserver1  800DT$HuVWBX /vol/new_asm/LUN1 6
vserver1  800DT$HuVWBY /vol/new_asm/LUN2 6
vserver1  800DT$HuVWBZ /vol/new_asm/LUN3 5
vserver1  800DT$HuVWBa /vol/new_asm/LUN5 4
vserver1  800DT$HuVWBb /vol/new_asm/LUN6 4
vserver1  800DT$HuVWBc /vol/new_asm/LUN7 4
vserver1  800DT$HuVWBd /vol/new_asm/LUN8 4
vserver1  800DT$HuVWBe /vol/new_asm/LUN9 4
vserver1  800DT$HuVWBf /vol/new_lvm/LUN0 5
vserver1  800DT$HuVWBg /vol/new_lvm/LUN1 4
vserver1  800DT$HuVWBh /vol/new_lvm/LUN2 4
vserver1  800DT$HuVWBi /vol/new_lvm/LUN3 3
vserver1  800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1  800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1  800DT$HuVWBl /vol/new_lvm/LUN6 4
vserver1  800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1  800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1  800DT$HuVWBo /vol/new_lvm/LUN9 2
20 entries were displayed.
```

If you require an offline process, delay rediscovering or restarting services until the `lun import show` command indicates that all migration is successful and complete. You can then complete the migration process as described in Foreign LUN Import—Completion.

If you require an online migration, proceed to rediscover the LUNs in their new home and bring up the services.

Scan for SCSI device changes

In most cases, the simplest option to rediscover new LUNs is to restart the host. Doing so automatically removes old stale devices, properly discovers all new LUNs, and builds associated devices such as multipathing devices. The example here shows a wholly online process for demonstration purposes.

Caution: Before restarting a host, make sure that all entries in `/etc/fstab` that reference migrated SAN resources are commented out. If this is not done and there are problems with LUN access, the OS might not boot. This situation does not damage data. However, it can be very inconvenient to boot into rescue mode or a similar mode and correct the `/etc/fstab` so that the OS can be booted to enable troubleshooting.

The LUNs on the version of Linux used in this example can be rescanned with the `rescan-scsi-bus.sh` command. If the command is successful, each LUN path should appear in the output. The output can be difficult to interpret, but, if the zoning and igroup configuration was correct, many LUNs should appear that include a `NETAPP` vendor string.

```
[root@host1 /]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 0 2 0 0 ...
OLD: Host: scsi0 Channel: 02 Id: 00 Lun: 00
      Vendor: LSI      Model: RAID SAS 6G 0/1  Rev: 2.13
      Type:   Direct-Access                    ANSI SCSI revision: 05
Scanning host 1 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 1 0 0 0 ...
OLD: Host: scsi1 Channel: 00 Id: 00 Lun: 00
      Vendor: Optiarc  Model: DVD RW AD-7760H  Rev: 1.41
      Type:   CD-ROM                           ANSI SCSI revision: 05
Scanning host 2 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 3 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 4 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 5 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 6 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 7 for  all SCSI target IDs, all LUNs
 Scanning for device 7 0 0 10 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 10
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 11 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 11
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 12 ...
...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 18
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 9 0 1 19 ...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 19
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.
```

Check for multipath devices

The LUN discovery process also triggers the recreation of multipath devices, but the Linux multipathing driver is known to have occasional problems. The output of `multipath -ll` should be checked to verify that the output looks as expected. For example, the output below shows multipath devices associated with a `NETAPP` vendor string. Each device has four paths, with two at a priority of 50 and two at a priority of 10. Although the exact output can vary with different versions of Linux, this output looks as expected.

Reference the host utilities documentation for the version of Linux you use to verify that the `/etc/multipath.conf` settings are correct.

```
[root@host1 /]# multipath -ll
3600a098038303558735d493762504b36 dm-5 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:4  sdat 66:208 active ready running
| `- 9:0:1:4  sdbn 68:16  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:4  sdf  8:80   active ready running
  `- 9:0:0:4  sdz  65:144 active ready running
3600a098038303558735d493762504b2d dm-10 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:8  sdax 67:16  active ready running
| `- 9:0:1:8  sdbr 68:80  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:8  sdj  8:144  active ready running
  `- 9:0:0:8  sdad 65:208 active ready running
...
3600a098038303558735d493762504b37 dm-8 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:5  sdau 66:224 active ready running
| `- 9:0:1:5  sdbo 68:32  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:5  sdg  8:96   active ready running
  `- 9:0:0:5  sdaa 65:160 active ready running
3600a098038303558735d493762504b4b dm-22 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:19 sdbi 67:192 active ready running
| `- 9:0:1:19 sdcc 69:0   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:19 sdu  65:64  active ready running
  `- 9:0:0:19 sdao 66:128 active ready running
```

Reactivate LVM volume group

If the LVM LUNs have been properly discovered, the `vgchange --activate y` command should succeed.

This is a good example of the value of a logical volume manager. A change in the WWN of a LUN or even a serial number is unimportant because the volume group metadata is written on the LUN itself.

The OS scanned the LUNs and discovered a small amount of data written on the LUN that identifies it as a physical volume belonging to the `sanvg volumegroup`. It then built all of the required devices. All that is required is to reactivate the volume group.

```
[root@host1 /]# vgchange --activate y sanvg
  Found duplicate PV fpCzdLTuKfy2xDZjai1NliJh3TjLUBiT: using
/dev/mapper/3600a098038303558735d493762504b46 not /dev/sdp
  Using duplicate PV /dev/mapper/3600a098038303558735d493762504b46 from
subsystem DM, ignoring /dev/sdp
  2 logical volume(s) in volume group "sanvg" now active
```

Remount file systems

After the volume group is reactivated, the file systems can be mounted with all of the original data intact. As discussed previously, the file systems are fully operational even if data replication is still active in the back group.

```
[root@host1 /]# mount /orabin
[root@host1 /]# mount /backups
[root@host1 /]# df -k
Filesystem                      1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root            52403200   8837100  43566100  17% /
devtmpfs                         65882776         0  65882776   0% /dev
tmpfs                             6291456        84   6291372   1%
/dev/shm
tmpfs                            65898668      9884  65888784   1% /run
tmpfs                            65898668         0  65898668   0%
/sys/fs/cgroup
/dev/sda1                          505580    224828    280752  45% /boot
fas8060-nfs-public:/install      199229440 119368256  79861184  60%
/install
fas8040-nfs-routable:/snapomatic   9961472     30528   9930944   1%
/snapomatic
tmpfs                            13179736        16  13179720   1%
/run/user/42
tmpfs                            13179736         0  13179736   0%
/run/user/0
/dev/mapper/sanvg-lvorabin       20961280  12357456   8603824  59%
/orabin
/dev/mapper/sanvg-lvbackups      73364480  62947536  10416944  86%
/backups
```

Rescan for ASM devices

The ASMlib devices should have been rediscovered when the SCSI devices were rescanned. Rediscovery can be verified online by restarting ASMlib and then scanning the disks.

This step is only relevant to ASM configurations where ASMlib is used.

Caution: Where ASMlib is not used, the `/dev/mapper` devices should have been automatically recreated. However, the permissions might not be correct. You must set special permissions on the underlying devices for ASM in the absence of ASMlib. Doing so is usually accomplished through special entries in either the `/etc/multipath.conf` or `udev` rules, or possibly in both rule sets. These files might need to be updated to reflect changes in the environment in terms of WWNs or serial numbers to make sure that the ASM devices still have the correct permissions.

In this example, restarting ASMlib and scanning for disks show the same 10 ASM LUNs as the original environment.

```
[root@host1 /]# oracleasm exit
Unmounting ASMlib driver filesystem: /dev/oracleasm
Unloading module "oracleasm": oracleasm
[root@host1 /]# oracleasm init
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
[root@host1 /]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASM0"
Instantiating disk "ASM1"
Instantiating disk "ASM2"
Instantiating disk "ASM3"
Instantiating disk "ASM4"
Instantiating disk "ASM5"
Instantiating disk "ASM6"
Instantiating disk "ASM7"
Instantiating disk "ASM8"
Instantiating disk "ASM9"
```

Restart grid services

Now that the LVM and ASM devices are online and available, the grid services can be restarted.

```
[root@host1 /]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

Restart database

After the grid services have been restarted, the database can be brought up. It might be necessary to wait a few minutes for the ASM services to become fully available before trying to start the database.

```
[root@host1 bin]# su - oracle
[oracle@host1 ~]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base has been set to /orabin
[oracle@host1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> startup
ORACLE instance started.
Total System Global Area 3221225472 bytes
Fixed Size                  4502416 bytes
Variable Size            1207962736 bytes
Database Buffers         1996488704 bytes
Redo Buffers               12271616 bytes
Database mounted.
Database opened.
SQL>
```

Completion

From a host point of view, the migration is complete, but I/O is still served from the foreign array until the import relationships are deleted.

Before deleting the relationships, you must confirm that the migration process is complete for all LUNs.

```
Cluster01::*> lun import show -vserver vserver1 -fields foreign-
disk,path,operational-state
vserver   foreign-disk path              operational-state
--------- ------------ ----------------- -----------------
vserver1 800DT$HuVWB/ /vol/new_asm/LUN4 completed
vserver1 800DT$HuVWBW /vol/new_asm/LUN0 completed
vserver1 800DT$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBY /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN3 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBc /vol/new_asm/LUN7 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN8 completed
vserver1 800DT$HuVWBe /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBi /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBl /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBo /vol/new_lvm/LUN9 completed
20 entries were displayed.
```

Delete import relationships

When the migration process is complete, delete the migration relationship. After you have done so, I/O is served exclusively from the drives on ONTAP.

```
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN9
```

Deregister foreign LUNs

Finally, modify the disk to remove the `is-foreign` designation.

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign false
Cluster01::*>
```

Protocol conversion

Changing the protocol used to access a LUN is a common requirement.

In some cases, it is part of an overall strategy to migrate data to the cloud. TCP/IP is the protocol of the cloud, and changing from FC to iSCSI allows easier migration into various cloud environments. In other cases, iSCSI might be desirable to leverage the decreased costs of an IP SAN. On occasion, a migration might use a different protocol as a temporary measure. For example, if a foreign array and ONTAP based LUNs cannot coexist on the same HBAs, you can use iSCSI LUNs long enough to copy data from the old array. You can then convert back to FC after the old LUNs are removed from the system.

The following procedure demonstrates conversion from FC to iSCSI, but the overall principles apply to a reverse iSCSI to FC conversion.

Install iSCSI initiator

Most operating systems include a software iSCSI initiator by default, but if one is not included, it can be easily installed.

```
[root@host1 /]# yum install -y iscsi-initiator-utils
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-
              : manager
Resolving Dependencies
--> Running transaction check
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.el7 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.el7 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.el7 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
will be an update
```

```
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
===
 Package                         Arch   Version              Repository
Size
================================================================================
===
Updating:
 iscsi-initiator-utils           x86_64 6.2.0.873-32.0.2.el7 ol7_latest 416
k
Updating for dependencies:
 iscsi-initiator-utils-iscsiuio x86_64 6.2.0.873-32.0.2.el7 ol7_latest  84
k
Transaction Summary
================================================================================
===
Upgrade  1 Package (+1 Dependent package)
Total download size: 501 k
Downloading packages:
No Presto metadata available for ol7_latest
(1/2): iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_6 | 416 kB   00:00
(2/2): iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2. |  84 kB   00:00
--------------------------------------------------------------------------------
---
Total                                           2.8 MB/s | 501 kB
00:00Cluster01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
1/4
  Updating   : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
2/4
  Cleanup    : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Cleanup    : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
rhel-7-server-eus-rpms/7Server/x86_64/productid       | 1.7 kB     00:00
rhel-7-server-rpms/7Server/x86_64/productid           | 1.7 kB     00:00
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
1/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
2/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
```

```
3/4
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
Updated:
  iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7
Dependency Updated:
  iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
Complete!
[root@host1 /]#
```

Identify iSCSI initiator name

A unique iSCSI initiator name is generated during the installation process. On Linux, it is located in the `/etc/iscsi/initiatorname.iscsi` file. This name is used to identify the host on the IP SAN.

```
[root@host1 /]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0
```

Create new initiator group

An initiator group (igroup) is part of the ONTAP LUN masking architecture. A newly created LUN is not accessible unless a host is first granted access. This step is accomplished by creating an igroup that lists either the FC WWNs or iSCSI initiator names that require access.

In this example, an igroup is created that contains the iSCSI initiator of the Linux host.

```
Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0
```

Shut down environment

Before changing the LUN protocol, the LUNs must be fully quiesced. Any database on one of the LUNs being converted must be shut down, file systems must be dismounted, and volume groups must be deactivated. Where ASM is used, make sure that the ASM disk group is dismounted and shut down all grid services.

Unmap LUNs from FC network

After the LUNs are fully quiesced, remove the mappings from the original FC igroup.

```
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
```

Remap LUNs to IP network

Grant access to each LUN to the new iSCSI-based initiator group.

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxiscsi
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxiscsi
Cluster01::*>
```

Discover iSCSI targets

There are two phases to iSCSI discovery. The first is to discover the targets, which is not the same as discovering a LUN. The `iscsiadm` command shown below probes the portal group specified by the `-p argument` and stores a list of all IP addresses and ports that offer iSCSI services. In this case, there are four IP addresses that have iSCSI services on the default port 3260.

> This command can take several minutes to complete if any of the target IP addresses cannot be reached.

```
[root@host1 ~]# iscsiadm -m discovery -t st -p fas8060-iscsi-public1
10.63.147.197:3260,1033 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
10.63.147.198:3260,1034 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.203:3260,1030 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.202:3260,1029 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
```

Discover iSCSI LUNs

After the iSCSI targets are discovered, restart the iSCSI service to discover the available iSCSI LUNs and build associated devices such as multipath or ASMlib devices.

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart  iscsi.service
```

Restart environment

Restart the environment by reactivating volume groups, remounting file systems, restarting RAC services, and so on. As a precaution, NetApp recommends that you reboot the server after the conversion process is complete to be certain that all configuration files are correct and all stale devices are removed.

Caution: Before restarting a host, make sure that all entries in `/etc/fstab` that reference migrated SAN resources are commented out. If this step is not taken and there are problems with LUN access, the result can be an OS that does not boot. This issue does not damage data. However, it can be very inconvenient to boot into rescue mode or a similar mode and correct `/etc/fstab` so that the OS can be booted to allow troubleshooting efforts to begin.

Sample scripts

The scripts presented are provided as examples of how to script various OS and database tasks. They are supplied as is. If support is required for a particular procedure, contact NetApp or a NetApp reseller.

Database shutdown

The following Perl script takes a single argument of the Oracle SID and shuts down a database. It can be run as the Oracle user or as root.

```
#! /usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
77 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF4
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
`;};
print @out;
if ("@out" =~ /ORACLE instance shut down/) {
print "$oraclesid shut down\n";
exit 0;}
elsif ("@out" =~ /Connected to an idle instance/) {
print "$oraclesid already shut down\n";
exit 0;}
else {
print "$oraclesid failed to shut down\n";
exit 1;}
```

Database startup

The following Perl script takes a single argument of the Oracle SID and shuts down a database. It can be run as the Oracle user or as root.

```
#! /usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
'
`;}
else {
@out=`. oraenv << EOF3
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`;};
print @out;
if ("@out" =~ /Database opened/) {
print "$oraclesid started\n";
exit 0;}
elsif ("@out" =~ /cannot start already-running ORACLE/) {
print "$oraclesid already started\n";
exit 1;}
else {
78 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
print "$oraclesid failed to start\n";
exit 1;}
```

Convert file system to read-only

The following script takes a file- system argument and attempts to dismount and remount it as read-only. Doing so is useful during migration processes in which a file system must be kept available to replicate data and yet must be protected against accidental damage.

```
#! /usr/bin/perl
use strict;
#use warnings;
my $filesystem=$ARGV[0];
my @out=`umount '$filesystem'`;
if ($? == 0) {
 print "$filesystem unmounted\n";
 @out = `mount -o ro '$filesystem'`;
 if ($? == 0) {
 print "$filesystem mounted read-only\n";
 exit 0;}}
else {
 print "Unable to unmount $filesystem\n";
 exit 1;}
print @out;
```

Replace file system

The following script example is used to replace one file system with another. Because it edits the `/etc/fstab `file, it must run as root. It accepts a single comma-delimited argument of the old and new file systems.

1. To replace the file system, run the following script:

```
#! /usr/bin/perl
use strict;
#use warnings;
my $oldfs;
my $newfs;
my @oldfstab;
my @newfstab;
my $source;
my $mountpoint;
my $leftover;
my $oldfstabentry='';
my $newfstabentry='';
my $migratedfstabentry='';
($oldfs, $newfs) = split (',',$ARGV[0]);
open(my $filehandle, '<', '/etc/fstab') or die "Could not open
/etc/fstab\n";
while (my $line = <$filehandle>) {
 chomp $line;
 ($source, $mountpoint, $leftover) = split(/[ , ]/,$line, 3);
 if ($mountpoint eq $oldfs) {
 $oldfstabentry = "#Removed by swap script $source $oldfs $leftover";}
 elsif ($mountpoint eq $newfs) {
```

```
 $newfstabentry = "#Removed by swap script $source $newfs $leftover";
 $migratedfstabentry = "$source $oldfs $leftover";}
 else {
 push (@newfstab, "$line\n")}}
79 Migration of Oracle Databases to NetApp Storage Systems © 2021
NetApp, Inc. All rights reserved
push (@newfstab, "$oldfstabentry\n");
push (@newfstab, "$newfstabentry\n");
push (@newfstab, "$migratedfstabentry\n");
close($filehandle);
if ($oldfstabentry eq ''){
 die "Could not find $oldfs in /etc/fstab\n";}
if ($newfstabentry eq ''){
 die "Could not find $newfs in /etc/fstab\n";}
my @out=`umount '$newfs'`;
if ($? == 0) {
 print "$newfs unmounted\n";}
else {
 print "Unable to unmount $newfs\n";
 exit 1;}
@out=`umount '$oldfs'`;
if ($? == 0) {
 print "$oldfs unmounted\n";}
else {
 print "Unable to unmount $oldfs\n";
 exit 1;}
system("cp /etc/fstab /etc/fstab.bak");
open ($filehandle, ">", '/etc/fstab') or die "Could not open /etc/fstab
for writing\n";
for my $line (@newfstab) {
 print $filehandle $line;}
close($filehandle);
@out=`mount '$oldfs'`;
if ($? == 0) {
 print "Mounted updated $oldfs\n";
 exit 0;}
else{
 print "Unable to mount updated $oldfs\n";
 exit 1;}
exit 0;
```

As an example of this script's use, assume that data in `/oradata` is migrated to `/neworadata` and `/logs` is migrated to `/newlogs`. One of the simplest methods to perform this task is by using a simple file copy operation to relocate the new device back to the original mountpoint.

2. Assume that the old and new file systems are present in the `/etc/fstab` file as follows:

```
cluster01:/vol_oradata /oradata nfs rw,bg,vers=3,rsize=65536,wsize=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsize=65536,wsize=65536
0 0
```

3. When run, this script unmounts the current file system and replaces it with the new:

```
[root@jfsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jfsc3 scripts]# ./swap.fs.pl /logs,/newlogs
/newlogs unmounted
/logs unmounted
Mounted updated /logs
```

4. The script also updates the `/etc/fstab` file accordingly. In the example shown here, it includes the following changes:

```
#Removed by swap script cluster01:/vol_oradata /oradata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsize=65536,wsize=65536 0
0
```

Automated database migration

This example demonstrates the use of shutdown, startup, and file system replacement scripts to fully automate a migration.

```
#! /usr/bin/perl
use strict;
#use warnings;
```

```
my $oraclesid=$ARGV[0];
my @oldfs;
my @newfs;
my $x=1;
while ($x < scalar(@ARGV)) {
   ($oldfs[$x-1], $newfs[$x-1]) = split (',',$ARGV[$x]);
   $x+=1;}
my @out=`./dbshut.pl '$oraclesid'`;
print @out;
if ($? ne 0) {
   print "Failed to shut down database\n";
   exit 0;}
$x=0;
while ($x < scalar(@oldfs)) {
   my @out=`./mk.fs.readonly.pl '$oldfs[$x]'`;
   if ($? ne 0) {
      print "Failed to make filesystem $oldfs[$x] readonly\n";
      exit 0;}
   $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
   my @out=`rsync -rlpogt --stats --progress --exclude='.snapshot'
'$oldfs[$x]/' '/$newfs[$x]/'`;
   print @out;
   if ($? ne 0) {
      print "Failed to copy filesystem $oldfs[$x] to $newfs[$x]\n";
      exit 0;}
   else {
      print "Succesfully replicated filesystem $oldfs[$x] to
$newfs[$x]\n";}
   $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
   print "swap $x $oldfs[$x] $newfs[$x]\n";
   my @out=`./swap.fs.pl '$oldfs[$x],$newfs[$x]'`;
   print @out;
   if ($? ne 0) {
      print "Failed to swap filesystem $oldfs[$x] for $newfs[$x]\n";
      exit 1;}
   else {
      print "Swapped filesystem $oldfs[$x] for $newfs[$x]\n";}
   $x+=1;}
my @out=`./dbstart.pl '$oraclesid'`;
print @out;
```

Display file locations

This script collects a number of critical database parameters and prints them in an easy-to-read format. This script can be useful when reviewing data layouts. In addition, the script can be modified for other uses.

```
#! /usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
        my $command = @_[0];
        my @lines;
        my $uid=$<;
        if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `;}
        else {
        $command=~s/\\\\\\\/\\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `;};
return @lines}
print "\n";
@out=dosql('select name from v\\\\\$datafile;');
print "$oraclesid datafiles:\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select member from v\\\\\$logfile;');
print "$oraclesid redo logs:\n";
for $line (@out) {
```

```
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name from v\\\\\$tempfile;');
print "$oraclesid temp datafiles:\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('show parameter spfile;');
print "$oraclesid spfile\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name||\' \'||value from v\\\\\$parameter where
isdefault=\'FALSE\';');
print "$oraclesid key parameters\n";
for $line (@out) {
        chomp($line);
        if ($line =~ /control_files/) {print "$line\n";}
        if ($line =~ /db_create/) {print "$line\n";}
        if ($line =~ /db_file_name_convert/) {print "$line\n";}
        if ($line =~ /log_archive_dest/) {print "$line\n";}}
        if ($line =~ /log_file_name_convert/) {print "$line\n";}
        if ($line =~ /pdb_file_name_convert/) {print "$line\n";}
        if ($line =~ /spfile/) {print "$line\n";}
print "\n";
```

ASM migration cleanup

```
#! /usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
        my $command = @_[0];
        my @lines;
        my $uid=$<;
        if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
```

```
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `;}
        else {
        $command=~s/\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `;}
return @lines}
print "\n";
@out=dosql('select name from v\\\\\$datafile;');
print @out;
print "shutdown immediate;\n";
print "startup mount;\n";
print "\n";
for $line (@out) {
        if (length($line) > 1) {
                chomp($line);
                ($first, $second,$third,$fourth)=split('_',$line);
                $fourth =~ s/^TS-//;
                $newname=lc("$fourth.dbf");
                $path2file=$line;
                $path2file=~ /(^.*.\/)/;
                print "host mv $line $1$newname\n";}}
print "\n";
for $line (@out) {
        if (length($line) > 1) {
                chomp($line);
                ($first, $second,$third,$fourth)=split('_',$line);
                $fourth =~ s/^TS-//;
                $newname=lc("$fourth.dbf");
                $path2file=$line;
                $path2file=~ /(^.*.\/)/;
                print "alter database rename file '$line' to
'$1$newname';\n";}}
print "alter database open;\n";
print "\n";
```

ASM to file system name conversion

```
set serveroutput on;
set wrap off;
declare
    cursor df is select file#, name from v$datafile;
    cursor tf is select file#, name from v$tempfile;
    cursor lf is select member from v$logfile;
    firstline boolean := true;
begin
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('Parameters for log file conversion:');
    dbms_output.put_line(CHR(13));
    dbms_output.put('*.log_file_name_convert = ');
    for lfrec in lf loop
        if (firstline = true) then
            dbms_output.put('''' || lfrec.member || ''', ');
            dbms_output.put('''/NEW_PATH/' ||
regexp_replace(lfrec.member,'^.*/','') || '''');
        else
            dbms_output.put(',''' || lfrec.member || ''', ');
            dbms_output.put('''/NEW_PATH/' ||
regexp_replace(lfrec.member,'^.*/','') || '''');
        end if;
        firstline:=false;
    end loop;
    dbms_output.put_line(CHR(13));
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('rman duplication script:');
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('run');
    dbms_output.put_line('{');
    for dfrec in df loop
        dbms_output.put_line('set newname for datafile ' ||
            dfrec.file# || ' to ''' || dfrec.name ||''';');
    end loop;
    for tfrec in tf loop
        dbms_output.put_line('set newname for tempfile ' ||
            tfrec.file# || ' to ''' || tfrec.name ||''';');
    end loop;
    dbms_output.put_line('duplicate target database for standby backup
location INSERT_PATH_HERE;');
    dbms_output.put_line('}');
end;
/
```

Replay logs on database

This script accepts a single argument of an Oracle SID for a database that is in mount mode and attempts to replay all currently available archive logs.

```
#! /usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
84 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
print @out;
```

Replay logs on standby database

This script is identical to the preceding script, except that it is designed for a standby database.

```
#! /usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
`;
}
print @out;
```

Additional notes

Performance optimization and benchmarking

Accurate testing of database storage performance is an extremely complicated subject. It requires an understanding of the following issues:

- IOPS and throughput
- The difference between foreground and background I/O operations
- The effect of latency upon the database
- Numerous OS and network settings that also affect storage performance

In addition, there are nonstorage databases tasks to consider. There is a point at which optimizing storage performance yields no useful benefits because storage performance is no longer a limiting factor for performance.

A majority of database customers now select all-flash arrays, which creates some additional considerations.

For example, consider performance testing on a two-node AFF A900 system:

- With a 80/20 read/write ratio, two A900 nodes can deliver over 1M random database IOPS before latency even crosses the 150µs mark. This is so far beyond the current performance demands of most databases that it is difficult to predict the expected improvement. Storage would be largely erased as a bottleneck.
- Network bandwidth is an increasingly common source of performance limitations. For example, spinning disk solutions are often bottlenecks for database performance because the I/O latency is very high. When latency limitations are removed by an all-flash array, the barrier frequently shifts to the network. This is especially notable with virtualized environments and blade systems where the true network connectivity is difficult to visualize. This can complicate performance testing if the storage system itself cannot be fully utilized due to bandwidth limitations.
- Comparing the performance of an all-flash array with an array containing spinning disks is generally not possible because of the dramatically improved latency of all-flash arrays. Test results are typically not meaningful.
- Comparing peak IOPS performance with an all-flash array is frequently not a useful test because databases are not limited by storage I/O. For example, assume one array can sustain 500K random IOPS, whereas another can sustain 300K. The difference is irrelevant in the real world if a database is spending 99% of its time on CPU processing. The workloads never utilize the full capabilities of the storage array. In contrast, peak IOPS capabilities might be critical in a consolidation platform in which the storage array is expected to be loaded to its peak capabilities.
- Always consider latency as well as IOPS in any storage test. Many storage arrays in the market make claims of extreme levels of IOPS, but the latency renders those IOPS useless at such levels. The typical target with all-flash arrays is the 1ms mark. A better approach to testing is not to measure the maximum possible IOPS, but to determine how many IOPS a storage array can sustain before average latency is greater than 1ms.

Oracle Automatic Workload Repository and benchmarking

The gold standard for Oracle performance comparisons is an Oracle Automatic Workload Repository (AWR) report.

There are multiple types of AWR reports. From a storage point of view, a report generated by running the `awrrpt.sql` command is the most comprehensive and valuable because it targets a specific database instance and includes some detailed histograms that break down storage I/O events based on latency.

Comparing two performance arrays ideally involves running the same workload on each array and producing an AWR report that precisely targets the workload. In the case of a very long-running workload, a single AWR report with an elapsed time that encompasses the start and stop time can be used, but it is preferable to break out the AWR data as multiple reports. For example, if a batch job ran from midnight to 6 a.m., create a series of one-hour AWR reports from midnight–1 a.m., 1 a.m.–2 a.m., and so on.

In other cases, a very short query should be optimized. The best option is an AWR report based on an AWR snapshot created when the query begins and a second AWR snapshot created when the query ends. The database server should be otherwise quiet to minimize the background activity that would obscure the activity of the query under analysis.

Where AWR reports are not available, Oracle statspack reports are a good alternative. They contain most of the same I/O statistics as an AWR report.

Oracle AWR and troubleshooting

An AWR report is also the most important tool for analyzing a performance problem.

As with benchmarking, performance troubleshooting requires that you precisely measure a particular workload. When possible, provide AWR data when reporting a performance problem to the NetApp support center or when working with a NetApp or partner account team about a new solution.

When providing AWR data, consider the following requirements:

- Run the `awrrpt.sql` command to generate the report. The output can be either text or HTML.
- If Oracle Real Application Clusters (RACs) are used, generate AWR reports for each instance in the cluster.
- Target the specific time the problem existed. The maximum acceptable elapsed time of an AWR report is generally one hour. If a problem persists for multiple hours or involves a multihour operation such as a batch job, provide multiple one-hour AWR reports that cover the entire period to be analyzed.
- If possible, adjust the AWR snapshot interval to 15 minutes. This setting allows a more detailed analysis to be performed. This also requires additional executions of `awrrpt.sql` to provide a report for each 15-minute interval.
- If the problem is a very short running query, provide an AWR report based on an AWR snapshot created when the operation begins and a second AWR snapshot created when the operation ends. The database server should be otherwise quiet to minimize the background activity that would obscure the activity of the operation under analysis.
- If a performance problem is reported at certain times but not others, provide additional AWR data that demonstrates good performance for comparison.

calibrate_io

The `calibrate_io` command should never be used to test, compare, or benchmark storage systems. As stated in the Oracle documentation, this procedure calibrates the I/O capabilities of storage.

Calibration is not the same as benchmarking. The purpose of this command is to issue I/O to help calibrate database operations and improve their efficiency by optimizing the level of I/O issued to the host. Because the type of I/O performed by the `calibrate_io` operation does not represent actual database user I/O, the results are not predictable and are frequently not even reproducible.

SLOB2

SLOB2, the Silly Little Oracle Benchmark, has become the preferred tool for evaluating database performance. It was developed by Kevin Closson and is available at https://kevinclosson.net/slob/. It takes minutes to install and configure, and it uses an actual Oracle database to generate I/O patterns on a user-definable tablespace. It is one of the few testing options available that can saturate an all-flash array with I/O. It is also useful for generating much lower levels of I/O to simulate storage workloads that are low IOPS but latency sensitive.

Swingbench

Swingbench can be useful for testing database performance, but it is extremely difficult to use Swingbench in a way that stresses storage. NetApp has not seen any tests from Swingbench that yielded enough I/O to be a significant load on any AFF array. In limited cases, the Order Entry Test (OET) can be used to evaluate storage from a latency point of view. This could be useful in situations where a database has a known latency dependency for particular queries. Care must be taken to make sure that the host and network are properly configured to realize the latency potentials of an all-flash array.

HammerDB

HammerDB is a database testing tool that simulates TPC-C and TPC-H benchmarks, among others. It can take a lot of time to construct a sufficiently large data set to properly execute a test, but it can be an effective tool for evaluating performance for OLTP and data warehouse applications.

Orion

The Oracle Orion tool was commonly used with Oracle 9, but it has not been maintained to ensure compatibility with changes in various host operation systems. It is rarely used with Oracle 10 or Oracle 11 due to incompatibilities with OS and storage configuration.

Oracle rewrote the tool, and it is installed by default with Oracle 12c. Although this product has been improved and uses many of the same calls that a real Oracle database uses, it does not use precisely the same code path or I/O behavior used by Oracle. For example, most Oracle I/Os are performed synchronously, meaning the database halts until the I/O is complete as the I/O operation completes in the foreground. Simply flooding a storage system with random I/Os is not a reproduction of real Oracle I/O and does not offer a direct method of comparing storage arrays or measuring the effect of configuration changes.

That said, there are some use cases for Orion, such as general measurement of the maximum possible performance of a particular host-network-storage configuration, or to gauge the health of a storage system. With careful testing, usable Orion tests could be devised to compare storage arrays or evaluate the effect of a configuration change so long as the parameters include consideration of IOPS, throughput, and latency and attempt to faithfully replicate a realistic workload.

Stale NFSv3 locks

If an Oracle database server crashes, it might have problems with stale NFS locks upon restart. This problem is avoidable by paying careful attention to the configuration of name resolution on the server.

This problem arises because creating a lock and clearing a lock use two slightly different methods of name resolution. Two processes are involved, the Network Lock Manager (NLM) and the NFS client. The NLM uses `uname n` to determine the host name, while the `rpc.statd` process uses `gethostbyname()`. These host names must match for the OS to properly clear stale locks. For example, the host might be looking for locks owned by `dbserver5`, but the locks were registered by the host as `dbserver5.mydomain.org`. If `gethostbyname()` does not return the same value as `uname –a`, then the lock release process did not succeed.

The following sample script verifies whether name resolution is fully consistent:

```
#! /usr/bin/perl
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\n";
print "gethostbyname yields: $name\n";
```

If `gethostbyname` does not match `uname`, stale locks are likely. For example, this result reveals a potential problem:

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

The solution is usually found by changing the order in which hosts appear in `/etc/hosts`. For example, assume that the hosts file includes this entry:

```
10.156.110.201  dbserver5.mydomain.org dbserver5 loghost
```

To resolve this issue, change the order in which the fully qualified domain name and the short host name appear:

```
10.156.110.201  dbserver5 dbserver5.mydomain.org loghost
```

`gethostbyname()` now returns the short `dbserver5` host name, which matches the output of `uname`. Locks are thus cleared automatically after a server crash.

WAFL alignment verification

Correct WAFL alignment is critical for good performance. Although ONTAP manages blocks in 4KB units, this fact does not mean that ONTAP performs all operations in 4KB units. In fact, ONTAP supports block operations of different sizes, but the underlying accounting is managed by WAFL in 4KB units.

The term “alignment” refers to how Oracle I/O corresponds to these 4KB units. Optimum performance requires an Oracle 8KB block to reside on two 4KB WAFL physical blocks on a drive. If a block is offset by 2KB, this block resides on half of one 4KB block, a separate full 4KB block, and then half of a third 4KB block. This arrangement causes performance degradation.

Alignment is not a concern with NAS file systems. Oracle datafiles are aligned to the start of the file based on the size of the Oracle block. Therefore, block sizes of 8KB, 16KB, and 32KB are always aligned. All block operations are offset from the start of the file in units of 4 kilobytes.

LUNs, in contrast, generally contain some kind of driver header or file system metadata at their start that creates an offset. Alignment is rarely a problem in modern OSs because these OSs are designed for physical drives that might use a native 4KB sector, which also requires I/O to be aligned to 4KB boundaries for optimum performance.

There are, however, some exceptions. A database might have been migrated from an older OS that was not optimized for 4KB I/O, or user error during partition creation might have led to an offset that is not in units of 4KB in size.

The following examples are Linux-specific, but the procedure can be adapted for any OS.

Aligned

The following example shows an alignment check on a single LUN with a single partition.

First, create the partition that uses all partitions available on the drive.

```
[root@host0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240):
Using default value 10240
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@host0 iscsi]#
```

The alignment can be checked mathematically with the following command:

```
[root@host0 iscsi]# fdisk -u -l /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              32    20971519    10485744   83  Linux
```

The output shows that the units are 512 bytes, and the start of the partition is 32 units. This is a total of 32 x 512 = 16,834 bytes, which is a whole multiple of 4KB WAFL blocks. This partition is correctly aligned.

To verify correct alignment, complete the following steps:

1. Identify the universally unique identifier (UUID) of the LUN.

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0
              Vserver Name: jfs
                  LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
                    Mapped: mapped`                `
```

2. Enter the node shell on the ONTAP controller.

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found.  Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by NetApp
         personnel.
```

3. Start statistical collections on the target UUID identified in the first step.

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffff08b9536188'
FAS8040SAP-02*>
```

4. Perform some I/O. It is important to use the `iflag` argument to make sure that I/O is synchronous and not buffered.

> Be very careful with this command. Reversing the `if` and `of` arguments destroys data.

```
[root@host0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000
bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. Stop the stats and view the alignment histogram. All I/O should be in the `.0` bucket, which indicates I/O that is aligned to a 4KB block boundary.

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
```

Misaligned

The following example shows misaligned I/O:

1. Create a partition that does not align to a 4KB boundary. This is not default behavior on modern OSs.

```
[root@host0 iscsi]# fdisk -u /dev/sdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519):
Using default value 20971519
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

2. The partition has been created with a 33-sector offset instead of the default 32. Repeat the procedure outlined in Aligned. The histogram appears as follows:

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

The misalignment is clear. The I/O mostly falls into the* *`.1` bucket, which matches the expected offset. When the partition was created, it was moved 512 bytes further into the device than the optimized default, which means that the histogram is offset by 512 bytes.

Additionally, the `read_partial_blocks` statistic is nonzero, which means I/O was performed that did not fill up an entire 4KB block.

Redo logging

The procedures explained here are applicable to datafiles. Oracle redo logs and archive logs have different I/O patterns. For example, redo logging is a circular overwrite of a single file. If the default 512-byte block size is used, the write statistics look something like this:

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-
9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

The I/O would be distributed across all histogram buckets, but this is not a performance concern. Extremely high redo-logging rates might, however, benefit from the use of a 4KB block size. In this case, it is desirable to make sure that the redo-logging LUNs are properly aligned. However, this is not as critical to good performance as datafile alignment.

Copyright information

Trademark information