Technical Report

Electronic Design Automation best practices in ONTAP

Justin Parisi, NetApp
April 2025 | TR-4617

Abstract

This document highlights best practices and implementation tips in NetApp® ONTAP® for Electronic Design Automation (EDA) workloads. It also calls attention to NetApp FlexGroup volumes, which are ideal for handling the high metadata overhead in EDA environments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Overview

NetApp ONTAP

NetApp ONTAP is a data management software solution that offers the following benefits.

Performance

Scale your cluster up by adding larger, more powerful nodes. Scale your cluster out by providing more compute and capacity to Electronic Design Automation (EDA) workloads that can grow in number of nodes rapidly or do both—all nondisruptively. With ONTAP, you can grow your performance needs as your application grows, while providing a single namespace that can deliver millions of IOPS for your workloads.

Flexibility

Provision unified storage with SAN and NAS connectivity, with the ability to standardize data management across flash, disk, and cloud. Deploy ONTAP on NetApp hardware, with software-defined solutions like ONTAP Select, or deploy in the cloud with Cloud Volumes ONTAP. ONTAP provides data access anywhere, anytime.

Resiliency and high availability

Leverage patented RAID technologies, including Triple Erasure Coding (RAID-TEC) for extra protection against drive failures, particularly with larger drives that have longer rebuild times. Additionally, ONTAP high-availability, active-active controller failover means minimal downtime in the event of planned or unplanned outages. Storage stacks are also connected using the latest multipath I/O technology for both redundancy and performance.

Scalability

HA pairs can be clustered together to form a single NAS namespace (up to 24 nodes) or SAN target (up to 12 nodes). Scale up by adding larger controllers and more disk to storage stacks to provide more capacity or performance. Scale out by adding additional HA pairs and disks to existing clusters, nondisruptively. Deploy massive storage containers with new NetApp ONTAP FlexGroup volume technology. Localize read-heavy workloads to remote sites and the cloud with NetApp FlexCache® volumes. Automate load and performance balancing with the latest ONTAP releases and their feature sets.

Efficiency

Deliver multiple efficiency features to allow administrators to squeeze the most out of their existing storage. Inline deduplication, compression, and data compaction let you shrink your data footprint as data is ingested. Inline aggregate deduplication enables deduplication across multiple volumes in the same aggregate. Thin provisioning and NetApp FlexClone® technology offer administrators flexible storage use without taking up valuable capacity.

Security

Implement up-to-date security enhancements, such as Federal Information Processing Standard (FIPS) 140-2 compliant data encryption technologies and AES-256 encryption for Kerberos in SMB and NFS, as well as NetApp Volume Encryption (NVE) and NetApp Storage Encryption drives (NSE) for encryption at rest.

For more information ,see the SemiWiki blog about data security in EDA using NetApp: NetApp Enables Secure B2B Data Sharing for the Semiconductor Industry.

Container and orchestration integration

Use NetApp Trident, an open-source project that NetApp maintains for application container persistent storage. Trident has been implemented as an external provisioner controller that runs as a pod itself, monitoring volumes and completely automating the provisioning process. Trident builds upon NetApp's 32 plus years of experience and is fully supported by NetApp. Trident integrates fully with NetApp ONTAP and fits well in EDA workload environments that are looking towards containers for compute.

Cloud enablement

ONTAP data management capabilities allow storage administrators to move in and out of the cloud quickly and efficiently. NetApp SnapMirror®, Cloud Volumes ONTAP, NetApp Cloud Volumes Service, and storage tiering to the cloud with ONTAP FabricPool offer multiple ways to leverage cloud infrastructures for enterprise needs – whether it is cloud native or a hybrid cloud approach. ONTAP provides a cloud offering in Google GCP, Amazon AWS, and Microsoft Azure.

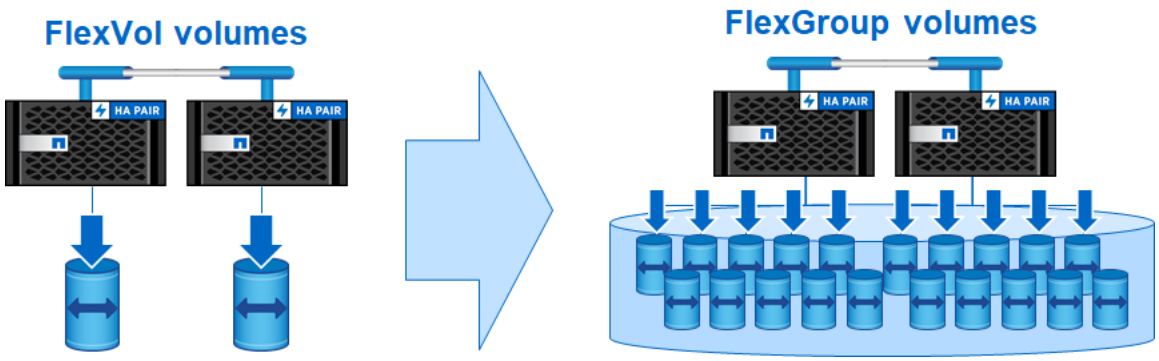
NetApp FlexGroup volumes

NetApp FlexVol® volumes have traditionally been a good fit with EDA workloads. However, as hard-drive costs are driven down and flash hard-drive capacity grows exponentially, file systems are following suit. The days of file systems that number in the tens of gigabytes are over. Storage administrators face increasing demands from application owners for large buckets of capacity with enterprise-level performance.

Big data frameworks such as Hadoop and EDA workloads, in which storage needs for a single namespace can extend into the petabyte range (with potentially billions of files), are becoming more prevalent. NetApp FlexGroup is the ideal solution for these architectures. SemiWiki touted NetApp FlexGroup volumes as "a game changer" in their recent NetApp's FlexGroup Volumes – A Game Changer for EDA Workflows blog, and said "with the higher performance and the ability to scale capacity transparently, it means that the most precious resource of an EDA design cycle, more time, is now available to be used in whatever way is most beneficial."

With FlexGroup volumes, a storage administrator can easily provision a massive single namespace in a matter of seconds. FlexGroup volumes have virtually no capacity or file count constraints outside of the physical limits of hardware and the total volume limits of ONTAP. Limits are determined by the overall number of constituent member volumes that work in collaboration to dynamically balance load and space allocation evenly across all members. There is no required maintenance or management overhead with a FlexGroup volume. You simply create the volume and share it with your NAS clients. ONTAP does the rest.

Figure 1) Evolution of NAS file systems in ONTAP.



Advantages of NetApp FlexGroup volumes

NetApp FlexGroup volumes provide some distinct advantages over FlexVol volumes.

Massive capacity and predictable low latency for high-metadata workloads

Previously, NetApp ONTAP technology did not have a solution for the need for high capacity beyond 100TiB combined with enterprise-level performance. Earlier versions are constrained by architectural limitations and the notion of volume affinity: the tendency of ONTAP operations, particularly metadata operations, to operate in a single serial CPU thread.

The FlexGroup feature solves this problem by automatically balancing ingest workloads across multiple constituent FlexVol members to provide multiple affinities to handle high-metadata workloads.

Flexible, nondisruptive growth

When NAS storage containers reach their hard limits of 300TiB, FlexGroup volumes provide a way to grow without needing to take a maintenance window, delete data or copy large amounts of data to new locations. With a FlexGroup volume, if you hit a capacity limit, simply add more disk or nodes to the cluster and expand across the new hardware.

Additionally, if you have existing FlexVol volumes that you would like to move to FlexGroup, ONTAP 9.7 and later provides a minimally disruptive way to convert volumes in-place without needing to copy data.

Efficient use of all cluster hardware

Previously, file systems in ONTAP were tied to a single FlexVol container, which, in turn, was limited to a single node's hardware resources in a cluster. Although it was possible to scale multiple FlexVol volumes across multiple nodes in a cluster, the management overhead was cumbersome, and the process did nothing to increase the total capacity of a single namespace. To achieve this type of scale, volumes could be junctioned to one another. FlexGroup volumes provide a way to scale across all nodes in a cluster and provide a true scale-out solution for workloads that require a single namespace.

Simple, easy-to-manage architecture and balancing

To achieve scale beyond the single node or aggregate that owns the FlexVol volume, several volumes had to be junctioned to one another. This concept required design, architecture, and management overhead that took valuable time away from storage administrators' day-to-day operations. A FlexGroup volume can provision storage across every node and aggregate in a cluster in less than a minute in NetApp ONTAP System Manager.

Truly global namespace

NetApp FlexGroup volumes provide a way to present a large single namespace across an ONTAP cluster. But what if you want a truly global namespace – one that can serve read-writeable datasets across multiple sites across the globe?

NetApp FlexCache volumes can act as remote, virtual sparse caches to localize data reads in the data center, at the edge or in the cloud.

Superior density

A FlexGroup volume lets you condense copious amounts of data into smaller data center footprints by using the superb storage efficiency features of ONTAP, including the following:

- Thin provisioning
- Inline data compaction, data compression, and deduplication
- Cross-volume and post-process deduplication

- FlexClone volumes
- FlexCache volumes
- Cloud-tiering with FabricPool

In addition, ONTAP supports dense solid-state drives (SSDs), which can deliver hundreds of terabytes of raw capacity in a single 24-drive enclosure.

NetApp FlexGroup volumes: SpecStorage Solution 2020_EDA_Blended

NetApp recently posted results for the SPECstorage Solution 2020 EDA_Blended benchmark, which focuses on simulating the EDA process from end to end. A-series controllers, using FlexGroup volumes, delivered predictable and consistent high performance.

Notably, the 8-node A90 used 60% less rack space to deliver 28% more jobs than the record set previously by the A900. These results can be scaled linearly by adding additional nodes to the cluster.

For more information on the results see:

- https://www.netapp.com/blog/accelerate-eda-builds-by-an-order-of-magnitude
- https://www.spec.org/storage2020/results/swbuild.html

NetApp FlexGroup volumes: Powering NetApp's own workloads

One of the truest tests of a software feature is this: Does the creator of the software use its own features?

The answer to this question is a resounding yes. NetApp leverages FlexGroup volumes in its own development environment, in NetApp Active IQ® data lakes, and for use with numerous other workload use cases.

For more information about how NetApp uses FlexGroup volumes for Active IQ see:

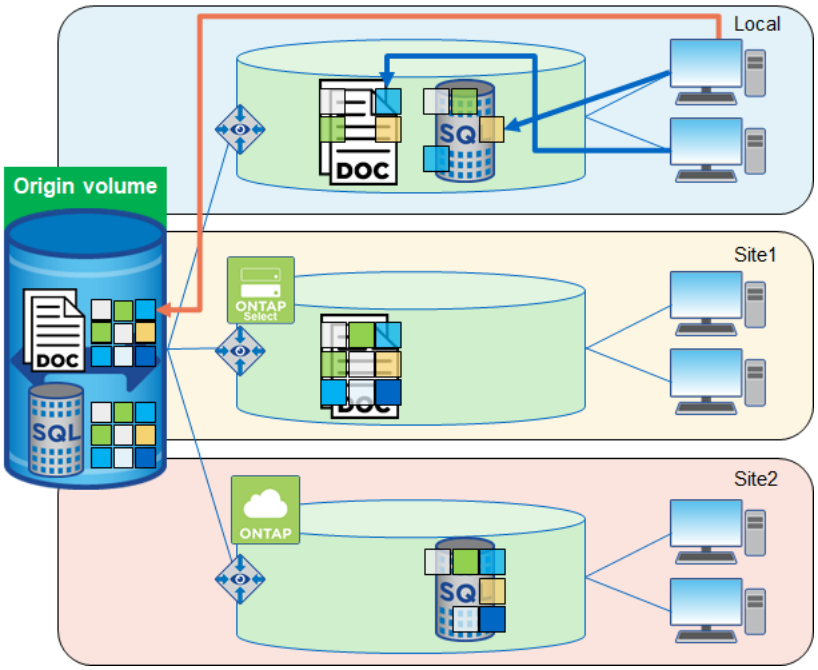
- ONTAP FlexGroup Technology Powers NetApp's Massive Active IQ Data Lake
- Tech ONTAP Podcast Episode 182: NetApp on NetApp – FlexGroups and Active IQ

NetApp FlexCache volumes

FlexCache in ONTAP provides a writable, persistent cache of a volume in a remote place that is consistent, coherent, and current.

A cache is a temporary storage location that resides between a host and a source of data. The objective of a cache is to store frequently accessed portions of source data in a way that allows the data to be served faster than it would be by fetching the data from the source. Caches are most beneficial in read-intensive environments where data is accessed more than once and is shared by multiple hosts.

Figure 2) NetApp FlexCache volumes.



A cache can serve data faster in one of two ways:

- The cache system is faster than the system with the data source. This can be achieved through faster storage (for example, SSD versus HDD, increased processing power, or increased (or faster) memory in the platform that serves the cache.
- The storage space for the cache is physically closer to the host, so it does not take as long to reach the data.

Caches are implemented with different architectures, policies, and semantics so that the integrity of the data is protected as it is stored in the cache and served to the host.

FlexCache offers the following benefits:

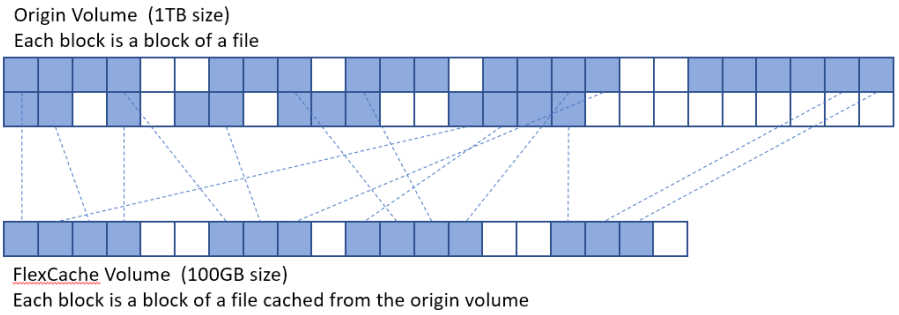
- Improved performance by providing load distribution
- Reduced latency by locating data closer to the point of client access
- Enhanced availability by serving cached data in a network disconnection situation

FlexCache provides all of the above advantages while maintaining cache coherency, data consistency, data currency, and efficient use of storage in a scalable and high-performing manner.

A FlexCache is a sparse container; not all files from the origin dataset are cached, and, even then, not all data blocks of a cached inode can be present in the cache. Storage is used efficiently by prioritizing retention of the working dataset (recently used data).

With FlexCache, the management of disaster recovery and other corporate data strategies only needs to be implemented at the origin. Because data management is only on the source, FlexCache enables better and more efficient use of resources and simpler data management and disaster recovery strategies. For EDA workloads, SemiWiki discusses how FlexCache volumes offer a way for geographically dispersed design teams to stay in synchronization with a current cache of the working dataset in Concurrency and Collaboration – Keeping a Dispersed Design Team in Sync with NetApp.

Figure 3) Sparse volume details.



Use cases

FlexCache in ONTAP offers the most benefit in specific use cases, and those specific use cases are listed as “ideal.” Other use cases for a FlexCache volume are possible but the benefits have not been fully vetted. In most instances, the use case is limited to the supported feature set. Non-ideal use cases are not discouraged, but you should compare the benefits of FlexCache to the costs associated with the non-ideal use case.

Ideal use cases

Prior to ONTAP 9.15.1, FlexCache was limited to write-around model, which works better with read heavy workloads. In ONTAP 9.15.1, FlexCache introduced write-back. This can improve write performance for workloads that are latency sensitive.

Best practice 1: FlexCache proof of concept
Prior to deploying FlexCache in a production environment, it is essential for customers to perform a proof of concept. Each workload has unique characteristics, and it is critical to ascertain the appropriate method for handling writes, whether it be write-around or write-back. Selecting an incorrect method may result in suboptimal application performance.

Some examples include, but are not limited to, the following:

- EDA
- Media rendering
- Artificial Intelligence (AI), machine learning (ML), and deep learning (DL) workloads
- Unstructured NAS data such as home directories
- Software-build environments such as Git
- Common tool distribution
- Hot volume performance balancing
- Cloud bursting, acceleration, and caching
- Stretched NAS volumes across NetApp MetroCluster configurations

NetApp FabricPool

NetApp FabricPool, first available in ONTAP 9.2, is a NetApp technology that enables automated tiering of data to low-cost object storage tiers either on or off premises.

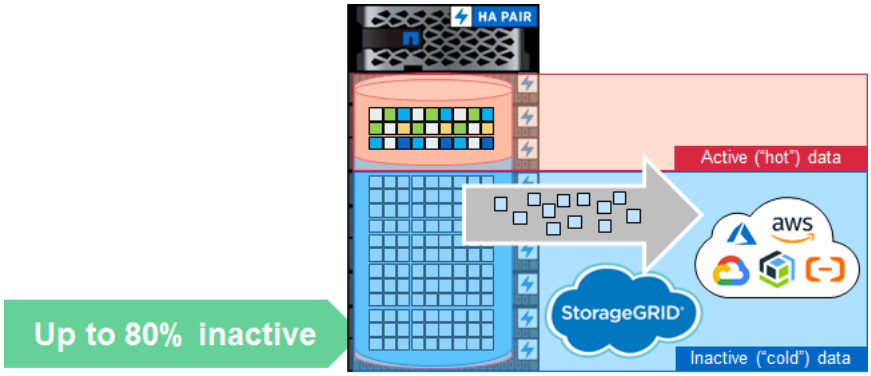
Unlike manual tiering solutions, FabricPool reduces total cost of ownership by automating the tiering of data to lower the cost of storage. It delivers the benefits of cloud economics by tiering to public clouds such as Alibaba Cloud Object Storage Service, Amazon S3, Google Cloud Storage, IBM Cloud Object

Storage, and Microsoft Azure Blob Storage, as well as to private clouds such as ONTAP S3 or NetApp StorageGRID®.

FabricPool is transparent to applications and allows enterprises to take advantage of cloud economics without sacrificing performance or having to rearchitect solutions to leverage storage efficiency.

- ONTAP supports FabricPool on SSD and HDD local tiers (also known as storage aggregates in the ONTAP CLI). NetApp Flash Pool aggregates are not supported.
- ONTAP Select supports FabricPool. NetApp recommends using all-SSD FabricPool local tiers.
- Cloud Volumes ONTAP supports data tiering with Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage.

Figure 4) NetApp FabricPool.



Use cases

The primary purpose of FabricPool is to reduce storage footprints and associated costs. Active data remains on high-performance local tiers, and inactive data is tiered to low-cost object storage while preserving ONTAP functionality and data efficiencies.

FabricPool has two primary use cases:

- Reclaim capacity on primary storage
- Shrink the secondary storage footprint

Although FabricPool can significantly reduce storage footprints in primary and secondary data centers, it is not a backup solution. Access control lists (ACLs), directory structures, and NetApp WAFL® metadata always stay on the local tier. If a catastrophic disaster destroys the local tier, you cannot create a new environment by using the data on the cloud tier because it contains no WAFL metadata.

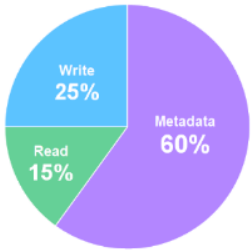
For complete data protection, consider using an existing ONTAP feature such as SnapMirror®

For more information about FabricPool, see TR-4598: FabricPool Best Practices.

EDA workloads

EDA workloads present a unique set of challenges to storage systems, mainly due to the massive capacity, high file count and heavy metadata operations, and high-performance requirements for manufacturers that need to continually ship product to stay competitive in their business silos. Simplicity and usability in these environments are required because administrators need to focus on supporting the application and its users, rather than on managing complex storage architectures. ONTAP can help address these EDA workload challenges with a multifaceted solution.

Figure 5) Workload types: EDA.



Capacity

NetApp FlexVol volumes provide up to 300TB of space in a single container. However, EDA workloads might need more than that amount in some instances. FlexGroup volumes offer a multipetabyte container in a single namespace for EDA workloads over NAS protocols that can scale up or scale out nondisruptively as the dataset grows.

High-file-count environments

NetApp FlexVol volumes support up to 2 billion files in a single container. In some cases, that amount might not be enough. EDA file system layouts can contain thousands of files per directory, with deep directory structures. NetApp FlexGroup volumes can increase file counts exponentially across multiple member volumes and nodes in a cluster to provide containers that can contain file counts in the hundreds of billions.

Performance

NetApp FlexGroup volumes provide multithreaded parallel operations for high-file-count, metadata-heavy workloads, such as EDA. By spreading the ingest load across multiple FlexVol member volumes, multiple network interfaces, and multiple cluster nodes, NetApp FlexGroup volumes can deliver high throughput and IOPS at predictable, low latencies that still perform well at scale. Do you need to scale out performance? Add more nodes to the cluster nondisruptively. In addition, using ONTAP with flash optimizations in NetApp AFF can improve performance and density for EDA workloads. For more information about FlexGroup volume performance, see “Performance”, which describes the performance for EDA workloads in different scenarios.

Simplicity

NetApp FlexGroup volumes blend capacity, high-file-count handling, and performance with a simple, easy-to-deploy container under a single NAS namespace. Data ingestion and load balancing are handled automatically by the ONTAP subsystems used by FlexGroup volumes, with no need to worry about whether data is being placed locally or remotely. For more information, see “NetApp FlexGroup volumes.”

Performance

This section describes some real-world results from an EDA environment, as well as some results from internal EDA workload benchmarking.

Real-world EDA performance testing

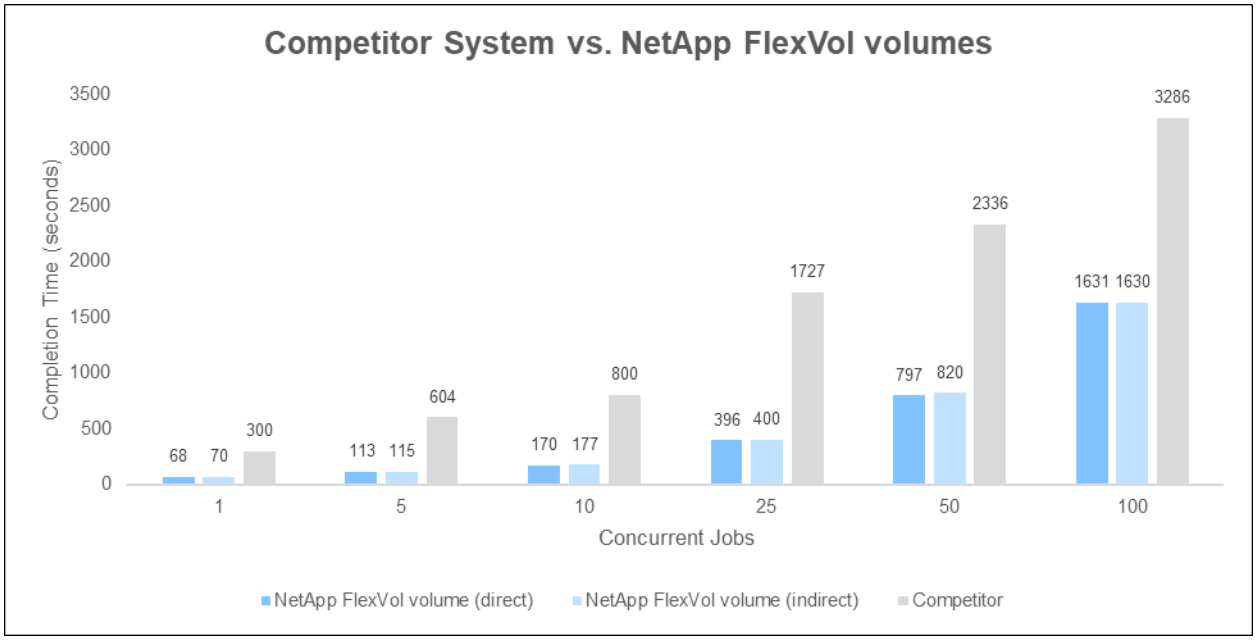
The best way to show that a system can handle your workload is by testing that workload. However, not everyone is afforded the luxury of being able to run tests. This section provides the real-world EDA testing data that was collected by using specific tests from a chip design manufacturer.

EDA customer benchmarking – ONTAP

In the early days of FlexGroup volumes, an EDA customer decided to compare the performance on a battery of real-world tests using a sixteen node NetApp AFF ONTAP cluster against a competitor system.

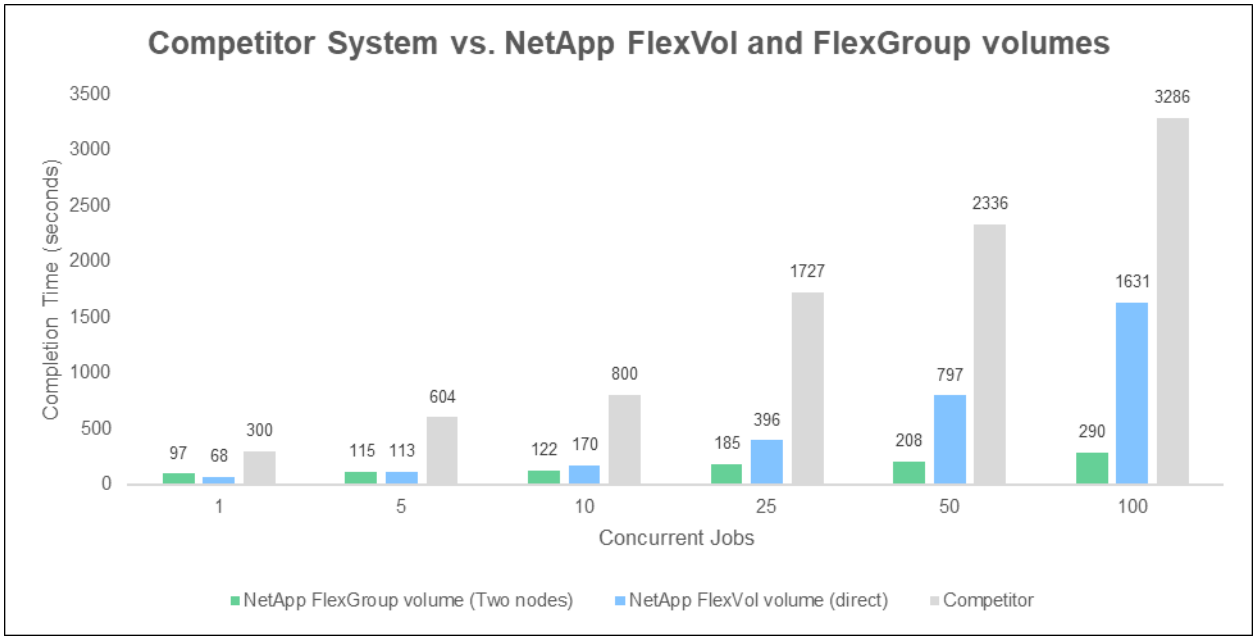
The first set of tests use a kernel extract and compare the competitor system using twenty-one nodes against a NetApp AFF system using FlexVol volumes. In one test, the FlexVol volume used indirect access (as in, clients were attached to a network interface on a separate node than where the volume resided). In another test, the FlexVol volume used direct access (network connection on the same node as the volume). Then, the number of concurrent jobs was increased in increments. In all cases, the NetApp AFF system using FlexVol volumes exceeded the performance of the competitor system. In most of the tests, the local FlexVol volume performed slightly better than the indirectly accessed FlexVol volume, but not considerably so.

Figure 6) Kernel extract: Competitor versus NetApp FlexVol volumes.



The same tests were used to compare the competitor system and FlexVol volume performance with NetApp FlexGroup volumes, which can leverage more hardware resources for these high metadata workloads. The FlexGroup volumes scaled across two of the nodes in the sixteen-node cluster. As you can see following, the FlexGroup volume greatly outperformed even the locally accessed FlexVol volume on the same system—especially as the concurrent jobs increased.

Figure 7) Kernel extract: Competitor versus NetApp FlexVol volumes and NetApp FlexGroup volumes.



The kernel extract test was also used to show the benefits of scaling a FlexGroup volume across more nodes in a single cluster. Figure 8 shows the NetApp FlexGroup volume outperforming the competitor system's twenty-one node cluster with a fraction of the hardware needed and performing even better as the concurrent jobs scale to the point where a single node's resources start to be exhausted.

Figure 8) Kernel extract: Competitor versus NetApp FlexGroup volumes; Scale-out.

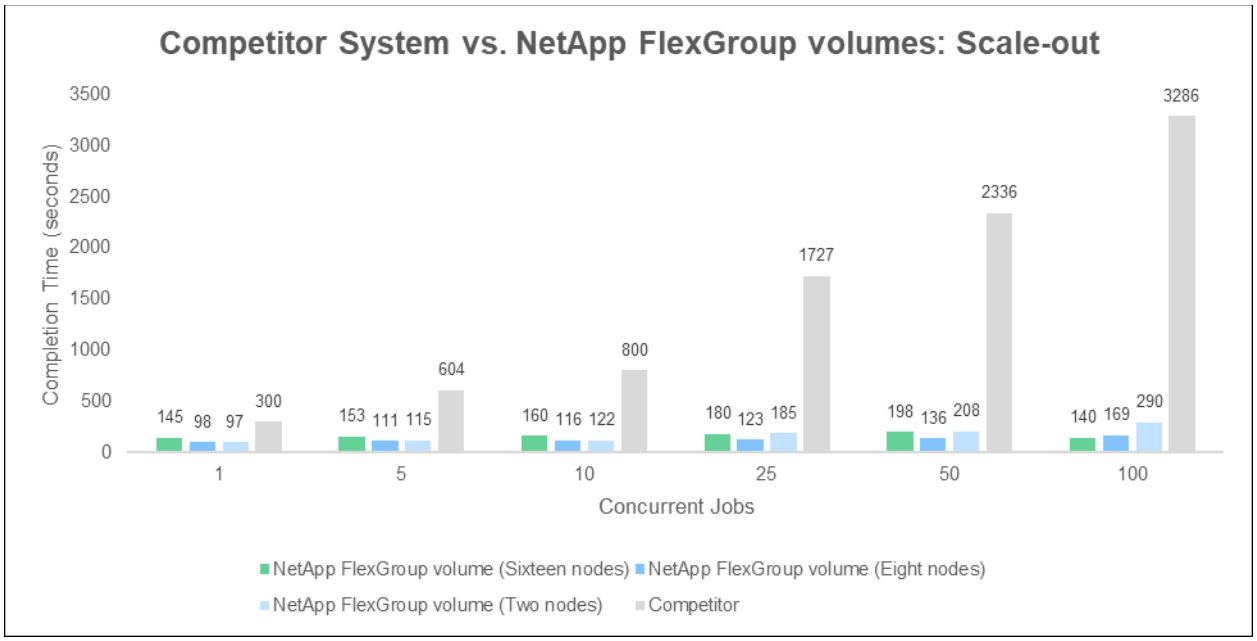
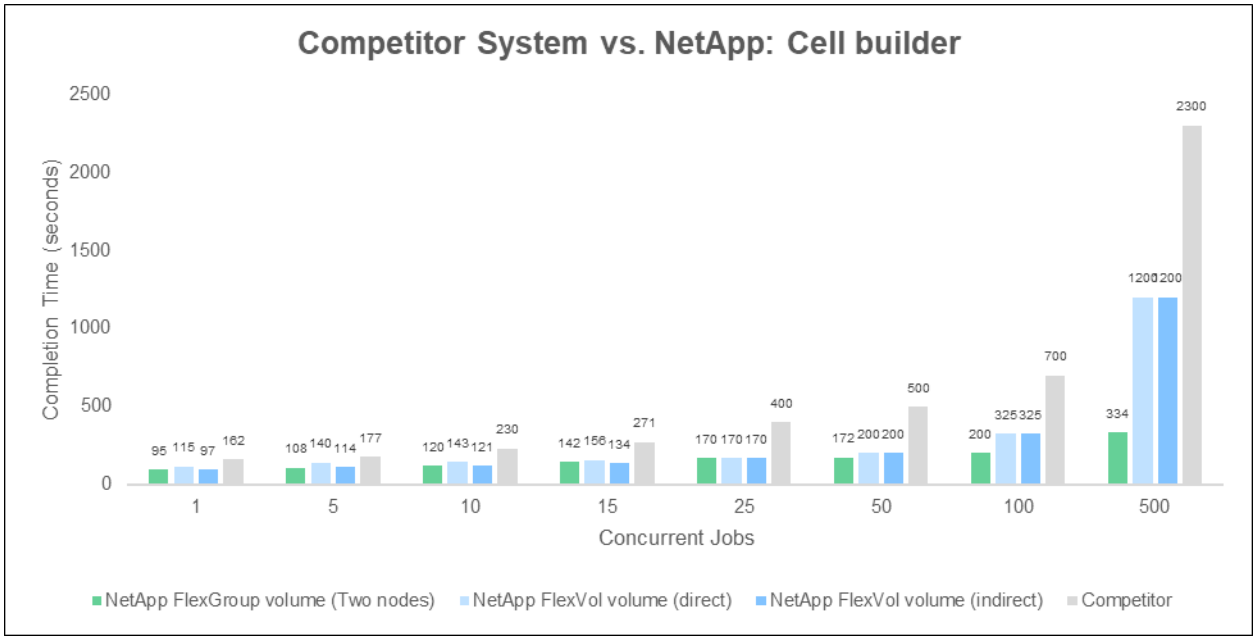


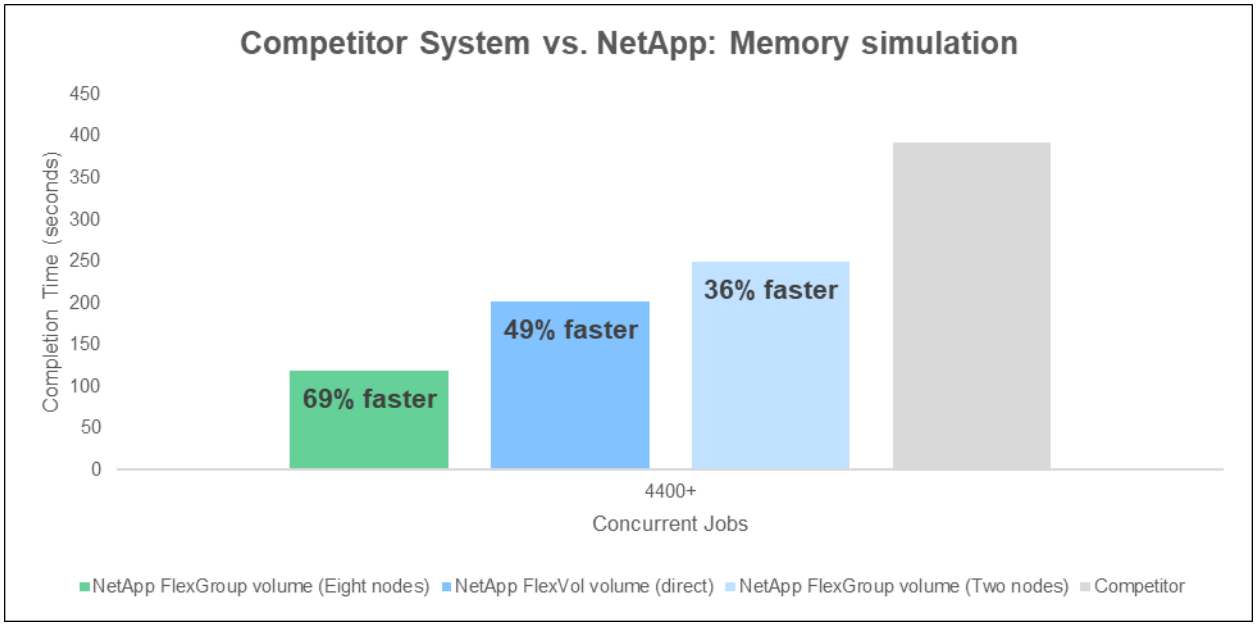
Figure 9 shows completion times for a cell builder workload, again comparing FlexVol volumes, FlexGroup volumes, and a competitor system. Again, note the lower completion times for the NetApp FlexGroup volume.

Figure 9) EDA workload: Cell builder.



Memory simulation and validation tests are also a common EDA workload. Figure 10 shows the average run time for 4,400+ concurrent jobs. A FlexGroup volume across eight nodes was 69% faster than the competitor system across twenty-one nodes for this test.

Figure 10) EDA workload: Memory simulation and validation.



EDA customer benchmarking – ONTAP 9.7

Several releases later, the same customer ran another set of benchmarks. This time, they used the standard NAS benchmark testing suite for EDA workloads and compared eight NetApp ONTAP AFF A800 nodes using FlexGroup volumes to 14 all-flash nodes of a leading competitor system (Table 1).

The maximum results showed a sizable difference in performance and scale—more concurrent jobs, lower latencies, and higher IOPS/throughput.

Table 1) NetApp FlexGroup volumes versus competitor system: Standard NAS EDA benchmark.

Competitor maximum results (14 nodes)	NetApp ONTAP 9.7 maximum results (eight nodes)
• 624 concurrent jobs • ~6.8ms latency • 259,664 achieved ops • 4.36GBps	• **2,000** concurrent jobs • **~2.6ms** latency • **897,241** achieved ops • **15.6**GBps

Figure 11) Customer EDA benchmark: Latency versus achieved IOPS.

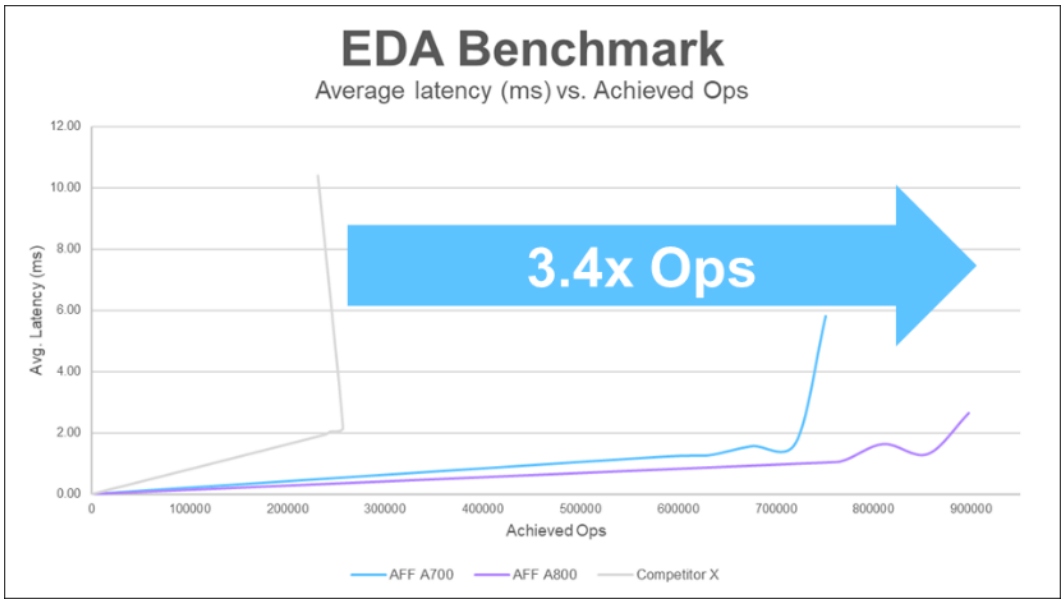
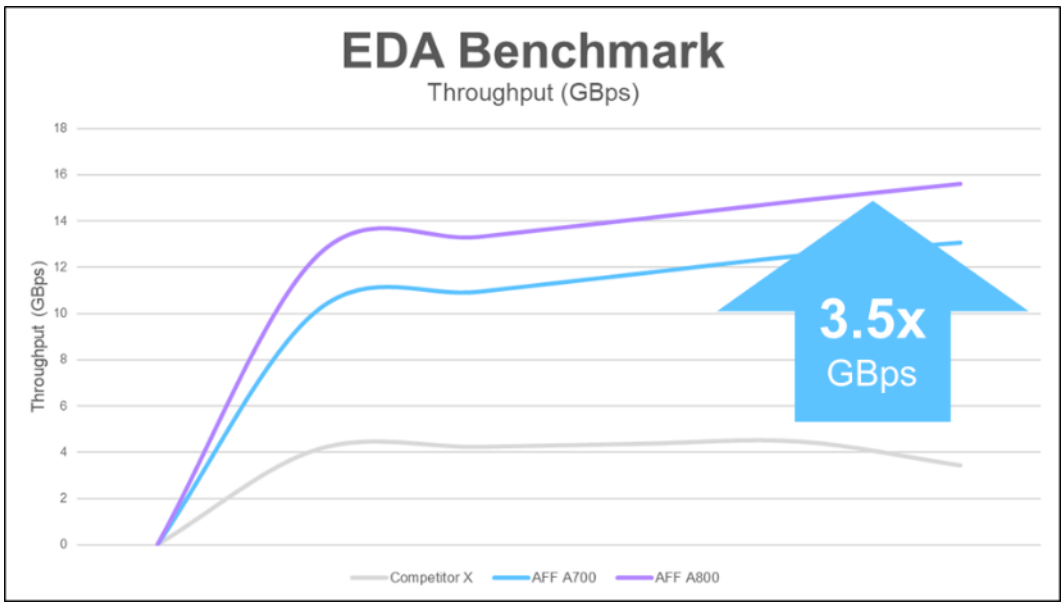


Figure 12) Customer EDA benchmark: Throughput (GBps).



ONTAP best practices for EDA

This section covers ONTAP best practices for EDA environments. Although FlexGroup volumes are a more natural fit for the types of workloads EDA throws at a storage system, this section also covers FlexVol volumes, because FlexGroup volumes might be missing features or functionality needed for specific EDA environments.

Hardware considerations

EDA workloads perform best when the following storage hardware conditions are met:

- Large memory/RAM footprint
- Greater number of cores/CPU for concurrent processing
- Large capacities

NetApp highly recommends using all-flash storage platforms with guidance from your NetApp sales teams (midrange, such as the AFF A50, or higher end platforms such as the AFF A90 or AFF A1K) to maximize the available RAM and CPU in each node.

All-flash storage provides the following benefits to EDA workloads:

- Higher density capacity and lower power consumption for smaller rack footprints.
  - Smaller rack footprints also mean lower cost.
- More effective storage efficiencies due to flash-only features such as inline deduplication, inline compression, and inline data compaction.
- Longer lifespan for drives versus. spinning disk.
- Optimized performance with lower latency than serial-attached SCSI (SAS) or SATA.
- Support for NVMe connected drives.
- More I/O per drive than an entire shelf of spinning drives.

For project tiering, hot data workloads should reside on AFF systems. Cool and cold data workloads can reside on any platform and media type, including the AFF C-Series with capacity flash. Table 2 shows the CPU and RAM for AFF A-Series systems intended for hot data workloads. For more information and the latest hardware specifications, see the hardware specifications at NetApp hardware universe.

Table 2) NetApp all-flash system CPU and RAM per HA pair.

System	CPU information	RAM
AFF A1K	Four 52-core 64-bit 1.7GHz (208 total cores)	2048GB
AFF A90	Four 32-core 64-bit 2.0GHz (128 total cores)	2048GB
AFF A70	Four 16-Core 64-bit 2.0GHz (64 total cores)	256GB
AFF A50	Two 24-core 64-bit 1.9 GHz (48 total cores)	256GB
AFF A30	Two 16-core 64-bit 2.0 GHz (32 total cores)	128GB
AFF A20	Two 8-core 64-bit 2.1 GHz (16 total cores)	128GB

Being able to provide more memory and CPU for EDA workloads can have a positive effect on the completion times of these workloads, which can mean a greater return on investment, because the money saved in build times can offset the costs of more expensive nodes.

Providing additional memory and CPU for EDA workloads can improve the completion times of these tasks. This efficiency can potentially offset the costs associated with more expensive nodes by reducing build times.

Note: NetApp highly recommends engaging your NetApp sales account team to evaluate your business requirements before architecting the cluster scale-out setup in your environment.

Aggregate layout considerations

An aggregate is a collection of physical disks that are laid out into RAID groups and provide the back-end storage repositories for virtual entities such as FlexVol and FlexGroup volumes. Each aggregate is owned by a specific node and is reassigned during storage failover events.

Starting in ONTAP 9, aggregates have dedicated NVRAM partitions for consistency points to avoid scenarios in which slower or degraded aggregates cause issues on the entire node. These consistency points are also known as per-aggregate consistency points and allow mixing of disk shelf types on the same nodes for more flexibility in the design of the storage system.

Best practice 2: Aggregate usage with NetApp FlexGroup and multiple FlexVol volumes

For consistent performance when using NetApp FlexGroup volumes or multiple FlexVol volumes, make sure that the design of the FlexGroup volume or FlexVol volumes spans only aggregates with the same disk type and RAID group configurations for active workloads. For tiering of cold data, predictable performance is not as crucial, so mixing disk types or aggregates should not have a noticeable impact.

Table 3 shows NetApp's recommended best practices for aggregate layout when using FlexGroup volumes or multiple FlexVol volume layouts. Keep in mind that these practices are not hard requirements. The one-aggregate per node recommendation for AFF systems originates from disk cost with NetApp RAID-TEC and no Advanced Disk Partitioning (ADP), because you might not want to use up expensive SSD space just for parity. However, with ADP, partitions are spread across data disks, so two aggregates per node on AFF systems are better because there are more available volume affinities per node with more aggregates present.

Note: In ONTAP 9.7 and earlier, two aggregates per node provided the best results when dealing with a large number of Snapshot copy creations at any given time. ONTAP 9.8 and later improves this for single aggregates.

Table 3) Best practices for aggregate layout with NetApp FlexGroup volumes or multiple FlexVol volumes.

Spinning disk or hybrid aggregates	AFF system
Two aggregates per node	One aggregate per node (without ADP) Two aggregates per node (with ADP)

Note: Aggregates should ideally have the same number of drives and RAID groups.

Deploying a FlexGroup volume on aggregates with existing FlexVol volumes

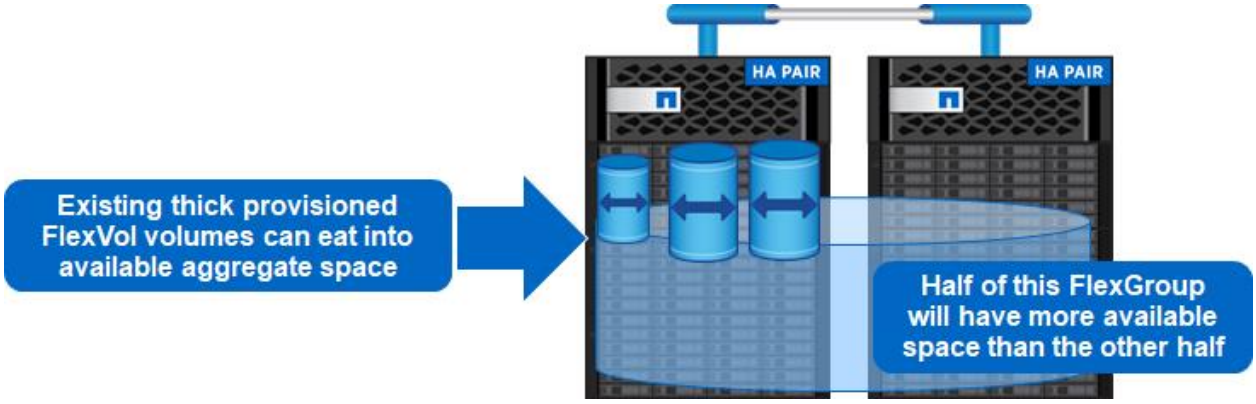
Because a FlexGroup volume can span multiple aggregates in a cluster and can coexist in the same storage virtual machine (SVM) as normal FlexVol volumes, it is possible that a FlexGroup volume might have to share an aggregate with pre-existing FlexVol volumes. Therefore, it is essential to consider the factors described in this section when you are deploying a FlexGroup volume.

Consider the capacity footprint of the existing FlexVol volumes

A FlexGroup volume can span multiple aggregates and each of those aggregates might not have the same number of FlexVol volumes on them. Therefore, the aggregates might have disparate free space that can affect the ingest distribution of a FlexGroup volume that has space guarantees disabled, because the existing FlexVol volume capacity might eat into the FlexGroup volume's capacity.

For example, if aggr1 on node1 has four FlexVol volumes at 1TB each and aggr2 on node2 has two FlexVol volumes at 1TB each, then node1's aggregate has 2TB less space than node2. If you deploy a FlexGroup volume that spans both nodes and is overprovisioned to fill both aggregates, then node1's member volumes already have "space used" in their capacity reports, which causes node2's members to absorb most of the ingest of data until the capacity used is even across all member volumes.

Figure 13) How FlexVol capacity can affect FlexGroup load distribution.



Note: This is an issue only if the FlexGroup volume is thin provisioned. Space-guaranteed FlexGroup volumes will not have other volumes eating into the space footprint. However, space-guaranteed FlexGroup volumes might not be created as large as desired if other volumes in the system prevent the space from being allocated.

Consider the performance impact of the existing FlexVol volumes

When you deploy a FlexGroup volume, it is also important to consider the amount of work the existing FlexVol volumes are doing. If a set of FlexVol volumes on one node is being hit heavily at given times, that can negatively affect the performance of a FlexGroup volume that spans the same nodes and aggregates as the existing FlexVol volumes. This is similar to the impact that can be seen with FlexVol volumes, but because a FlexGroup volume can span multiple nodes, the performance impact might appear to be intermittent from the client perspective depending on which node the data I/O is occurring on. In addition, the FlexGroup volume itself should span only homogenous hardware, as discussed in "Cluster considerations."

One way to mitigate this impact is to make use of storage quality of service (QoS) policies to help limit IOPS and throughput to those volumes or guarantee performance with QoS minimums on the FlexGroup volume. Alternately, you can use nondisruptive volume move to redistribute the volumes across nodes to balance the performance impact.

Consider the volume count limits

ONTAP has volume count limits per node that depend on the type of node in use and the personality of the node. For instance, an A1K node has higher volume count limits than a FAS9XXX series. A system with the data protection personality allows more total volumes than a system without. Additionally, there is a cluster-wide volume limit of 30,000 regardless of the node type in use that can impact how many FlexVol volumes can be provisioned per ONTAP cluster.

Because FlexGroup volumes generally contain multiple FlexVol member volumes, these member volumes count against this total limit. In addition, many ONTAP features also leverage FlexGroup volumes for their architectures. For instance, a single FlexGroup volume might use 16 member FlexVol volumes. If you use FlexClone for that volume, then you use 32 FlexVol volumes. If you create a FlexCache volume in that cluster (which is also a FlexGroup volume), then you use whatever number ONTAP selects for that cache volume.

In many cases, you will not need to create multiple small FlexGroup volumes. Instead, a larger FlexGroup volume can be provisioned and qtrees can be used to separate workloads.

Best practice 3: Deploying FlexGroup volumes with existing FlexVol volumes in place

Before deploying a FlexGroup volume, be aware of the following:

- If you have existing FlexVol volumes, verify that adding multiple FlexGroup volumes and their corresponding features to the mix will not exceed the volume count limits.
- Use the performance headroom features in NetApp Active IQ Unified Manager and ONTAP System Manager to review which nodes are being more heavily used.
- If there is an imbalance, use nondisruptive volume moves to migrate “hot” volumes to other less-used nodes to achieve as balanced a workload across nodes as possible. Note that in some cases, some EDA FlexVol volumes might be too busy to successfully move, so you might have to perform the move at another time.
- Evaluate the free space on the aggregates to be used with the FlexGroup volume and make sure that the available space is roughly equivalent.
- If the effect of volume count limit is a potential factor, create the FlexGroup volumes across nodes that have room to add more new volumes, or use nondisruptive volume moves to relocate volumes and balance out volume counts.
- Alternately, create FlexGroup volumes with fewer member volumes if volume count limits are a concern.

Networking considerations

When you use CIFS/SMB or NFS, the default is each mount point is made over a single TCP connection to a single IP address. In ONTAP, these IP addresses are attached to data LIFs, which are virtual network interfaces in an SVM.

NOTE: There are options to have multiple TCP connections per mount point. See <docs link>. For the rest of this document, we will assume these options are not in use unless noted otherwise

General networking considerations

IP addresses in ONTAP SVMs can live on a single hardware Ethernet port or multiple hardware Ethernet ports that participate in a Link Aggregation Control Protocol (LACP) or another trunked configuration. However, in ONTAP, these ports always reside on a single node, which means that they are sharing that node’s CPU, PCI bus, and so on. To help alleviate this situation, ONTAP allows TCP connections to be made to any node in the cluster, after which ONTAP redirects that request to the appropriate node through the cluster back-end network. This approach helps distribute network connections and load appropriately across hardware systems.

Best practice 4: Network design with NetApp FlexGroup volumes

FlexGroup networking best practices are similar to FlexVol networking best practices. When you design a NAS solution in ONTAP, consider the following networking best practices regardless of the volume style:

- Create at least one data LIF per node, per SVM to confirm a path to each node.
- Present multiple IP addresses to clients behind a single fully qualified domain name (FQDN) by using some form of DNS load balancing. For DNS load balancing details, see TR-4523.
- When possible, use LACP ports to host data LIFs for throughput and failover considerations.
- When you manually mount clients, spread the TCP connections across cluster nodes evenly. Otherwise, allow DNS load balancing to handle the client TCP connection distribution.
- For clients that do frequent mounts and unmounts, consider using on-box DNS to help balance the load. (If clients are not mounted and unmounted frequently, on-box DNS does not help much.)

- If the workload is that of a mount storm (that is, hundreds or thousands of clients mounting at the same time), use off-box DNS load balancing and/or consider using NetApp FlexCache volumes. A mount storm to a single node can result in a denial of service to clients or performance issues.
- If you are using NFSv4.1, consider leveraging pNFS for data localization and parallel connections to files. (pNFS works best with sequential I/O workloads; high metadata workloads might bottleneck over the single metadata server connection.)
- If you have clients that support it (such as the latest SUSE and Ubuntu clients), the nconnect mount option can provide even greater performance for NFS mounts on single clients.
- For SMB3 workloads, consider enabling the multichannel and large MTU features on the CIFS server.
- For NFSv4.1+ clients, consider enabling Session Trunking. This allows for multiple NICs to be aggregated together in an active/active setup.
- Jumbo frames are the preferred MTU size. If you are using jumbo frames on your network, ensure jumbo frames are enabled at each endpoint in the network architecture; mismatched jumbo frame configurations can introduce hard to diagnose performance issues for any volume type.
- NFS clients can get greater performance with multiple mount points from the same client connected to the same volume in ONTAP across multiple network interfaces. However, this configuration can introduce complexity. If your NFS client supports it, use Nconnect.

LACP considerations

There are valid reasons for choosing to use an LACP port on client-facing networks. LACP can offer benefits to throughput and resiliency, but you should consider the complexity of maintaining LACP environments when making the decision. For more information, see TR-4100: Nondisruptive Operations with SMB File Shares.

Network connection concurrency and TCP slots: NFSv3

When an NFS mount is established from a client to an ONTAP NFS server, a connection ID (CID) is also established. Each incoming NFS operation gets assigned a placeholder resource in ONTAP called an executive context (exec_ctx or exec). As operations complete, the reserved execs are freed to the system for use with a new incoming operation.

For each CID, ONTAP allows 128 execs to be used at any given moment. If a client sends more than 128 operations at a time, then ONTAP will push back on that client until a new resource is freed. These pushbacks are only microseconds per operation in most cases, but over the course of millions of requests across hundreds of clients, the pushback can manifest into performance issues that don't have the usual signatures of performance issues on storage systems, such as protocol, disk, or node latency. As a result, isolating these issues can be difficult and time consuming.

Older Linux kernels (pre-RHEL 6.x days) had a static setting for RPC slot tables of 16. In newer Linux clients, that setting was changed to a maximum of 65,536 and the NFS client would dynamically increase the number of slot tables needed. As a result, newer NFS clients could potentially flood NFS servers with more requests than they can handle at one time.

NFSv4.x operations are sent as compound requests (such as three or four NFS operations per packet) and NFSv4.x sessions slots are used to parallelize requests instead of RPC slot tables. For more information, see "NFSv4.x concurrency: Session slots."

There are a few ways to address performance issue caused by slot tables:

- Use more NFSv3 mount points per client (must be to different locations in the volume to be effective).
- Throttle the number of NFSv3 requests a single client can send per TCP connection/session.
- Use the nconnect mount option to get more TCP connections/sessions per mount, or

- Use Session Trunking to establish more TCP connections

However, before you decide to address RPC slots in your environment, it is important to keep in mind that lowering the RPC slot tables on an NFS client is effectively a form of throttling and can negatively affect performance depending on the workload. An NFS client that needs to send one million NFS requests will send those requests regardless of RPC slot table settings. Setting RPC slot tables is essentially telling the NFS client to limit the number of requests it can send at any given time. The decision of whether to throttle the NFS client or let the storage system enact a form of flow control depends on your workload and use case.

Before adjusting these values, it's important to test and identify if too many slot tables will cause performance issues/impact to applications.

Identifying potential issues with RPC slot tables

As previously mentioned, modern NFSv3 clients use dynamic values for RPC slot tables, which means that the client will send as many concurrent operations on a single TCP session as possible—up to 65,336. However, ONTAP allows only 128 concurrent operations per TCP connection, so if a client sends more than 128, ONTAP will enact a form of flow control on NFSv3 operations to prevent rogue clients from overrunning storage systems by blocking the NFS operation (exec contexts in ONTAP) until resources free up. This flow control can manifest as performance issues that cause extra latency and slower job completion times that might not have a readily apparent reason from the general storage system statistics. These issues can appear to be network related, which can send storage administrators down the wrong troubleshooting path.

To investigate whether RPC slot tables might be involved, use the ONTAP performance counter. You can check whether the number of exec contexts blocked by the connection being overrun is incrementing.

To gather those statistics, run the following `diag privilege` command.

```
::*> statistics start -object cid -instance cid
```

Then, review the statistics over a period of time to see if they are incrementing.

```
::*> statistics show -object cid -instance cid -counter execs_blocked_on_cid
```

On NFS clients, you can leverage the `nfsiostat` command to show active in-flight slot tables.

```
# nfsiostat [/mount/path] [interval seconds]
```

When you set lower RPC slot table values on a client, the RPC slot table queue will shift from the storage to the client, so the `rpc bklog` values will be higher.

With the slot tables set to 128, the `rpc bklog` got as high as 360 when creating 5,000,000 files from two clients and sent around 26,000 ops/s.

```
# nfsiostat /mnt/FGNFS 1 | grep "bklog" -A 1
           ops/s       rpc bklog
       25319.000         354.091
--
           ops/s       rpc bklog
       24945.000         351.105
--
           ops/s       rpc bklog
       26022.000         360.763
```

But ONTAP didn't have to block any of the incoming operations.

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid -sample-id
All_Multi1_bs65536

    Counter                                                     Value
```

```
    ------------------------------- -------------------------------
    execs_blocked_on_cid                                          0

    Counter                                                   Value
    ------------------------------- -------------------------------
    execs_blocked_on_cid                                          0
```

If RPC slot table values are set to higher values, then the RPC queue (`rpc bklog`) will be lower on the client. In this case, the slot tables were left as the default 65,536 value. The client backlog was 0 and the ops/s were higher.

```
# nfsiostat /mnt/FGNFS 1 | grep "bklog" -A 1
           ops/s       rpc bklog
       22308.303           0.000
--
           ops/s       rpc bklog
       30684.000           0.000
```

That means the storage would need absorb more of those RPC calls, as the client isn't holding back as many of those operations. We can see that in the ONTAP statistics.

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid -sample-id
All_Multi1_bs65536

    Counter                                                   Value
    ------------------------------- -------------------------------
    execs_blocked_on_cid                                     145324

    Counter                                                   Value
    ------------------------------- -------------------------------
    execs_blocked_on_cid                                     124982
```

When we exceed a certain number of blocked execs, we'll log an EMS. This is the EMS that was generated:

```
cluster::*> event log show -node tme-a300-efs01-0* -message-name nblade.execsOverLimit
Time                Node             Severity      Event
------------------- ---------------- ------------- ---------------------------
4/8/2021 17:01:30   node1
                                     ERROR         nblade.execsOverLimit: The number of in-flight
requests from client with source IP x.x.x.x to destination LIF x.x.x.x (Vserver 20) is greater
than the maximum number of in-flight requests allowed (128). The client might see degraded
performance due to request throttling.
```

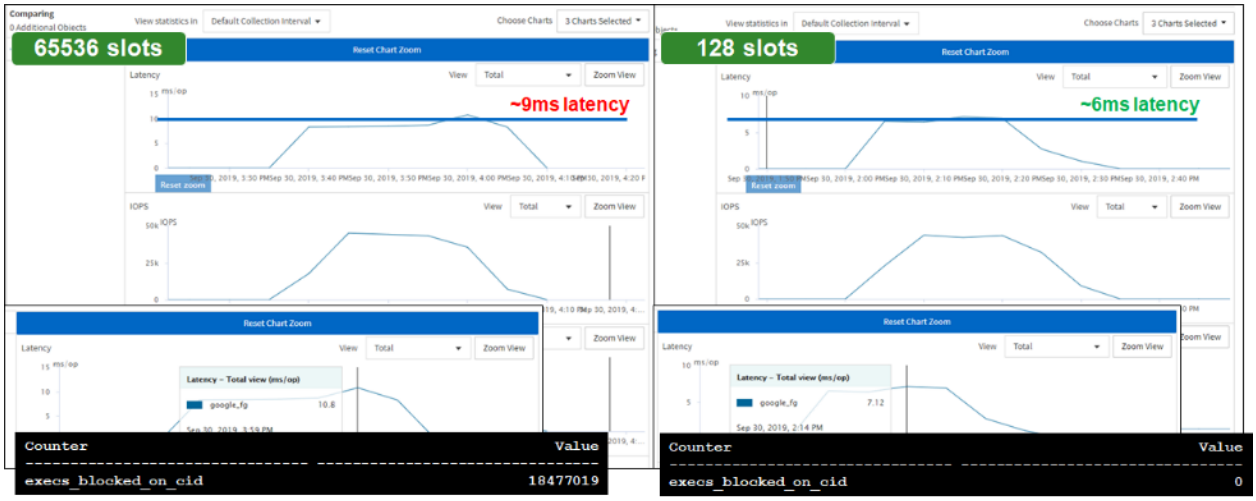
In general, if you aren't seeing `nblade.execOverLimit` EMS events in ONTAP 9.8 and later, RPC slot tables aren't likely causing problems for your workloads. In ONTAP 9.7 and earlier, these events do not exist, so you would want to monitor the CID stats and watch for large increments of `exec_blocked_on_cid`. If you're unsure if your environment is having an issue, contact NetApp support.

Example #1: RPC slot table impact on performance – high file count workload

In the following example, a script was run to create 18 million files across 180,000 subdirectories. This load generation was done from three clients to the same NFS mount. The goal was to generate enough NFS operations with clients that had the default RPC slot table settings to cause ONTAP to enter a flow-control scenario. Then, the same scripts were run again on the same clients - but with the RPC slot tables set to 128.

The result was that the default slot tables (65,536) generated 18 million `execs_blocked_on_cid` events and added 3ms of latency to the workload versus the run with the lower RPC slot table setting (128).

Figure 14) Impact of RPC slot tables on NFSv3 performance.



Although 3ms might not seem like a lot of latency, it can add up over millions of operations, considerably slowing down job completion.

Example #2: RPC slot table impact on performance – sequential I/O workload

In TR-4067, we show a number of tests that illustrate NFSv3 versus NFSv4.1 performance differences, along with different wsize/rsize mount option values. While running these tests, we also saw the negative effects of RPC slot tables increasing the number of execs blocked on CIDs causing performance bottlenecks that added 14.4ms of write latency to some of the performance runs, which in turn added 5.5 minutes to the overall job completion times.

The tests were run across two clients on a 10GB network, using a script that runs multiple dd operations in parallel. Overall, eight folders with two 16GB files each were created and then read and deleted.

- When the RPC slots were set to the maximum dynamic value of 65,536, the dd operations took **20 minutes, 53 seconds**.
- When the RPC slots were lowered to 128, the same script took just **15 minutes, 23 seconds**.

With the 1MB wsize/rsize mount options and 65,536 RPC slots, the `execs_blocked_on_cid` incremented to approximately 1,000 per node.

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid

Scope: node1

    Counter                                                     Value
    -------------------------------- ----------------------------------
    execs_blocked_on_cid                                         1001

Scope: node2

    Counter                                                     Value
    -------------------------------- ----------------------------------
    execs_blocked_on_cid                                         1063
```

Figure 15 shows the side-by-side comparison of latency, IOPS and throughput for the jobs using a 1MB wsize/rsize mount value.

Figure 15) Parallel dd performance: NFSv3 and RPC slot tables; 1MB rsize/wsize.

Figure 16 shows the side-by-side comparison of latency, IOPS and throughput for the jobs using a 256K wsize/rsize mount value.

Figure 16) Parallel dd performance: NFSv3 and RPC slot tables; 256K rsize/wsize.

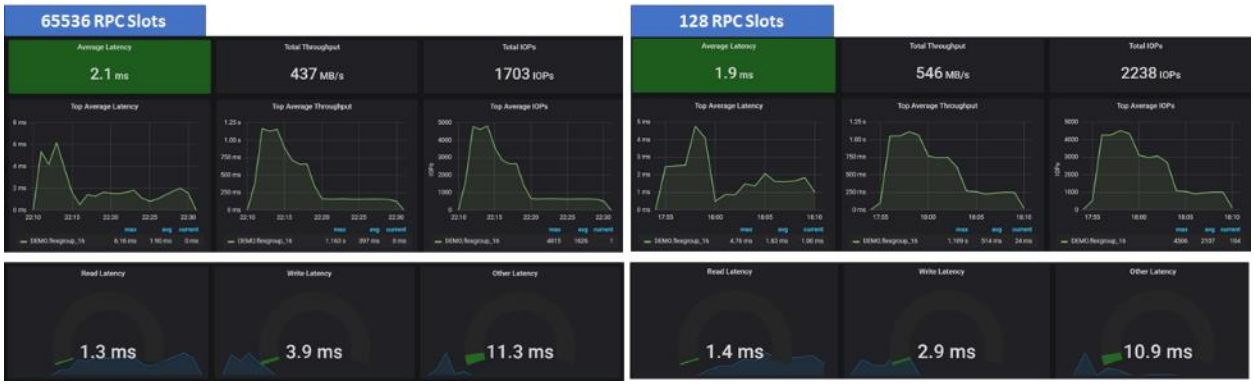


Table 4 shows the comparison of job times and latency with 65,536 and 128 RPC slots.

Table 4) Job comparisons: Parallel dd with 65,536 and 128 RPC slots.

Test	Average read latency (ms)	Average write latency (ms)	Completion time
NFSv3 – 1MB wsize/rsize; 65,536 slot tables	9.2 (+3.2ms)	42.3 (+14.4ms)	20m53s (+5m30s)
NFSv3 – 1MB wsize/rsize; 128 slot tables	6	27.9	15m23s
NFSv3 – 256K wsize/rsize; 65,536 slot tables	1.3 (-.1ms)	3.9 (+1ms)	19m12s (+4m55s)
NFSv3 – 256K wsize/rsize; 128 slot tables	1.4	2.9	14m17s
NFSv3 – 64K wsize/rsize; 65,536 slot tables	.2 (0ms)	3.4 (+1.2ms)	17m2s (+2m14s)
NFSv3 – 64K wsize/rsize; 128 slot tables	.2	2.2	14m48s

Resolving issues with RPC slot tables

In some cases, slot tables on NFS clients don't need to be adjusted at all, as the incrementing counters might not be creating a performance issue. Keep in mind that if a client wants to send one million requests to an ONTAP NFS mount, it sends those requests regardless of what the client is tuned to send. The consideration is, where will the pushback come from: the client or the server?

Setting RPC slot tables to 128 means that the client only sends 128 concurrent requests at a time to the storage system and holds any excess requests in its own queues, which can be viewed with the

`nfsiostat` command (for an example of `nfsiostat`, see “Identifying potential issues with RPC slot tables”).

In some cases, that value might stop the incrementing exec counters in ONTAP (and suppress the EMS messages), but performance might actually suffer, depending on the client’s configuration and available resources.

ONTAP cannot control the number of slot tables that a client sends per TCP session for NFSv3 operations. Therefore, if there is a performance issue caused by the client slot table settings, clients must be configured to limit the maximum slot tables sent through NFS to no more than 128. The configuration of this setting varies depending on the client operating system version, the number of clients, and a remount of the NFS mount on the client required is required for this to take effect. For more information about how to set the values for your client operating system, contact the client vendor.

As previously mentioned, the value to set the RPC slot tables to can vary depending on workload and number of clients. For example, in the tests above in Table 4, 128 RPC slots using a 1MB wsize/rsize achieved a job completion time of 15 minutes, 23 seconds. Setting the RPC slots to 64 lowered that job completion time to 14 minutes, 2 seconds. However, setting the value too low can have the opposite effect, where the clients will be throttled unnecessarily. Test different values in your environment for the best possible results.

Table 5) Job comparisons: Parallel dd with 65,536, 128, and 64 RPC slots.

Test	Average read latency (ms)	Average write latency (ms)	Completion time
NFSv3 – 1MB wsize/rsize; 65,536 slot tables	9.2 (+3.2ms)	42.3 (+19.4ms)	20m53s (+6m51s)
NFSv3 – 1MB wsize/rsize; 128 slot tables	6 (-.3ms)	27.9 (+5ms)	15m23s (+1m21s)
NFSv3 – 1MB wsize/rsize; 64 slot tables	6.3	22.9	14m2s

It is possible to get more performance out of a client’s NFS connectivity by connecting more mount points to different IP addresses in the cluster on the same client, but that approach can create complexity. For example, rather than mounting a volume at `SVM:/volumename`, multiple mount points on the same client across different folders and IP addresses in the volume could be created.

For example:

```
LIF1:/volumename/folder1
LIF2:/volumename/folder2
LIF3:/volumename/folder3
```

Using nconnect to avoid slot table exhaustion issues

Another possible option is to use the nconnect mount option available for some Linux distributions that can perform multiplexing of NFSv3 over the same TCP connection. This option provides more available concurrent sessions and better overall performance. For example, if you use nconnect=8, you get 128 RPC slots * 8 sessions, for concurrency up to 1,024 slots. This also helps reduce the need to adjust client-side slot table settings.

For example, a test that was run using NFSv3 and **no nconnect** applied to the NFS mount with the default 65,536 slot tables on the client showed the following:

- One million 4K files and 2,000 directories created by two clients in **approximately 81 seconds.**
- A total of **835,324** execs blocked across two AFF A300 nodes.

When the same test used **nconnect=2**, these were the results:

- One million 4K files and 2,000 directories created by two clients in **approximately 82 seconds.**
- A total of **827,275** execs blocked across two A300 AFF nodes

The test was then run with **nconnect=8**:

- One million 4K files and 2,000 directories created by two clients in approximately **85 seconds**
- A total of **0** execs blocked across two A300 AFF nodes

The file creation script used can be found here:

https://github.com/whyistheinternetbroken/NetAppFlexGroup/blob/master/file-create-hfc.py

Table 6 shows a side-by-side comparison of this test using different nconnect settings and their results. The table also illustrates the previously mentioned caveats – blocked `exec contexts` in ONTAP do not always equate to worse performance and that nconnect can add performance based on the number of sessions you specify and the workload in use. This set of tests was run against ONTAP 9.9.1, which includes the new exec context throttle feature. Ten runs of each were used and averaged out and the default client settings were used. The mount wsize and rsize used was 64K.

This was the workload makeup:

```
    create_percent                                  33%
    lookup_percent                                  33%
    write_percent                                   33%
```

Table 6) High file count creation (one million files): NFSv3 – with and without nconnect – default slot tables.

Test	Average completion time	Average total execs blocked in ONTAP
NFSv3 – no nconnect	~69.5 seconds	214,770
NFSv3 – nconnect=2	~70.14 seconds	88,038
NFSv3 – nconnect=4	~70.1 seconds	11,658
NFSv3 – nconnect=8	~71.8 seconds	0
NFSv3 – nconnect=16	~71.7 seconds	0

In these tests, we made the following observations:

- Nconnect doesn't help much with this type of workload.
- Exec contexts being blocked can create performance issues (as described in "Example #1: RPC slot table impact on performance – high file count workload"), but don't always create performance issues.

The same tests were run again using the 128 maximum RPC slot table setting and only without nconnect and with nconnect=8. In this case, throttling the RPC slot tables on the clients not only had a slightly positive effect on performance for this workload's average completion time, but it also had more predictability. For instance, when the slot tables were set to 65,536, completion times out of 10 runs had a wide variance - from 55.7 seconds to 81.7 seconds for this workload - whereas 128 slot tables kept performance at a more consistent range of 68–71 seconds. This is because the storage never had to push back on the NFS operations. With more clients added to the mix, the predictability becomes more impactful, as one or two clients can potentially bully other clients into poor performance results – especially in ONTAP releases prior to 9.9.1 (where exec context throttling was added to help mitigate bully workloads).

When nconnect was added to these tests, the performance suffered a bit, as each client wasn't able to push as many NFS operations across the multiple TCP sessions since the client was throttled to 128. If you plan on using nconnect for your workloads, you should consider not setting the RPC slot table values on the clients at all and instead let nconnect spread the workload across TCP sessions.

Table 7) High file count creation (one million files): NFSv3 – with and without nconnect – 128 slot tables.

Test	Average completion time	Average total execs blocked in ONTAP
NFSv3 – no nconnect	~69.4 seconds	0
NFSv3 – nconnect=8	~71.2 seconds	0

When the slot tables were scaled back even further (to 16), the performance suffered significantly, because the client is now sending just 16 requests at a time, so the script takes ~28.1 seconds more to complete without nconnect. With nconnect, we were able to retain about the same average completion time as the other tests with more slot tables (~70.6 seconds), because we get 16 slot tables **per** session (16 * 8 = 128), rather than 16 slot tables on a single session. Because we can send 128 operations per session with nconnect and only 16 per session without, the performance is greatly improved when using nconnect with this configuration. However, in environments with hundreds of clients, setting the slot tables to 16 might be the only way to avoid performance problems caused by slot table overruns.

Table 8) High file count creation (one million files): NFSv3 – with and without nconnect – 16 slot tables.

Test	Average completion time	Average total execs blocked in ONTAP
NFSv3 – no nconnect	~99.3 seconds	0
NFSv3 – nconnect=8	~70.6 seconds	0

These results prove that nconnect provides a way to better distribute the RPC slot requests across TCP connections without costing much in the way of performance and removes the need to adjust RPC slot tables in environments that require high performance and a large number of NFS operations. For more information about nconnect, see the “Nconnect” section.

Note: Because of the variant nature of workloads and effects on performance, you should always test various client settings, mount options, and so on in your environment because there is no “one size fits all” NFS configuration.

Setting RPC slot tables for environments with large client counts

Although the ONTAP session limit for slot tables per TCP connection is 128, there are also node-level limits for exec contexts that can be exceeded.

For example, if a single node can handle up to 3,000 exec contexts, and each TCP session can handle up to 128 exec contexts, then you can have up to 24 concurrent TCP sessions maxing out the RPC slot tables at any given time before the resources are exhausted and ONTAP has to pause to let the resources release back for use (3,000/128 = 23.47). For examples of how to see the available exec contexts per node, see “Exec context throttling.”

This does not mean that ONTAP can only support up to 24 clients per node—ONTAP supports up to 100,000 TCP connections per node (platform dependent). Rather, what this means is that if 24 or more clients are sending the maximum allowed slot entries per connection (128) all at the same time to a single node, then there will be some latency build-up as ONTAP works to free up resources.

Table 9 shows examples of how many clients can send the maximum concurrent operations per connection to a single node.

Table 9) Total clients at maximum concurrent operations (128) before node exec context exhaustion.

Node type	Total available exec contexts per node	Total clients sending maximum operations per connection (128) per node
AFF8040	1,500	1,500/128 = ~11.7
AFF A300	3,000	3,000/128 = ~23.4

AFF A800	10,000	10,000/128 = ~78.1*

Note: *This number varies based on the exec context throttling.

If the RPC slot table maximum entries are higher on the client, then fewer clients will be needed to hit that maximum value. If you have hundreds of clients in an environment all working at the same time to the same nodes in the cluster (such as with EDA workloads), then you should consider setting the RPC slot tables to much lower values based on the number of clients and the number of cluster nodes participating in the workload.

Table 10) Total clients using 16 slot tables before node exec context exhaustion.

Node type	Total available exec contexts per node	Total clients sending max operations per connection (128) per node
AFF8040	1500	1500 / 16 = ~93.7
A300	3000	3000 / 16 = ~187.5
A800	10000	10000 / 16 = ~625*

In addition, the following approaches can help improve overall performance for workloads with many clients:

- Clusters with more nodes and data LIFs on each node
- DNS round robin to spread network connections across more nodes on initial mount
- FlexGroup volumes to leverage more hardware in the cluster
- FlexCache volumes to spread read-heavy workloads across more nodes and mount points

  Consider setting RPC slot table values lower on NFS clients to reduce the number of concurrent operations; values will be platform/ONTAP version-dependent and client count-dependent. For example, see Table 9.
- Nconnect to increase the single client performance
- ONTAP 9.9.1 or later when using platforms with 256GB RAM or greater for benefits of “Exec context throttling” to help mitigate bully workload impact

  Note: The slot table recommendations adjust based on ONTAP hardware, ONTAP version, NFS mount options, and so on. Testing different values in your environment is highly recommended.

Does the RPC slot table limit affect other NAS protocols?

RPC slot table limits affect only NFSv3 traffic:

- SMB clients use different connection methodologies for concurrency, such as SMB multichannel, SMB multiplex, and SMB credits. The SMB connection methodology depends on client/server configuration and protocol version. For example, SMB 1.0 uses SMB multiplex (mpx), whereas SMB2.x uses SMB credits.
- NFSv4.x clients do not use RPC slot tables—instead, they use state IDs and session slots to control the flow of concurrent traffic from clients.

Table 11 shows test run results from NFSv3 and NFSv4.1 using the same 65,536 slot table values.

Table 11) Job comparisons: Parallel dd – NFSv3 and NFSv4.1 with 65536 RPC slots.

Test	Average read latency (ms)	Average write latency (ms)	Completion time
NFSv3 – 1MB wsize/rsize; 65,536 slot tables	9.2 (+2.7ms)	42.3 (+5.5ms)	20m53s (+5m47s)

NFSv4.1 – 1MB wsize/rsize; 65,536 slot tables	6.5	36.8	15m6s
NFSv3 – 256K wsize/rsize; 65,536 slot tables	1.3 (-.1ms)	3.9 (+.7ms)	19m12s (+7m2s)
NFSv4.1 – 256K wsize/rsize; 65,536 slot tables	1.4	3.2	12m10s
NFSv3 – 64K wsize/rsize; 65,536 slot tables	.2 (+.1ms)	3.4 (+2.2ms)	17m2s (+1m54s)
NFSv4.1 – 64K wsize/rsize; 65,536 slot tables	.1	1.2	15m8s

Exec context throttling

When an NFS operation is sent from a client, ONTAP reserves a resource on the node as a placeholder while that operation is processed, called an execution context (exec context). When the operation is finished, the exec context is freed back to the system for use.

These resources are finite within a node and depend on factors such as the ONTAP version, platform/memory, and so on. You can see the per-node limit of these values with the following command:

```
cluster::> set diag
cluster::*> systemshell -node node1 -command "sysctl -a | grep preallocated"
```

Table 12 shows some examples of per-node values for available exec contexts.

Table 12) Exec contexts per node.

Node type	Memory	ONTAP version	Total preallocated exec contexts
AFF8040		9.8	1,500
AFF A300		9.9.1	3,000
AFF A800		9.8	10,000
FAS9000		9.9.1	10,000
AFF A50	256 GB	9.16.1	3,000
AFF A70	256 GB	9.16.1	3,000
AFF A90	2048 GB	9.16.1	10,000
AFF A1K	2048 GB	9.16.1	10,000

In addition to per-node limits, there is also a limit of 128 concurrent operations (assigned exec contexts) per TCP CID. If a client sends more than 128 concurrent operations, ONTAP blocks those operations until a new resource is freed up. By default, Linux clients are configured to send up to 65,536 of these operations concurrently, so the limits can start to be reached relatively quickly.

The implications of this are covered in more detail in “Identifying potential issues with RPC slot tables.”

In some cases, a single client might overwhelm a node’s WAFL layer with requests to the same volume, which then increases latency in WAFL. This increased latency increases the amount of time it takes to free up exec contexts and release them back to the system – thus reducing the total number of exec contexts available to other workloads connecting to the same node. This is commonly seen in grid computing applications, where many clients are sending NFS operations to the same volumes at once. For example, if every NFS client is sending 128 concurrent operations at a time, then it would only take 24 clients to overwhelm the limits.

ONTAP 9.9.1 introduces a new feature for platforms with >256GB of RAM that helps limit the impact of bully workloads that send more the supported concurrent operations have on other workloads in the system by throttling back the number of available exec contexts for all connections. This throttling is based on the total utilization of exec contexts on the node and helps scale back the operations to ensure the node totals don’t get exceeded.

Table 13) Exec context throttle scale.

Node utilization of exec contexts	Scale-back factor	Per-connection exec limit
60%	1	128
70%	8	16
80%	16	8

After a node reaches 70% of the total available exec contexts, each connection will only be able to perform 16 concurrent operations until the node's total utilization drops back to 60%. The exec limit will then increase back to 128.

Some considerations:

- This feature is only available in ONTAP 9.9.1 and later, and only on platforms with more than 256GB of memory (such as the AFF A700, AFF A800, and FAS9000).
- These platforms also increase the total number of available exec contexts per node to 10,000. Because throttling does not occur until over 6,000 execs are allocated (previous total exec limit was 3,000), existing workloads should not notice any negative difference in performance.
- This throttling does not help reduce the number of blocked exec contexts due to per-client overruns. You should still follow the guidance in the "Network connection concurrency and TCP slots: NFSv3" section for tuning Linux clients and/or using nconnect.

How do I know if connections are being throttled?

If you are running ONTAP 9.9.1 on platforms with 256GB or greater memory, then exec throttling is enabled by default. For all other platforms, this feature does not apply.

To view whether exec throttling is occurring on your cluster, start the `exec_ctx` statistics object.

```
cluster::> set diag
cluster::*> statistics start -object exec_ctx
```

Then, view the statistics. The following counters show whether exec throttling is in use:

- **throttle_scale.** The scale-back factor which is currently in effect (1, 8, or 16).
- **throttle_increases.** The number of times the scale-back factor has been increased.
- **throttle_decreases.** The number of time the scale-back-factor has been decreased.
- **throttle_hist.** A histogram of allocations at each scale factor (incrementing counters in 1, 8, or 16).

Virtual memory tuning for NFS clients

Another way to tune NFS clients to improve performance is to modify the virtual memory settings and the dirty buffer values (`vm.dirty`). The virtual memory cache is important to how a client performs file caching—and modifying the default values can control how well reads or writes perform on a client. The client writes file operations directly to RAM before flushing them to disk, which means there are fewer back-and-forth conversations between the NFS client and server for workloads. For more information, see https://www.kernel.org/doc/Documentation/sysctl/vm.txt.

There are three triggers for the file caches being flushed:

- **Time-based.** After a buffer reaches the age defined by these tunable options, it must be marked for cleaning (such as flushing; also known as writing to storage).
- **Memory pressure.** After the allocated memory has been filled, the file cache is flushed.
- **Close.** When a file handle is closed, all dirty buffers are asynchronously flushed to storage.

The default cache settings on clients provide a percentage of RAM dedicated for file caching that is usually adequate for most workloads. The following are the default values on a CentOS/RHEL 8.3 client:

```
vm.dirty_background_bytes = 0 ## modifying this sets vm.dirty_background_ratio to 0
vm.dirty_background_ratio = 10  ## modifying this sets vm.dirty_background_bytes to 0
vm.dirty_bytes = 0 ## modifying this sets vm.dirty_ratio to 0
vm.dirty_ratio = 30 ## modifying this sets vm.dirty_bytes to 0
vm.dirty_expire_centisecs = 3000
vm.dirty_writeback_centisecs = 500
vm.dirtytime_expire_seconds = 43200
```

The following sections describe the configuration options for `vm.dirty` and what they do. These values can be set without having to remount or reboot the client to have them take effect.

vm.dirty_ratio | vm.dirty_ bytes

This tunable option defines the amount of RAM made available for the data modified but not yet written to stable storage. Whichever tunable is set, the other tunable is automatically set to zero. Red Hat advises against manually setting either of the two tunables to zero. Setting the `vm.dirty_bytes` to a static value provides more consistent performance across clients regardless of the amount of RAM present, but that can artificially limit performance on systems with large amounts of available RAM for workloads.

vm.dirty_background_ratio | vm.dirty_background_bytes

This tunable defines the starting point at which the Linux write-back mechanism begins flushing dirty blocks to stable storage.

vm.dirty_expire_centisecs

This tunable defines how old a dirty buffer can be before it must be tagged for asynchronously writing out. Some workloads might not close file handles immediately after writing data. Without a file close, there is no flush until either memory pressure or 30 seconds passes (by default). Waiting for these values might prove suboptimal for application performance, so reducing the wait time can help performance in some use cases.

vm.dirty_writeback_centisecs

The kernel flusher thread is responsible for asynchronously flushing dirty buffers between each flush thread sleep. This tunable defines the amount of time spent sleeping between flushes in centiseconds. Lowering this value in conjunction with `vm.dirty_expire_centisecs` can also improve performance for some workloads.

Other effects of untuned file system caches

Because the default virtual memory tunables in modern systems might not do justice to the amount of available RAM, write back can potentially slow down other storage operations that have file system caches that have not been tuned to use more available virtual memory.

Some of these potential effects include:

- Slow or hung directory listings (ls)
- Read throughput significantly lower than write throughput
- High latency (seconds or higher) from nfsiostat

These issues would only be seen from the client performing the mixed read/write workloads and would be seen across all mounted NFS volumes from the storage endpoint on that client during the impact period.

Which workloads benefit the most from virtual memory tuning?

Workload types can be highly variant. The type of workload in use and how files are written controls how impactful virtual memory tuning will be.

In Table 14, one million files were created using a Python script with the f.write function. The following changes were made to the virtual memory settings between tests:

- `vm.dirty_ratio` changed from 30% to 40%
- `vm.dirty_background_ratio` from 10% to 20%
- `vm.dirty_expire_centisecs` from 3,000 to 300
- `vm.dirty_writeback_centisecs` from 500 to 100

The results were underwhelming because this type of operation closes the file much faster than the cache can expire, so there is little for the client to cache. However, when the client is configured to keep more file cache in memory, the total execs blocked by ONTAP are reduced, as the client isn't sending many requests to the storage as frequently.

Table 14) One million files using f.write – NFSv3, 65536 slots: VM firty bytes defaults versus tuned.

Test	Average completion time	Average total execs blocked	Time delta
NFSv3 – no nconnect – default vm.dirty settings	Approximately 69.1 seconds	214,770	-
NFSv3 – no nconnect – vm.dirty settings tuned	Approximately 69.5 seconds	144,790	+.4 seconds
NFSv3 – nconnect=8 – default vm.dirty settings	Approximately 71.8 seconds	0	-
NFSv3 – nconnect=8 – vm.dirty settings tuned	Approximately 69.5 seconds	0	-2.3 seconds

When the file size was increased, the overall completion times drastically improved with the virtual memory settings, while nconnect didn't make a lot of difference because our bottleneck was not due to TCP session limitations.

In this test, dd was used to create 50x 500MB (2 clients, 1 file per folder, 25 folders, 25 simultaneous processes per client) files, with a 256K wsize/rsize mount value.

Table 15) 50x 500MB files using dd – NFSv3, 65536 slots: VM dirty bytes defaults versus tuned.

Test	Average completion time	Average total execs blocked in ONTAP	Time delta (vm.dirty defaults vs. vm.dirty set)
NFSv3 – no nconnect – default vm.dirty settings	134.4 seconds	0	-
NFSv3 – no nconnect – vm.dirty settings tuned	Approximately 112.3 seconds	0	-22.1 seconds
NFSv3 – nconnect=8 – default vm.dirty settings	Approximately 132.8 seconds	0	-
NFSv3 – nconnect=8 – vm.dirty settings tuned	Approximately 112.9 seconds	0	-19.9 seconds

As seen in the high file count/small files example, these settings do not make a substantial difference for every workload, so it's important to test different values until you find the correct combination.

After you find the right values, you can use `/etc/sysctl.conf` to retain the values on reboot.

NFSv4.x concurrency: Session slots

NFSv3 can have performance limitations with TCP slot tables, but NFSv4.x has its own limits when dealing with connection concurrency. TCP slot table settings on NFS clients do not apply to NFSv4.x operations. Instead, NFSv4.x uses session slots for concurrency.

NFSv4.x session slots operate in a similar fashion, in that the number of requests sent by a client over a single TCP connection is limited to a set value, as specified by this **advanced privilege** option.

```
[-v4.x-session-num-slots <integer>] - Number of Slots in the NFSv4.x Session slot tables
(privilege: advanced)
This optional parameter specifies the number of entries in the NFSv4.x session slot table. By
default, the number of slots is 180. The maximum value is 2000.
```

Note: The stated maximum for NFSv4.x session slots in ONTAP is 2,000, but it is not recommended to exceed 1,024 because you might experience NFSv4.x session hangs (as per bug 1392470).

When an NFSv4.x session is set up, the client and server negotiate the maximum requests allowed for the session, with the lower value (client and server settings) being applied. Most Linux clients default to 64 session slots. This value is tunable in Linux clients through the `modprobe` configuration.

```
$ echo options nfs max_session_slots=180 > /etc/modprobe.d/nfsclient.conf
$ reboot
```

You can see the current value for an NFSv4.x mount with the `systool` command (found in the `sysfsutils` package).

```
# systool -v -m nfs | grep session
    max_session_slots   = "64"
```

After modifying the client option, run the following command:

```
# systool -v -m nfs | grep session
    max_session_slots   = "180"
```

Despite the value default of 180 for the NFS server, ONTAP still has a per-connection limit of 128 exec contexts per CID. Therefore, it is still possible for an NFSv4.x client to overrun the available exec contexts on a single TCP connection to ONTAP and enter a pause state while resources are freed to the system if the value is set too high. If session slots are overrun, this also increases the execs_blocked_by_cid counter mentioned in the “Identifying potential issues with RPC slot tables” section.

In most cases, the default 180 value does not encounter this issue, and setting the session slots to a higher value can create conditions where you might run out of system resources. To get more performance out of your clients, consider using the nconnect option for more parallel handling of these operations rather than adjusting the session slot values.

Note: Be sure to use the latest patched release of ONTAP for NFSv4.x to avoid issues such as this: High NFS Latency with multiple connections to node LIFs from common client TCP socket

Can increasing session slots increase the overall performance?

In short, yes—increasing session slots for the client and server can increase overall performance, because you’re providing more concurrency to the workload. However, the performance gains vary based on the workload type, number of slots set, robustness of the clients, as well as how busy the storage system is. It can also affect other workloads on the client, because available resources might be starved out by a single workload. Increasing session slots can potentially overrun the exec contexts per TCP session limits of 128, which can in turn cause blocked exec contexts and potential performance issues in some cases. Testing different session slot values is important to ensure the changes don’t negatively

affect your environment. In general, it is best to leave the NFSv4.x server session slots option to the default value of 180 or lower.

In Table 16, we measured the completion time, average IOPS and throughput for a high file count workload (one million x 4KB files) and a low count sequential write workload (32x 2GB files).

The following parameters were used:

- FlexGroup volume across two AFF A300 nodes
- Two CentOS 8.3 clients
- NFSv4.1 mounts (AUTH_SYS, 256K wsize, no pNFS, nconnect=8)
- Virtual memory tuning as per “Virtual memory tuning for NFS clients”

Table 16) NFSv4.x session slot performance comparison.

Test	Completion time (seconds)	Average IOPS	Average MBps
NFSv4.1 – 1 million x 4KB files (180 session slots)	~253.9	~11688	~15.2
NFSv4.1 – 1 million x 4KB files (256 session slots)	~240.6	~12685	~16.7
NFSv4.1 – 1 million x 4KB files (512 session slots)	~246.5	~12342	~16.1
NFSv4.1 – 1 million x 4KB files (1,024 session slots)	~245.5	~12378	~16.3
NFSv4.1 – 32 x 2GB files (180 session slots)	~148.3	~902	~224.5
NFSv4.1 – 32 x 2GB files (256 session slots)	~149.6	~889	~221.5
NFSv4.1 – 32 x 2GB files (512 session slots)	~148.5	~891	~222
NFSv4.1 – 32 x 2GB files (1,024 session slots)	~148.8	~898	~223.7

Table 17) NFSv4.x session slot performance: Percent change versus 180 slots.

Test	Completion time	IOPS	MBps
NFSv4.1 – 1 million x 4KB files (256 session slots)	-5.2%	+8.5%	+9.8%
NFSv4.1 – 1 million x 4KB files (512 session slots)	-2.9%	+5.6%	+5.9%
NFSv4.1 – 1 million x 4KB files (1,024 session slots)	-3.3%	+5.9%	+7.2%
NFSv4.1 – 32 x 2GB files (256 session slots)	+0.9%	-1.4%	-1.3%
NFSv4.1 – 32 x 2GB files (512 session slots)	+0.1%	-1.2%	-1.1%
NFSv4.1 – 32 x 2GB files (1,024 session slots)	+0.3%	-0.4%	-0.4%

Observations

In the high metadata/high file count creation test, more session slots improved completion time, IOPS, and overall throughput, but there seemed to be an optimum point at 256 session slots against 1,024 session slots, because 1,024 session slots tended to overrun the exec contexts per CID values. For more information, see “Identifying potential issues with RPC slot tables.”

In the sequential write workload, there was a slight performance degradation when using higher session slots. This shows that increasing session slots can help, but not in all workload types.

Border Gateway Protocol: ONTAP 9.5 and later

Starting in ONTAP 9.5, ONTAP supports Border Gateway Protocol (BGP) to provide a more modern networking stack for your storage system. BGP support provides layer 3 (L3) routing, improved load-balancing intelligence, and virtual IPs (VIPs) for more efficient port usage.

FlexGroup volumes do not require configuration changes to use this new networking element.

Volume considerations

When deploying NAS shares with ONTAP, data is contained within a construct known as a "volume." This volume represents a unique filesystem ID for clients, which helps ensure NAS shares maintain uniquity across the SVM. Volumes in ONTAP can be FlexVol or FlexGroup volumes.

Choosing FlexVol or FlexGroup volumes

When designing an EDA storage solution, it is important to consider the type of volume to use. A NetApp FlexGroup volume can provide exceptional performance for high-metadata workloads, such as those seen in EDA environments. But a FlexGroup volume might currently lack the necessary feature support compared to a FlexVol volume. For instance, if cascading SnapMirror or SVM DR is needed, you should choose a FlexVol volume. Table 18 lists the features that might be pertinent to EDA workloads, and notes whether the feature is present in FlexVol volumes and FlexGroup volumes alike. If a feature you are interested in is not listed in the table, review TR-4571 for a complete list of supported features.

Table 18) Feature comparison of FlexVol and FlexGroup volumes.

Feature	FlexVol support?	FlexGroup support?
All storage efficiencies (thin provisioning, inline, and postprocess efficiencies)	Yes	Yes (Aggregate inline deduplication in ONTAP 9.2 and later)
AFF	Yes	Yes
SAN (iSCSI/FCP)	Yes	No
SMB/CIFS	Yes	Yes (no SMB1, some SMB2.x and 3.x limitations; see TR-4571)
NFS	Yes	Yes (NFSv3 in all releases; NFSv4.0/4.1 in ONTAP 9.7; NFS 4.2 basic protocol support in ONTAP 9.8; Labeled NFSv4.2 in ONTAP 9.9.1)
FlexClone	Yes	Yes (ONTAP 9.7 and later)
NetApp Snapshot™	Yes	Yes
SnapMirror	Yes	Yes (Fan-out, cascade, SVM DR available in ONTAP 9.9.1 and later; synchronous not yet available)
SnapVault	Yes	Yes (ONTAP 9.3 and later)
NetApp SnapLock®	Yes	Yes (ONTAP 9.12.1 and later)
NDMP	Yes	Yes (ONTAP 9.7 and later)
NetApp SnapCenter®	Yes	Yes (ONTAP 9.8 and later for VMware datastores only)
NetApp SnapDiff™	Yes	Yes (ONTAP 9.4 and later; SnapDiff 2.0 and later)
QoS (minimum and maximum)	Yes	Yes • QoS maximums – ONTAP 9.3 and later (AFF only) • QoS minimums/adaptive QoS – ONTAP 9.4 and later • Qtree QoS/File-level QoS – ONTAP 9.8 and later
Qtrees	Yes	Yes*(ONTAP 9.3 and later)
Qtree QoS	Yes	Yes (ONTAP 9.8 and later)
Quota reporting	Yes	Yes (qtree reporting in ONTAP 9.3 and later)
Quota enforcement	Yes	Yes (ONTAP 9.5 and later)
Antivirus	Yes	Yes (ONTAP 9.4 and later)

NetApp FPolicy	Yes	Yes (ONTAP 9.4 and later)
Volume move	Yes	Yes (Member volume level)
NetApp Volume Encryption	Yes	Yes (ONTAP 9.2 and later)
Volume autogrow	Yes	Yes (ONTAP 9.3 and later)
Increasing volume size	Yes	Yes
Shrinking FlexGroup volume size	Yes	Yes (Manual shrink supported in ONTAP 9.6 and later; autoshrink supported in ONTAP 9.3)
Logical Space Accounting/Enforcement	Yes	Yes (ONTAP 9.9.1 and later)
MetroCluster	Yes	Yes (ONTAP 9.6 and later)
ONTAP Select	Yes	Yes
Cloud Volumes ONTAP	Yes	Yes (ONTAP 9.5 and later)

Questions to consider when deciding between FlexGroup and FlexVol volumes

You should consider the following questions when deciding between using FlexGroup and FlexVol volumes for your EDA storage solution.

- What are the application's needs?
  - Single namespace?
  - What protocol is used to access the volume?
  - What are the performance needs?
- How much space is needed?
- What is the workload type?
- Which features are absolute requirements, and which features are "nice to have"?

Protecting your namespace

A vsroot volume lives only on a single node in a cluster, even though the SVM is accessible through multiple nodes. Because the vsroot is how NFS clients traverse the namespace, it is vital to NFS operations.

```
cluster::> vol offline -vserver NFS -volume vsroot

Warning: Offlining root volume vsroot of Vserver NFS will make all volumes on that Vserver
inaccessible.
Do you want to continue? {y|n}: y
Volume "NFS:vsroot" is now offline.
```

If the vsroot volume is somehow unavailable, NFS clients will have issues whenever the vsroot volume is needed to traverse the filesystem.

This includes (but might not be limited to) the following behaviors:

- Mount requests will hang
- If "/″ is mounted, traversal from "/" to another volume, running ls, and so on will hang
- Unmount operations might fail because the mount is busy even when the volume is back online
- If a volume is already mounted below "/" (such as /vol1), then reads/writes/listings will still succeed

Load-sharing mirrors (LS mirrors) in ONTAP is a way to leverage ONTAP SnapMirror capability to increase vsroot resiliency.

Note: LS mirrors are supported only with vsroot volumes. To share load across data volumes, consider using NetApp FlexCache volumes instead.

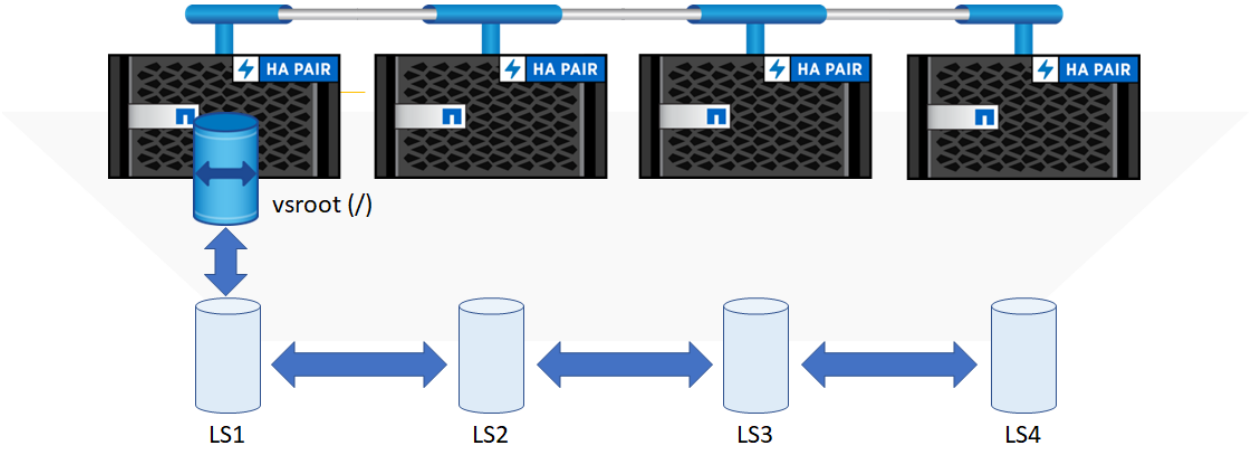
When LS mirrors are available for the vsroot volume, NFSv3 operations can leverage the LS mirror destination volumes to traverse the filesystem. When LS mirrors are in use, it is possible to access the source volume through the `.admin` folder within the NFS mount.

It is highly recommended to create LS mirror relationships for vsroot volumes in NFSv3 environments.

Note: NFSv4.x clients are unable to use LS mirror volumes to traverse file systems due to the nature of the NFSv4.x protocol.

The following figure shows how load-sharing mirrors can provide access to "/" in the event the vsroot is unavailable.

Figure 17) Load sharing mirror protection of vsroot volumes.



Create a load-sharing mirror for the vsroot volume

To create a load sharing mirror for the vsroot volume, do the following:

- Typically, the vsroot volume is 1GB in size. Verify the vsroot volume size before creating new volumes and make sure that the new volumes are all the same size.
- Create a destination volume to mirror the vsroot on each node in the cluster. For example, in a 4-node cluster, create four new volumes with the type, `DP`.
- Create a new SnapMirror relationship from the vsroot source to each new data protection volume that you create. Specify a schedule for updates depending on the change rate of your namespace root. For example, `hourly` if you create new volumes regularly; `daily` if you do not.
- Initialize the SnapMirror relationships by using the `initialize-ls-set` command.

Cluster considerations

An ONTAP cluster that uses only NAS functionality (CIFS/SMB and NFS) can expand up to 24 nodes (12 HA pairs). Each HA pair is a homogenous system (that is, two AFF A1K nodes, two FAS70 nodes, and so on), but the cluster itself can contain mixed system types. For example, a 10-node cluster can have a mix of four AFF nodes and six hybrid nodes for storage tiering functionality.

Cluster considerations: FlexVol volumes

A FlexVol volume is the standard container used in ONTAP to serve data to clients. It spans a single node and aggregate. Metadata operations are performed serially for NAS environments, which means that a single CPU is being used for FlexVol volumes for metadata. In EDA environments, this can affect

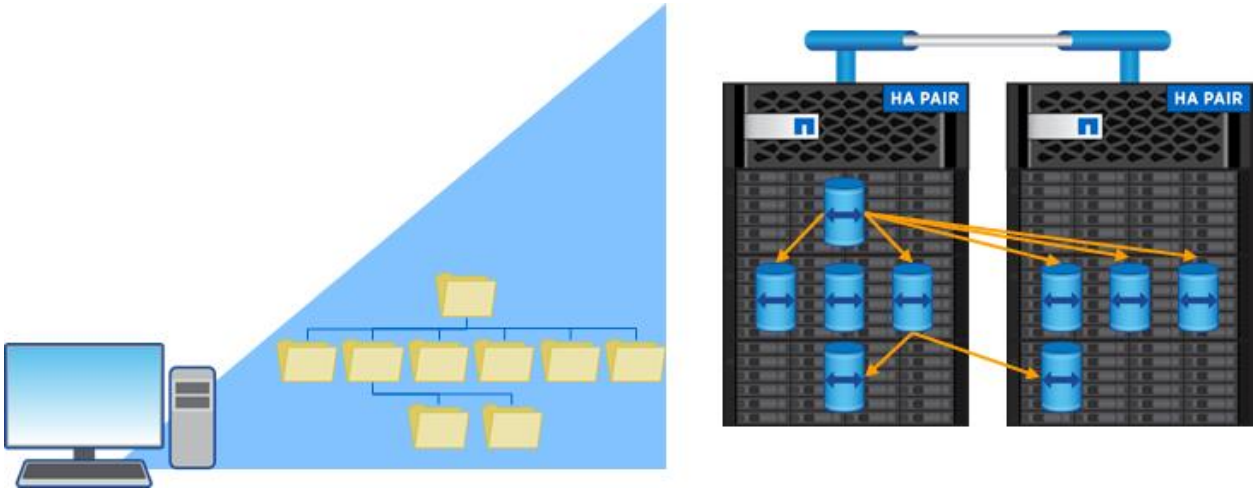
system performance and reduce the amount of hardware in a cluster that can be effectively used for the EDA workload.

To maximize the effectiveness of a FlexVol volume in an EDA workload, use the following tips.

Create multiple FlexVol volumes per node, across multiple nodes

You can design an ONTAP cluster to maximize the hardware available. By creating multiple FlexVol volumes on a node, you can take advantage of more available CPU threads for a workload. Extending those FlexVol volumes across the cluster's nodes gives the workload even more hardware resources to work with. When this is done, the FlexVol volumes appear as folders to the clients that mount the global namespace.

Figure 18) Example of junctioned FlexVol volumes.



If you can design EDA workloads to direct data into individual folders, this can be a viable design to get the most performance out of ONTAP for EDA workloads. If the application requires a namespace that cannot point to specific folders, then use a single FlexVol volume or consider a FlexGroup volume.

When using multiple FlexVol volumes, create them on homogenous hardware

Using the same types of disk (SSD), same size of RAID groups, same number of spindles, same node types, and so on for multiple FlexVol volumes helps ONTAP maintain consistency of performance. Be sure to span nodes and aggregates that are identical to avoid surprises. The exception to this rule is if the EDA workload is designed to leverage some form of project tiering. For example, active workloads can reside on flash storage, while inactive workloads can be tiered off to spinning disk by using nondisruptive volume moves or to disaster recovery sites by using SnapMirror. For more information, see "Project tiering considerations".

Additionally, you can tier Snapshot copies or DR destination data to S3 by using FabricPool. For more information about FabricPool, see TR-4598: FabricPool Best Practices.

Create a local data LIF per node and make sure that clients mount volumes local to the owning node

FlexVol volumes are owned by aggregates, which in turn are owned by nodes in a cluster. Clients can access any volume in a cluster through any of the SVM's data LIFs. If an SVM has a single data LIF, but volumes that live on multiple nodes, then some of the cluster traffic ends up being remote. Although this scenario is usually fine, it can introduce latency into NAS requests. EDA workloads are sensitive to latency, so it is best to avoid remote access to volumes. Having a data LIF per node per SVM, enables

clients to mount to the local path and receive the performance benefits of accessing a volume locally. For more information, see Networking considerations.

With NAS protocols, ONTAP supports features such as CIFS autolocation, NFSv4.x referrals, and pNFS, to help confirm data locality in a clustered file system. For more information on those features, refer to the product documentation for your version of ONTAP.

Enable QoS for performance monitoring and throttling

ONTAP offers the ability to collect statistics through QoS and limit workloads at a volume, qtree, or file level to prevent scenarios where a bully workload can impact other workloads.

Cluster considerations: FlexGroup volumes

A NetApp FlexGroup volume can potentially span an entire 24-node cluster. However, keep the following considerations in mind:

- **NetApp FlexGroup volumes should span only hardware systems that are identical.**

  Because hardware systems can vary greatly in terms of CPU, RAM, and overall performance capabilities, the use of only homogenous systems helps promote predictable performance across the FlexGroup volume. Data is balanced anywhere a FlexGroup volume has member volumes deployed; the storage administrator does not control this placement.
- **NetApp FlexGroup volumes should span only disk types that are identical.**

  Like hardware systems, disk type performance can vary greatly. Because a FlexGroup volume can span multiple nodes in a cluster and the storage administrator has no control over where the data is placed, it is best to make sure that the aggregates that are used are either all SSD, all spinning, or all hybrid. Mixing disk types can lead to unpredictable performance.
- **Disk sizes are not hugely important.**

  Although it is important to use similar disk types on aggregates that a FlexGroup might span, disk sizes are less important. For instance, if your aggregates have 3TB disks but you bought a set of new 16TB disks, feel free to deploy a FlexGroup across them, provided they are the same media type. The main caveat here is that the member volumes that you deploy must be an equivalent size to the others.
- **NetApp FlexGroup volumes can span portions of a cluster.**

  You can configure a FlexGroup volume to span any combination of nodes in the cluster, from a single node, an HA pair, to across all 24 nodes. The FlexGroup volume does not have to be configured to span the entire cluster. However, doing so can take advantage of all the hardware resources that are available.

FlexVol volume layout considerations

When using multiple FlexVol volumes in a cluster for EDA workloads, consider the following recommendations:

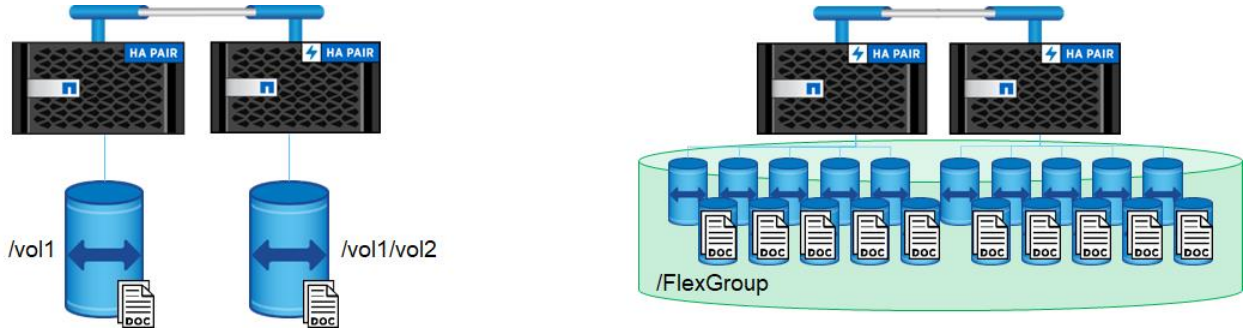
- Balance the FlexVol volumes across nodes evenly. ONTAP offers automatic balanced provisioning to place new volumes on nodes to help balance workloads.
- Create multiple FlexVol volumes per node to take advantage of volume affinities in ONTAP to maximize CPU usage and leverage junction paths to create a workspace for your projects.
- Do no put FlexVol volumes that are participating in the same application workload on different aggregate types/disk types unless you are using volumes to tier inactive data for archiving.
- Place volumes on nodes with local data LIFs and use local data LIFs when mounting from clients.

FlexGroup member volume layout considerations

FlexVol volumes are the building blocks to a FlexGroup volume. Each FlexGroup volume contains several member FlexVol volumes to provide concurrent performance and to expand the capacity of the volume past the usual 300TB limits of single FlexVol volumes.

Standard FlexVol volumes are provisioned from the available storage in an aggregate. FlexVol volumes are flexible and can be increased or decreased dynamically without affecting or disrupting the environment. A FlexVol volume is not tied to any specific set of disks in an aggregate and exists across all the disks in the aggregate. However, individual files themselves are not striped; they are allocated to individual FlexVol member volumes.

Figure 19) FlexVol volume and FlexGroup volume architecture comparison.



Because of this architecture, there are some considerations you should keep in mind when provisioning a FlexGroup volume.

- When you use automated FlexGroup volume creation methods such as `volume create -auto-provision-as flexgroup` (introduced in ONTAP 9.2) or ONTAP System Manager, the default number of member FlexVol volumes in a FlexGroup volume depends on several factors covered in this section.

  Note: For nearly all use cases, NetApp recommends that you let ONTAP determine the member volume count per node—provided you are creating larger FlexGroup volumes (10TB or greater). For smaller FlexGroup volumes, closer attention should be paid to the file sizes in the workload and the percentage of capacity per member volume they would potentially use.

- If a node with spinning disk does not contain two aggregates, the automated FlexGroup creation method might fail in some earlier ONTAP versions. If this happens, continue with manual creation.

FlexVol member volumes are deployed in even capacities, regardless of how the FlexGroup volume was created. For example, if an eight-member, 800TB FlexGroup volume was created, each member is deployed with 100TB. If a larger or smaller quantity of FlexVol member volumes are required at the time of deployment, use the `volume create` command with the `-aggr-list` and `-aggr-list-multiplier` options to customize the number of member volumes deployed per aggregate.

Deployment method #1: Command line

Using the ONTAP command line is the currently recommended way to deploy a FlexGroup volume. However, there are two different ways to do this from the CLI - manually and automatically.

Both commands use the "`volume create`" command set.

Automated FlexGroup creation (auto-provision-as): CLI

This is the preferred method for FlexGroup creation as it combines ease of use with predictable deployment logic and warnings during creation to help prevent misconfigured FlexGroup volumes. To use the automated CLI method, run: "`volume create -auto-provision-as flexgroup`". By default, this command will provision a FlexGroup volume with the following parameters:

N number of member volumes (100GB each; four per aggregate in the cluster – up to two aggregates/eight member volumes)

Total size = 100GB per member times the number of member volumes (16 member volumes = 1.6TB)

All nodes and data aggregates in the cluster used (regardless of node/hardware type)

When you run the command, a warning displays that informs you of the configuration. Be sure to review the warning before you enter `Y`. Are the member volumes the size you want? Are the listed aggregates correct and using the same media types?

In most cases, the default values specified with no options are not adequate. You likely want to specify aggregates or nodes to use in the deployment.

As such, there are some additional configuration option flags for FlexGroup auto provisioning. Below is an example of the command flags for ONTAP 9.16.1.

```
    -aggregate                          -aggr-list
    -aggr-list-multiplier               -optimize-aggr-list
    -auto-provision-as                  -support-tiering
    -use-mirrored-aggregates            -encryption-type
    -nodes                              -size
    -state                              -policy
    -user                               -group
    -security-style                     -unix-permissions
    -junction-path                      -comment
    -max-autosize                       -min-autosize
    -autosize-grow-threshold-percent    -autosize-shrink-threshold-percent
    -autosize-mode                      -space-slo
    -space-guarantee                    -type
    -snapdir-access                     -percent-snapshot-space
    -snapshot-policy                    -language
    -foreground                         -nvfail
    -constituent-role                   -qos-policy-group
    -qos-adaptive-policy-group          -caching-policy
    -snaplock-type                      -vserver-dr-protection
    -encrypt                            -is-space-reporting-logical
    -is-space-enforcement-logical       -tiering-policy
    -tiering-object-tags                -storage-efficiency-mode
    -analytics-state                    -activity-tracking-state
    -key-manager-attribute              -anti-ransomware-state
    -granular-data                      -snapshot-locking-enabled
    -is-large-size-enabled              -in-consistency-group
```

Use these flags to customize your FlexGroup volume's size, tiering policies, space guarantees, nodes, aggregates, and much more.

The following is the general behavior of the automated CLI commands:

- Uses two aggregates per node, if possible. If not, uses one aggregate per node.
- Uses the same number of aggregates on each node.
- Chooses the aggregates that have the most amount free space.
- Creates eight constituents per node if there are eight or fewer nodes.
- In clusters with more than eight nodes, ONTAP scales back to four member volumes per node.
- Uses the fastest aggregates possible. ONTAP first tries SSD, then hybrid, then spinning disk.
- CPU utilization, node performance, aggregate capacity, and so on are not currently considered.

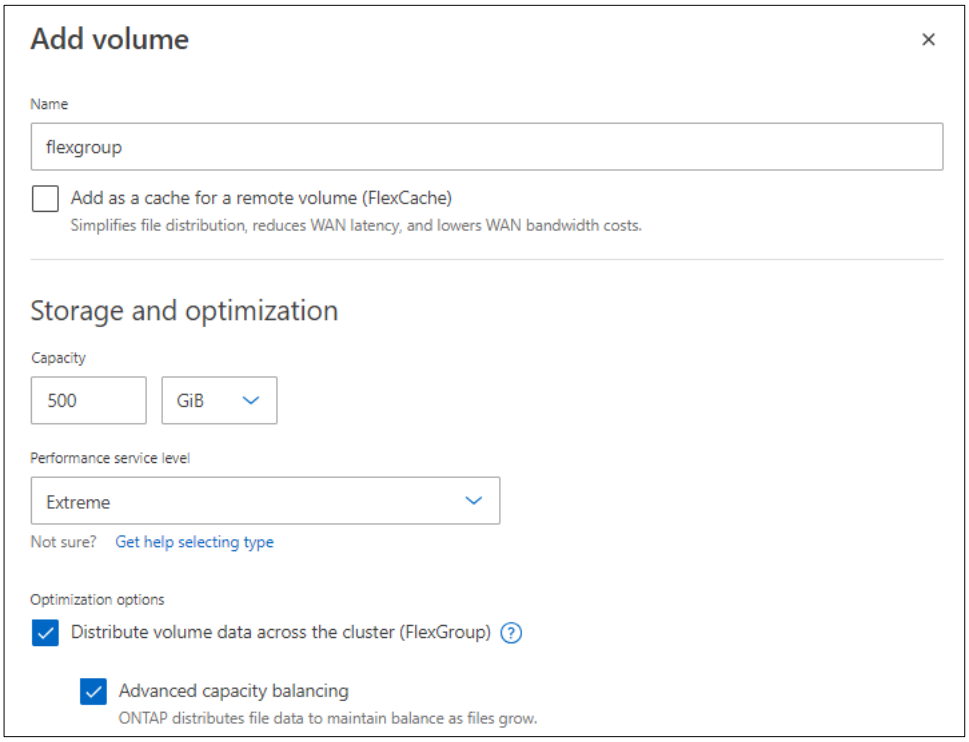
Manual FlexGroup creation: CLI

The CLI also has a more manual approach to creating FlexGroup volumes. In most cases, you should use the automated command, as it will cover most use cases. However, if you need to customize the number of member volumes per aggregate, specify the aggregates to be used—you still use the `volume`

`create` command but instead of using the `-auto-provision-as` option, you must specify `-aggr-list` along with it. Specifying `-aggregate` results in the creation of a normal FlexVol volume (which does not allow you to specify `-aggr-list`). To control the number of member volumes per aggregate, use `-aggr-list-multiplier`. Your member volume count will be the number of aggregates you specify multiplied by the `-aggr-list-multiplier`.

Deployment method #2: ONTAP System Manager

Figure 20) ONTAP System Manager FlexGroup volume creation.



ONTAP System Manager has an easy-to-use GUI for volume creation. However, there are some caveats to consider when deploying with the System Manager GUI that make using the CLI a better choice at this time when provisioning FlexGroup volumes.

To create a FlexGroup volume in ONTAP System Manager, to make sure the volume is a FlexGroup volume and not a FlexVol volume, click More Options and select Distribute Volume Data Across the Cluster.

This tells System Manager to create a FlexGroup volume that spans multiple nodes – no aggregate or node specification is required.

System Manager deploys a FlexGroup volume depending on the following rules.

- Member volumes are never smaller than 100GB
- The smallest allowed FlexGroup is 100GB (one 100GB member volume)
- Smaller FlexGroup volumes deploy fewer member volumes when necessary to adhere to the 100GB rule; for instance, a 200GB FlexGroup volume will deploy two 100GB member volumes.
- Aggregate and node selection for the FlexGroup volume is done automatically; to specify nodes or aggregates, use the CLI or REST API.
- When a FlexGroup volume is large enough to accommodate, then the member volume count is capped to the number of volume affinities available per node.

- System Manager limits your initial FlexGroup volume size to the total space available as if space guarantees were enabled; you can go back into System Manager and grow the volume larger with the Edit functionality
- System Manager uses only similar aggregates for the FlexGroup volume; in other words, it does not mix SSD and HDD aggregates.

Deployment method #3: REST API

Support for REST APIs is available from ONTAP 9.6. With REST APIs, you can create, monitor, and manage FlexGroup volumes. To use REST APIs to provision a FlexGroup volume, use the same guidance as described in the "Deployment method #1: Command line" section. For example, whether to let ONTAP decide the configuration or whether you should manually specify the options.

You can find REST API documentation at `https://[your_cluster_IP_or_name]/docs/api`. This site provides examples and an interactive "Try It Out" feature that enables you to generate your own REST APIs.

For example, to create a FlexGroup volume, you can use the `POST` REST API under `/storage/volumes`. What makes a FlexGroup a FlexGroup volume (and not a FlexVol volume) in this call are one or a combination of the following values:

- **Aggregates.** If you specify more than one, then the REST API creates a FlexGroup volume. This is the same behavior as `-aggr-list` in the CLI.
- **constituents_per_aggregate.** Specifies the number of times to iterate over the aggregates listed with `aggregates.name` or `aggregates.uuid` when a FlexGroup volume is created or expanded. If you create a volume on a single aggregate, the system creates a flexible volume if the `constituents_per_aggregate` field is not specified; it creates a FlexGroup volume if this field is specified. If you create a volume on multiple aggregates, the system always creates a FlexGroup volume. This is the same behavior as `-aggr-list-multiplier` in the CLI.
- **Style.** If you specify `style` as `flexgroup` and do not set the `constituents_per_aggregate` value or more than one aggregate, ONTAP automatically provisions a FlexGroup volume of four members per aggregate. This is the same behavior as `-auto-provision-as` in the CLI.

In the REST API documentation, the "Try It Out" functionality helps guide you as you try to create the correct REST API strings. When you make a mistake, the interface delivers error messages and a list of error codes. Also, a job string URL is given if the REST API command is correct, but the job fails for another reason (such as creating a FlexGroup volume that has members that are too small). You can access the job string through the browser at

```
https://[your_cluster_IP_or_name]/api/cluster/jobs/job_uuid].
```

Volume affinity and CPU saturation

To support concurrent processing, ONTAP assesses its available hardware at startup and divides its aggregates and volumes into separate classes called affinities. In general terms, volumes that belong to one affinity can be serviced in parallel with volumes that are in other affinities. In contrast, two volumes that are in the same affinity often must take turns waiting for scheduling time (serial processing) on the node's CPU.

A node's affinities are viewed with the `advanced privilege nodeshell` command `waffinity_stats -g`.

In ONTAP 9.3 and earlier, a node has up to eight affinities available (four per aggregate).

```
cluster::> set -privilege advanced

cluster::*> node run * waffinity_stats -g
```

```
    Waffinity configured with:
                                      # AGGR affinities :   2
       # AGGR_VBN_RANGE affinities / AGGR_VBN affinity :   4
                        # VOL affinities / AGGR affinity :   4
         # VOL_VBN_RANGE affinities / VOL_VBN affinity :   4
            # STRIPE affinities / STRIPEGROUP affinity :   9
               # STRIPEGROUP affinities / VOL affinity :   1
                     # total AGGR_VBN_RANGE affinities :   8
                                  # total VOL affinities :   8
                       # total VOL_VBN_RANGE affinities :  32
                              # total STRIPE affinities :  72
                                      # total affinities : 149
                                               # threads :  19
```

The sample NetApp FAS8080 EX node above reports that it can support fully concurrent operations on eight separate volumes simultaneously. It also says that to reach that maximum potential, it works best with at least two separate aggregates hosting four constituents each. Therefore, when you are building a new FlexGroup volume that is served by this node, the new FlexGroup volume will include eight constituents on this node, evenly distributed across two local aggregates. Provisioning tools such as ONTAP System Manager attempts to take these affinities into account when creating new FlexGroup volumes, provided the FlexGroup size is adequate to span the available affinities and stay above the minimum 100GB member volume size.

In ONTAP 9.4 and later, the number of available affinities increases to eight per aggregate (two aggregates, 16 per node) for high-end platforms like the AFF A700 and AFF A800:

```
cluster::*> node run * waffinity_stats -g

Waffinity configured with:
                                  # AGGR affinities :   2
   # AGGR_VBN_RANGE affinities / AGGR_VBN affinity :   8
                    # VOL affinities / AGGR affinity :   8
     # VOL_VBN_RANGE affinities / VOL_VBN affinity :   4
        # STRIPE affinities / STRIPEGROUP affinity :   3
           # STRIPEGROUP affinities / VOL affinity :   3
                 # total AGGR_VBN_RANGE affinities :  16
                              # total VOL affinities :  16
                   # total VOL_VBN_RANGE affinities :  64
                          # total STRIPE affinities : 144
                                  # total affinities : 325
                                           # threads :  18
                                            # pinned :   0
                                  # leaf sched pools :  18
                                       # sched pools :  21
```

However, storage administrators usually do not need to worry about volume affinities because ONTAP deploys a FlexGroup volume according to best practices for most use cases. For guidance on when you might need to manually create a FlexGroup volume, see TR-4571.

To simplify the experience, the `vol create -auto-provision-as flexgroup` command (introduced in ONTAP 9.2), the `flexgroup deploy` command, and the ONTAP System Manager GUI handle this setup for the storage administrator.

Initial volume size considerations: FlexVol volumes

When setting an initial FlexVol size, the most important consideration for EDA workloads is the default file count. EDA workloads can contain millions of files, so setting the initial maxfiles to an appropriate value helps avoid future Out of Space warnings when the maxfiles value is exceeded. Table 19 shows a sample of FlexVol sizes, inode defaults, and maximums. If the initial FlexVol maxfiles is not appropriate for the EDA workload, review “Planning for high file counts in ONTAP.” It is also important to keep in mind the maximum volume size and the maximum file count available to an individual FlexVol volume.

Table 19) Inode defaults and maximums according to FlexVol size.

FlexVol size	Default inode count	Maximum inode count
20MB*	566	4,855
1GB*	31,122	249,030
100GB*	3,112,959	24,903,679
1TB	21,251,126	255,013,682
7.8TB	21,251,126	2,040,109,451
100TB	21,251,126	2,040,109,451
300TB	21,251,126	2,040,109,451

Initial volume size considerations: FlexGroup volumes

A common deployment issue is undersizing a FlexGroup volume's member volume capacity. This is often done unbeknownst to the storage administrator because the storage administrator is focused on the total capacity and might not consider underlying member volumes. To them, 80TB should be 80TB. But in a FlexGroup volume, 80TB is actually 80TB divided by the total number of member volumes.

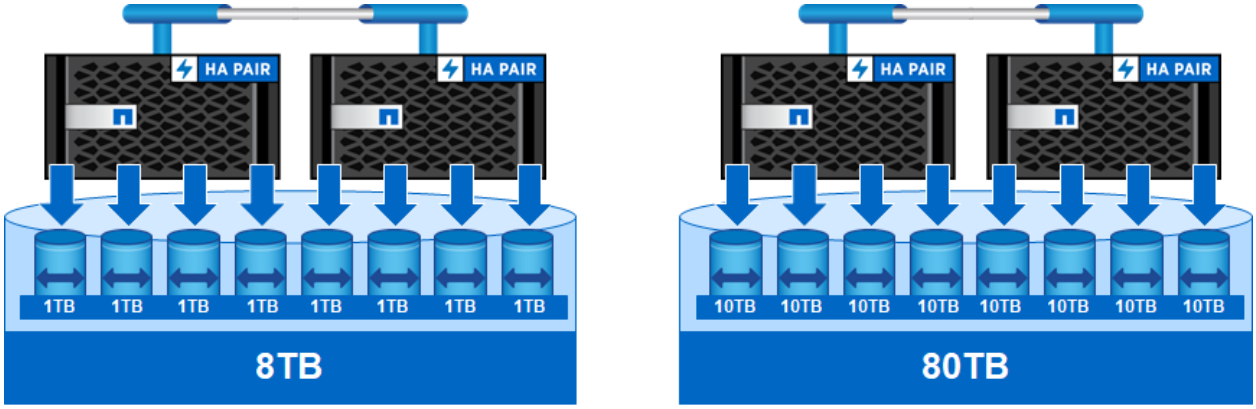
You can create FlexGroup volumes at almost any capacity, but it is important to remember that several FlexVol member volumes make up the total size of the FlexGroup volume. By default, automated FlexGroup commands create a default number of member volumes, depending on the deployment method used. For more information, see the "FlexGroup member volume layout considerations" section.

Best practice 5: Simplifying FlexGroup deployment

If you want to get out of the business of thinking so much about member volumes, upgrade to ONTAP 9.8 or later to get the benefits of proactive resizing, which makes member volume sizes less important to consider.

For example, in an 80TB FlexGroup volume with eight member volumes, each member volume is 10TB in size. These member volume sizes are intended to be inconsequential in most workload cases because 10TB is a large size to work with, but it is important to know what the file sizes of the workload are to help plan the capacity accordingly. For example, if you know your workload has 500GB files, then 10TB member volumes are okay, while 1TB member volumes are problematic.

Figure 21) FlexGroup volumes - member sizes versus FlexGroup volume capacity.



Capacity considerations: FlexGroup volumes

The stated supported limits for a FlexGroup volume are 200 constituent volumes, 20PB, and 400 billion files. However, these are simply the tested limits in a 10-node cluster. When you factor in the maximum volumes that are allowed per node in a cluster, the limits can potentially expand dramatically. In ONTAP 9.12.1P2 and later the maximum supported limits for a FlexGroup volume are 200 constituent volumes, 60PB, and 400 billion files due to the FlexVol size limit being increased to 300TB.

Ultimately, the architectural limitation for a FlexGroup volume is the underlying hardware capacities and the total number of allowed volumes in a single cluster.

Table 20) Theoretical maximums for FlexGroup based on allowed volume count in ONTAP.

Maximum cluster size	Current architectural maximum member volumes per cluster (ONTAP 9.16.1)	Theoretical maximum capacity per FlexGroup volume	Theoretical maximum inodes per FlexGroup volume
24 nodes	12,000 **Note:** SVM root, node root volumes and LS mirror volumes count against this value.	Approximately 3,585PB (based on 300TB per member volume * approximately 11,950 FlexGroup member volumes)	Approximately 23.9 trillion inodes (based on 2 billion inodes * Approximately 11,950 FlexGroup member volumes)

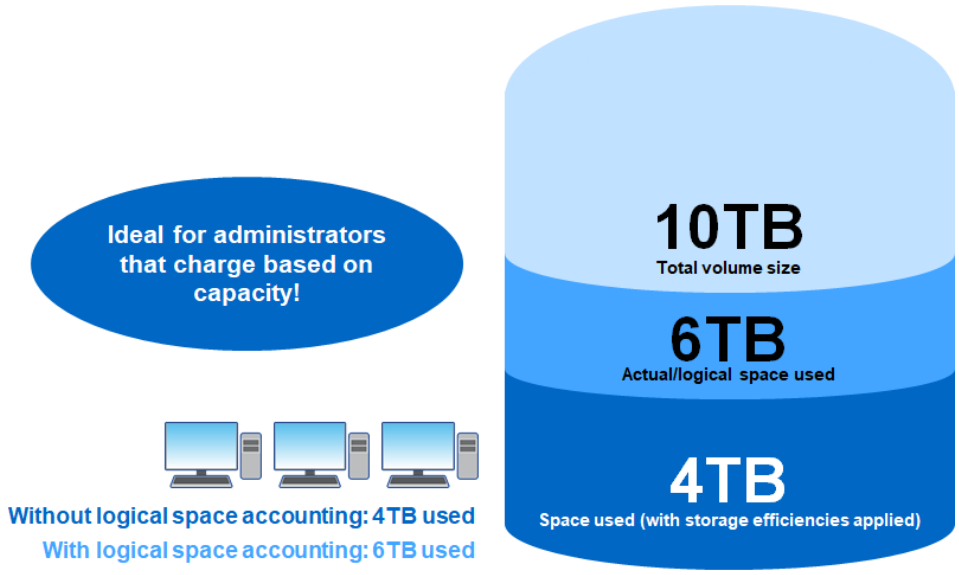
Note: The main limiting factor for the number of 300TB member volumes allowed in a cluster is the underlying physical hardware limitations, which vary depending on the system. If you want to exceed the stated 60PB/400 billion file/200-member volume limits, contact your NetApp sales representative to begin a qualification process.

Logical space accounting

Logical space accounting was introduced for FlexVol volumes in ONTAP 9.4. It enables storage administrators to mask storage efficiency savings so that end users avoid overallocating their designated storage quotas.

For example, if a user writes 6TB to a 10TB volume and storage efficiencies save 2TB, logical space accounting can control whether the user sees 6TB or 4TB.

Figure 22) How logical space accounting works.



ONTAP 9.5 enhanced this feature and added quota enforcement support to give more control to storage administrators by preventing new writes according to the logical space.

ONTAP 9.9.1 and later adds support for logical space reporting and enforcement with FlexGroup volumes.

Storage efficiency considerations

ONTAP provides enterprise-class storage efficiency technologies, including the following:

- Thin provisioning
- Data compaction
- Inline data compression
- Inline deduplication, including aggregate inline deduplication (ONTAP 9.2 and later)
- Temperature Sensitive Storage Efficiency (ONTAP 9.8 and later)

Many EDA workloads are made up mostly of small files. One benefit of storage efficiencies in ONTAP is that files smaller than 128KB are never copied twice due to the architecture of the WAFL file system. This gives ONTAP an innate storage efficiency advantage over some competitors in the EDA space.

Because of the nature of EDA workloads and files, ONTAP can deliver up to 20% efficiency by using deduplication and up to 30% efficiency by using compression, for a total of up to 50% space savings. This means greater return on investment (ROI) on your storage investment for EDA workloads.

Note: Efficiency saving percentages are dataset and workload dependent.

Enabling storage efficiencies can add up to 8% CPU overhead, so the decision to enable efficiencies on primary storage must be considered in terms of total space savings compared to the performance impact. For instance, if you have a dataset that is highly randomized or already compressed, storage efficiencies might not deliver space savings to justify the slight CPU cost. However, NetApp highly recommends enabling storage efficiencies on secondary storage in all cases.

Best practice 6: Storage efficiency in EDA workloads

To get the most out of your ONTAP storage system, NetApp recommends enabling all available storage efficiency options (inline and postprocess, as well as thin provisioning). Enabling these features has a very small impact on system performance, which is outweighed by the cost savings that ONTAP storage efficiencies offer.

Thin provisioning

Thin provisioning allows storage administrators to allocate more space to workloads than is physically available. For instance, if a storage system has 100TB available, it is possible to create four 100TB volumes with space guarantees enabled. The benefit of this approach in EDA workloads is that it frees up available storage and allows the applications to drive the capacity usage rather than the storage. When leveraging complementary features such as NetApp Snapshot™ technology, autodelete, and volume autogrow, and efficiencies such as compaction, deduplication, compression, and so on, thin provisioning EDA workloads can prove beneficial. In addition, nondisruptive volume moves can be incorporated to move volumes automatically (through workflow automation) as capacity is exhausted on a physical node.

Differences between FlexVol and FlexGroup storage efficiency

FlexVol and FlexGroup volumes both leverage the same storage efficiency features in ONTAP, but due to the way these features operate and the architectures of the volume types, there are distinct differences in how efficiencies operate and how much space savings you might see.

Storage efficiency feature comparison: FlexVol and FlexGroup

Table 21 shows a list of storage efficiency features and the volume style support (how it is applied) as of ONTAP 9.9.1 and later. See the product documentation/ONTAP release notes and TR-4571 for information on when each feature was introduced. Most of the storage efficiency features are applied at the volume level for FlexGroup volumes, except for inactive data compression. This must be applied at the member volume level. For more information, see the product documentation for using storage efficiencies in ONTAP.

Table 21) ONTAP storage efficiency support matrix: FlexVol and FlexGroup volumes.

Efficiency feature	FlexVol support?	FlexGroup support?
Inline deduplication*	Y	Y
Inline data compaction	Y	Y
Inline data compression (software) - 8K	Y	Y
Inline data compression (hardware) – 32K A30/A50/A70/A90/A1K	Y	Y
Inline cross-volume deduplication*	Y	Y
Post-process volume deduplication	Y	Y
Background cross-volume deduplication*	Y	Y
Inline adaptive compression	Y	Y
Compression algorithm modification (background only)	Y	Y
Temperature Sensitive Storage Efficiency (TSSE)*	Y	Y
Inactive data compression	Y	Y**
Post-process data compression***	Y	Y

*AFF/FlashPool supported feature only.

**Applied at the member volume level.

***Not supported for AFF.

Storage efficiency operations

ONTAP performs storage efficiency operations in two main ways: inline (such as during the initial data ingest) and post-process (after the data has already been written). Inline efficiencies are designed to be opportunistic - meaning, ONTAP applies efficiencies when it doesn't affect performance. If efficiencies don't get applied inline, then they are applied in the post-process phase.

Inline efficiencies are only supported on AFF platforms, FlashPool aggregates, and systems with the data protection optimization licenses. FAS systems rely on post-process storage efficiencies for space savings.

Storage efficiency domains

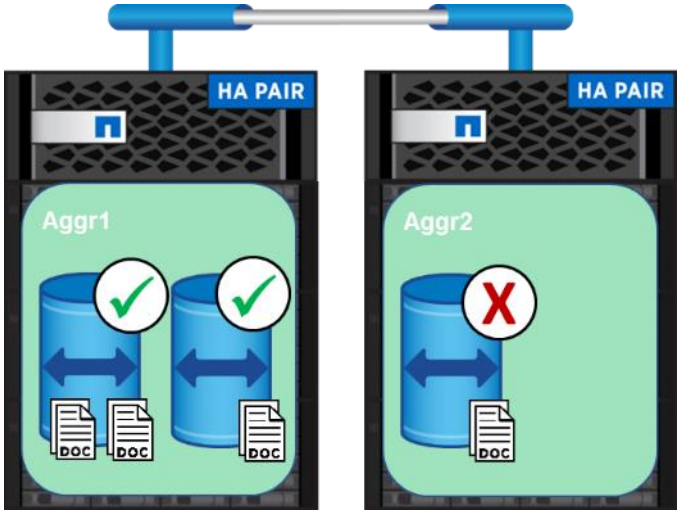
Storage efficiency operates in several different domains in ONTAP:

- File-level (compression – inline and post-process)
- Volume-level (deduplication – inline and post-process)
- Aggregate-level (deduplication – inline and post-process; inline data compaction)

This means that efficiency operations can have different behaviors based on which storage efficiency technologies are in use. For deduplication, ONTAP shows space savings at the volume level and the aggregate level. Therefore, if you have two copies of the same files in the storage system, deduplication reduces the space usage if both files either live in the same volume or the same aggregate (depending on the storage efficiency policy applied to the volume).

If those files reside in different volumes and/or different aggregates (again, depending on the volume efficiency policy in use), then ONTAP is not able to deduplicate the data, because they reside outside of the storage efficiency domains that ONTAP uses.

Figure 23) Storage efficiency domains—when will deduplication be effective?



Inline data compression takes place at the file level; therefore, compression savings should be similar for both FlexVol volumes and FlexGroup volumes because there is no concept of a storage efficiency domain to consider.

Data compaction occurs at the WAFL level in ONTAP, where data that is written to storage that doesn't completely fill a 4K block is compacted into a single 4K block. For instance, if two 2K blocks are written to ONTAP, they combine to a single 4K block. Data compaction savings are seen at the aggregate level.

FlexVol volume efficiency

- A FlexVol volume only resides on a single node and/or aggregate at any given time, so it is always in an eligible storage efficiency domain. Any file that exists within the FlexVol volume has volume-level efficiencies applied at all times.
- If two copies of the same file reside in different FlexVol volumes in the same aggregate, then storage efficiencies are applied at the aggregate level, provided cross-volume deduplication is enabled on the system.
- If two copies of the same file reside in different FlexVol volumes in different aggregates, then storage efficiency savings will not be applied for those files.
- If the FlexVol volumes in use have NVE enabled, then the identical copies of the files are no longer identical, because there is an encryption layer that uniquifies the data, which negates deduplication effects.
- If the FlexVol volumes in use are in an aggregate with NVE, then all volumes in that aggregate share the same encryption keys and storage efficiencies will be effective.

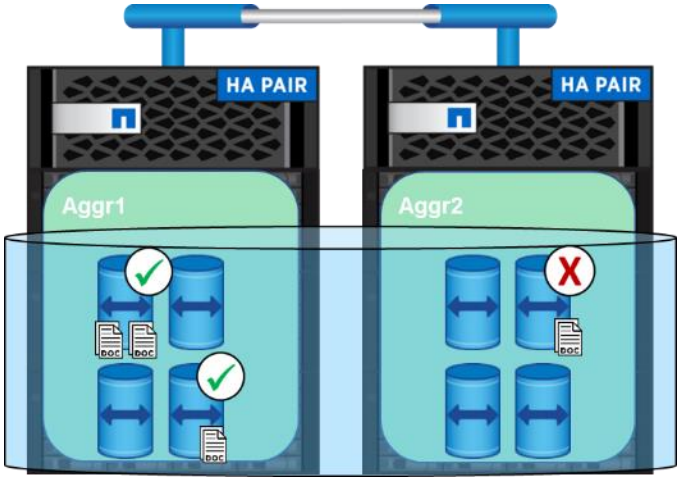
FlexGroup volume efficiency

- A FlexGroup volume can contain one or more FlexVol member volumes and span one or more aggregates in an ONTAP cluster. As a result, the effectiveness of storage efficiencies in a FlexGroup volume depends solely on the layout of the FlexGroup volume. In general, storage efficiency savings in a FlexGroup volume that spans multiple aggregates is less than a FlexVol volume's savings.
- Because data ingest and layout (where the files land when created) is controlled by ONTAP, storage efficiency effectiveness can vary greatly depending on which data lives where in the volume. After

data is placed in a FlexGroup volume, it stays where it lands and does not move retroactively unless a file-based copy is performed back in to the FlexGroup volume.

- If a FlexGroup volume has a single member volume and lives on one aggregate, then you effectively have a FlexVol volume and storage efficiencies are roughly the same as a FlexVol volume, because the data is still considered to be in the volume storage efficiency domain.
- If a FlexGroup volume has more than one member volume and identical copies of a single file reside on the same member volume, then you see similar storage efficiencies as a FlexVol volume, because the files are in the volume storage efficiency domain.
- If a FlexGroup volume has more than one member volume in the same aggregate and identical copies of a single file reside on different member volumes, then you only see storage efficiencies for aggregate-level deduplication and no efficiency savings for the volume storage efficiency domain. Reporting of savings are at the aggregate level rather than at the FlexGroup volume level.
- If a FlexGroup volume has more than one member volume in the different aggregates and identical copies of a single file reside on different member volumes, then storage efficiency effectiveness will depend solely on where those files live in the FlexGroup volume. If there are commonalities in the member volume locations or aggregate locations, you will see some storage efficiency savings. If each identical copy of the file happens to live in a unique member volume in unique aggregates, you won't see any storage efficiency savings for deduplication.
- Inline efficiencies do not gain extra efficiencies for FlexGroup volumes compared to post-process efficiencies, because they must follow the same rules for efficiency domains (volume and aggregate).
- The same volume encryption rules apply for a FlexGroup volume; volume level encryption uniquifies data and negate efficiencies. Aggregate-level encryption maintains a common key and maintains cross-volume deduplication.

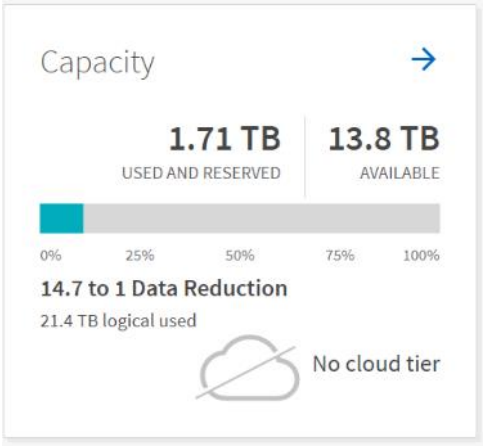
Figure 24) Storage efficiency domains with FlexGroup volumes



Viewing storage efficiency savings

There are several ways to view storage efficiency savings in ONTAP. The easiest, most common method is through ONTAP System Manager. The initial dashboard displays savings and you can click the arrow to navigate to a more granular view of the efficiencies.

Figure 25) Storage efficiency savings – ONTAP System Manager



In addition, you can view storage efficiencies in the ONTAP CLI with the following commands:

- `volume show -vserver SVM -volume volname` (use an asterisk with the volume name in diag privilege to include all FlexGroup member volumes or use `-is-constituent true`; look for `sis-space-saved` and `sis-space-saved-percent` for total efficiency savings)
- `aggregate show -aggregate aggrname1,aggrname2` (shows cross-volume deduplication and data compaction savings for all volumes in the aggregate)
- `aggregate show-efficiency -aggregate aggrname` (summary of efficiencies and ratios for the aggregate)
- `volume efficiency show -volume volname` (shows efficiency progress/status and savings for the volume)
- `aggregate show-space -aggregate-name aggrname -instance` (shows summary of physical and logical used space)

  Note: `sis-space-saved` is an aggregation of all storage efficiency savings (compression + deduplication + data compaction). To see more granular views of how space is being saved, see the `dedupe-space-saved`, `compression-space-saved` and `data-compaction-space-saved` (aggregate only) fields in CLI. Cross-volume deduplication is reported with `dedupe-space-saved` at the aggregate level.

FlexVol and FlexGroup efficiencies compared

In this example, an approximately 36GB, 16,000 file source dataset (in an ONTAP 9.8 system) was copied to a FlexVol volume and a FlexGroup volume (eight member volumes spanning two aggregates on two different nodes in a cluster) on an ONTAP 9.9.1 system to show the difference in effectiveness of storage efficiencies. The dataset is a mix of text files and user data (Word documents, packet trace files, and so on) and has a roughly 2:1 data reduction ratio on the source volume.

Source dataset details: ONTAP 9.8 FlexVol volume

```
cluster::*> vol show -vserver DEMO -volume files -fields used,files-used,dedupe-space-
saved,compression-space-saved,logical-used
vserver volume used    files-used dedupe-space-saved compression-space-saved logical-used
------- ------ ------- ---------- ------------------ ----------------------- ------------
DEMO    files  18.61GB 16152      15.62GB            1.51GB                  35.75GB

cluster::*> vol show -vserver DEMO -volume files -fields sis-space-saved-percent,compression-
space-saved-percent
vserver volume sis-space-saved-percent compression-space-saved-percent
------- ------ ----------------------- -------------------------------
```

```
DEMO    files  48%                    4%
```

Both the FlexGroup and FlexVol volumes have volume efficiencies (inline and post-process) enabled.

Volume configuration details

The storage efficiency comparisons were performed on a few different destination volume types/configurations to illustrate how storage efficiency savings can vary.

The configurations included:

- Standard FlexVol volume
- FlexGroup volume (single member volume; single aggregate)
- FlexGroup volume (four member volumes; single aggregate)
- FlexGroup volume (eight member volumes; two aggregates)

The FlexGroup volume configurations used were to illustrate the differences in efficiencies as compared to a FlexVol volume, but are not always the ideal way to configure FlexGroup volumes. Keep in mind the following cautions/caveats.

FlexGroup volume: Single member volume configuration

A FlexGroup volume with a single member volume is not a common, nor recommended, deployment for FlexGroup volumes, but it is a way to illustrate how storage efficiency savings differ based on the volume layout. In this case, a FlexGroup volume with a single member volume gets most of the benefits of deduplication that you would see with a FlexVol volume, because it's using the volume-level efficiency domain.

FlexGroup volume: Four member volumes, one aggregate configuration

To get better overall storage efficiency savings with a FlexGroup volume, it is possible to host all member volumes in the FlexGroup on the same aggregate. The possible downsides to this approach are:

- Limitation of possible FlexGroup volume size (can only be as large as the aggregate)
- Potential performance limitations (less CPU and disk that can be used for the workload)
- Overutilization of disk/aggregate performance
- Less effective inline efficiencies due to disk backpressure

As a result, one should weigh the drawbacks of hosting the FlexGroup volume on a single aggregate against the benefits of added efficiency savings.

The following example shows the space savings from deduplication on a FlexGroup volume before running any post-process efficiencies—only inline deduplication has been applied here. At the volume level, we see that only 292.8MB has been saved with deduplication. This is because ONTAP only reports savings from the volume-level efficiency domains here. Aggregate-level savings (such as with cross-volume deduplication) would be reported at the aggregate level.

```
cluster::*> vol show -vserver DEMO -volume fgsingleaggr* -fields used,dedupe-space-saved,logical-
used
vserver volume             used    dedupe-space-saved logical-used
------- ------------------ ------- ------------------ ------------
DEMO    fgsingleaggr       37.60GB 292.8MB            37.88GB
DEMO    fgsingleaggr__0001 4.91GB  7.43MB             4.92GB
DEMO    fgsingleaggr__0002 15.73GB 67.15MB            15.80GB
DEMO    fgsingleaggr__0003 16.17GB 212.3MB            16.38GB
DEMO    fgsingleaggr__0004 802.1MB 5.89MB             808.0MB

cluster::*> vol show -vserver DEMO -volume fgsingleaggr* -fields dedupe-space-saved-percent
vserver volume             dedupe-space-saved-percent
------- ------------------ --------------------------
DEMO    fgsingleaggr       1%
```

```
DEMO    fgsingleaggr__0001 0%
DEMO    fgsingleaggr__0002 0%
DEMO    fgsingleaggr__0003 1%
DEMO    fgsingleaggr__0004 1%
```

To see space savings when dealing with multiple FlexGroup member volumes in the same aggregate, look at the aggregate efficiency domain savings. Before data was copied to the FlexGroup volume on the aggr named `tme_a300_efs02_02_SSD_1`, this was what the space savings looked like:

```
cluster::*> aggr show -aggregate tme_a300_efs02_02_SSD_1 -fields size,usedsize,physical-
used,data-compaction-space-saved,sis-space-saved
aggregate               size   usedsize physical-used data-compaction-space-saved sis-space-saved
----------------------- ------ -------- ------------- ---------------------------- ---------------
tme_a300_efs02_02_SSD_1 7.75TB 725.2GB  775.7GB       16.17GB                      16.17GB
```

After the files were copied, you can see that inline cross-volume deduplication savings were approximately 16GB.

```
cluster::*> aggr show -aggregate tme_a300_efs02_02_SSD_1 -fields size,usedsize,physical-
used,data-compaction-space-saved,sis-space-saved
aggregate               size   usedsize physical-used data-compaction-space-saved sis-space-saved
----------------------- ------ -------- ------------- ---------------------------- ---------------
tme_a300_efs02_02_SSD_1 7.75TB 763.0GB  792.5GB       32.26GB                      32.26GB
```

Note: Cross-volume deduplication is not allowed when NVE is in use.

In the above example, we took a volume with approximately 38GB of total space and saved approximately 16.7GB. That's a savings of roughly 44%, which is the savings we saw for the FlexVol volume.

FlexGroup volume: Eight member volumes, two aggregate configuration

This FlexGroup volume was created with the default settings across two aggregates (four member volumes per aggregate, eight member volumes total).

The following example shows the deduplication savings on the default, two-node FlexGroup volume after copying data. Note how much lower the total deduplication savings are as compared to the FlexVol, volume single-member FlexGroup volume, and even the single aggregate FlexGroup volume (see Table 22), because we are spanning both volume-level and aggregate-level deduplication domains.

```
cluster::*> vol show -vserver DEMO -volume fgfiles* -fields used,dedupe-space-saved,logical-used
vserver volume        used    dedupe-space-saved logical-used
------- ------------- ------- ------------------ ------------
DEMO    fgfiles       38.09GB 1.80GB             39.89GB
DEMO    fgfiles__0001 8.88GB  275.1MB            9.15GB
DEMO    fgfiles__0002 4.08GB  195.9MB            4.27GB
DEMO    fgfiles__0003 4.09GB  187.8MB            4.27GB
DEMO    fgfiles__0004 4.85GB  290.2MB            5.13GB
DEMO    fgfiles__0005 4.03GB  243.0MB            4.27GB
DEMO    fgfiles__0006 4.07GB  204.3MB            4.27GB
DEMO    fgfiles__0007 4.03GB  238.0MB            4.27GB
DEMO    fgfiles__0008 4.06GB  206.2MB            4.26GB

cluster::*> vol show -vserver DEMO -volume fgfiles* -fields dedupe-space-saved-percent
vserver volume        dedupe-space-saved-percent
------- ------------- --------------------------
DEMO    fgfiles       5%
DEMO    fgfiles__0001 3%
DEMO    fgfiles__0002 4%
DEMO    fgfiles__0003 4%
DEMO    fgfiles__0004 6%
DEMO    fgfiles__0005 6%
DEMO    fgfiles__0006 5%
DEMO    fgfiles__0007 5%
DEMO    fgfiles__0008 5%
```

Cross-volume deduplication helps somewhat, but you're still limited to the aggregate deduplication domain. Here are the aggregate space usage and deduplication savings for each aggregate that this FlexGroup volume spans, before the data copy:

```
cluster::*> aggr show -aggregate tme_a300_efs02* -fields size,usedsize,sis-space-saved
aggregate               size   usedsize sis-space-saved
----------------------- ------ -------- ----------------------------
tme_a300_efs02_01_SSD_1 7.75TB 1.01TB   10.49GB
tme_a300_efs02_02_SSD_1 7.75TB 726.8GB  10.58GB
```

And here's the comparison after the data copy. Note that we saved approximately 2GB per aggregate (4GB total), a long way from the efficiency savings we saw in the FlexGroup volume spanning the single aggregate, but better than simply relying on volume-level deduplication.

```
cluster::*> aggr show -aggregate tme_a300_efs02* -fields size,usedsize,sis-space-saved
aggregate               size   usedsize sis-space-saved
----------------------- ------ -------- ---------------
tme_a300_efs02_01_SSD_1 7.75TB 1.03TB   12.36GB
tme_a300_efs02_02_SSD_1 7.75TB 743.6GB  12.59GB
```

Note: `sis-space-saved` doesn't just show deduplication savings, but instead shows all combined savings on the aggregate, so some of the savings above may be due to auto adaptive data compression.

FlexGroup volume (two aggregates, default member volume count): Compression savings only

On a FlexGroup volume (across two aggregates) using adaptive compression, this was the result of the inline compression savings.

```
cluster::*> vol show -vserver DEMO -volume fgfiles* -fields used,compression-space-saved
vserver volume        used    compression-space-saved
------- ------------- ------- ----------------------
DEMO    fgfiles       36.91GB 1.53GB
DEMO    fgfiles__0001 6.78GB  1.53GB
DEMO    fgfiles__0002 4.18GB  344KB
DEMO    fgfiles__0003 4.56GB  0B
DEMO    fgfiles__0004 4.20GB  0B
DEMO    fgfiles__0005 4.69GB  304KB
DEMO    fgfiles__0006 4.17GB  0B
DEMO    fgfiles__0007 4.18GB  0B
DEMO    fgfiles__0008 4.15GB  0B
```

Note: Most of the compression savings took place in a single member volume; this is because the file (or files) that were most compressible were located there. In this case, a 4GB packet trace file was the compressed file. No other files in that volume exceeded a few megabytes in size.

The key takeaway here is that, unlike deduplication, compression is unaffected by the layout of a FlexGroup volume and instead is associated with how compressible the individual files in the volume are. You can expect roughly the same compression rates on a FlexGroup volume that you would see with a FlexVol volume.

Storage efficiency comparison: Deduplication

Table 22 offers a side-by-side comparison of deduplication savings on the different volume configurations.

Table 22) Storage efficiency comparisons: Deduplication.

Volume configuration	Total dedupe space saved*	Dedupe percent saved*
FlexVol	15.63GB	44%
FlexGroup; single member volume	15.62GB	44%

FlexGroup; 4 member volumes, single aggregate**	16.17GB	44%
FlexGroup; 8 member volumes, two aggregates, two nodes**	1.8GB	5%

*After inline and post-process efficiencies applied.

**Aggregate level savings (includes compression and compaction).

Note: For a complete summary of storage efficiency savings (compaction, compression and deduplication), view storage efficiencies at the aggregate level.

FlexVol and FlexGroup storage efficiencies: Conclusion

Although you can apply storage efficiency policies at the volume level, the space savings you see in the FlexVol and FlexGroup volumes mostly depend on the FlexGroup volume layout, data types, data placement, and other factors listed above. As a result, in nearly all cases, a FlexGroup volume that spans multiple aggregates in a cluster show less space savings than a single FlexVol volume due to how deduplication works in ONTAP. Additionally, some of the space savings are reported at the aggregate level, so if space savings don't seem to align between FlexVol and FlexGroup volumes, be sure to compare the aggregate level efficiencies as well.

In summary, for FlexGroup volumes:

- In general, deduplication savings are effective on FlexGroup volumes compared to FlexVol volumes.
- Volume-level deduplication only achieves savings when identical blocks are in the same member volume (which is controlled by ONTAP data placement).
- Aggregate-level deduplication achieves savings when identical blocks are in the same aggregate, which bodes well for FlexGroup volumes until member volume span different aggregates. These savings are seen with aggregate storage efficiency commands, rather than volume-level commands.
- Inline data compaction and data compression savings are nearly identical for FlexGroup and FlexVol volumes, but you need to review both the volume and aggregate savings to see the space reduction.

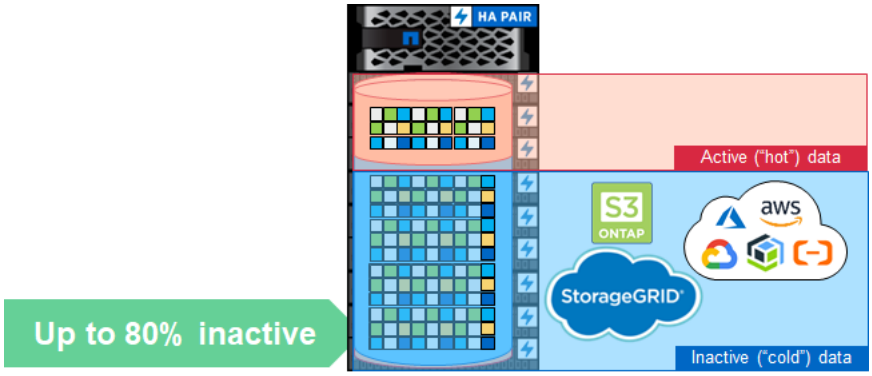
FabricPool

In some cases, inactive data might use as much as 80% of your available capacity on primary storage systems. NetApp FabricPool provides a way to automatically tier that inactive data from a primary ONTAP system to an S3 storage bucket and then automatically retrieve that data when accessed by clients. The goal is to keep cold data from using valuable primary storage capacity without the need to manually move data in and out of an S3 tier.

FabricPool requires a capacity-based BlueXP tiering license when attaching third-party object storage providers (such as Amazon S3) as cloud tiers for AFF and FAS systems.

A Cloud Tiering license is not required when using StorageGRID or ONTAP S3 as the cloud tier or when using Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage as the cloud tier for Cloud Volumes ONTAP.

Figure 26) FabricPool tiering.

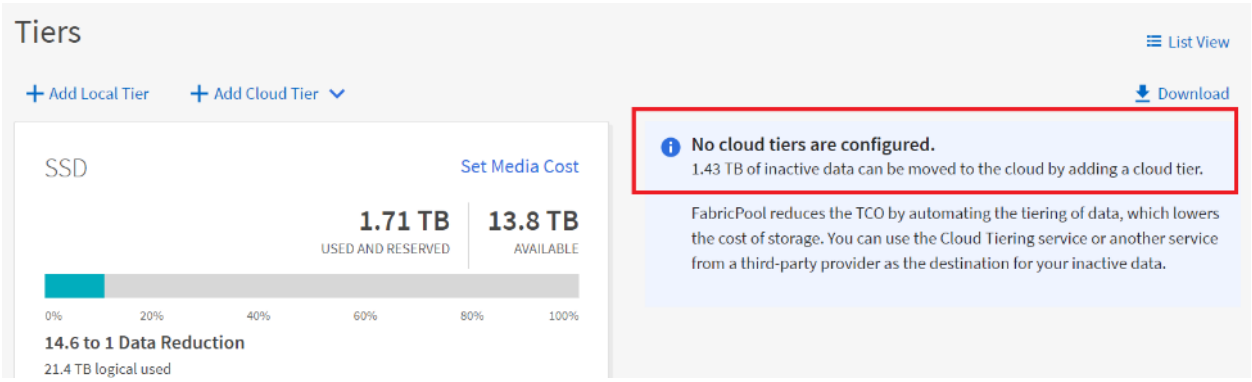


FabricPool reduces storage footprints and associated costs. Active data remains on high-performance local tiers, and inactive data is tiered to low-cost object storage while preserving ONTAP functionality and data efficiencies.

FabricPool is not a backup solution. For complete data protection, consider using existing ONTAP technologies such as SnapMirror.

It's possible to find how much space you could save with FabricPool without ever needing to set it up through inactive data reporting, which is available to enable through the CLI by using the storage `aggregate modify -aggregate -is-inactive-data-reporting-enabled true` and available for viewing at the aggregate level (`storage aggregate show-space -fields performance-tier-inactive-user-data, performance-tierinactive-user-data-percent`) and volume level (`volume show -fields performance-tier-inactive-user-data, performance-tier-inactive-user-data-percent`). Alternately, you can view inactive data in ONTAP System Manager even if it's not enabled via the Tiers view.

Figure 27) Inactive data reporting – ONTAP System Manager.



For more information about FabricPool, see FabricPool on the docs.netapp.com site.

FlexClone volumes

FlexClone volumes are Snapshot-backed copies of active FlexVol volumes that take up nominal amounts of space while enabling administrators to provide proven productivity and efficiency to their end users.

Some use cases for EDA workloads include:

- Better developer productivity through quick workspace creations and faster builds
- Improved performance by offloading code checkouts and providing faster deletes

- Reduced license and storage costs due to efficiencies
- Better DevOps lifecycle management with Snapshot copies for work in progress and tiering of workloads

FlexClone volumes, although not necessarily a best practice for EDA, offer value in EDA workflows that make them worth considering.

FlexClone with FlexGroup volumes

Starting in ONTAP 9.7, NetApp FlexClone is supported for use with FlexGroup volumes. This feature provides storage administrators with a way to create instant, space-efficient copies (backed by Snapshot technology) of volumes to use for testing, development, backup verification, and a variety of other use cases. There are no specific considerations for use with FlexGroup volumes, except that a FlexClone copy of a FlexGroup volume uses the same number of member volumes as the FlexGroup parent volume. As a result, the volume count on a node can start to add up as FlexClone copies are created.

For example, if you have a FlexGroup volume that contains 16 member volumes and then create a FlexClone copy of that FlexGroup volume, you now have used 32 volumes in the system. Each new clone of the volume uses 16 member FlexVol volumes as well.

Backup considerations

EDA workloads are CPU intensive. Therefore, it is a best practice to avoid running backups (either NDMP/tape or CIFS/NFS) on the primary storage. Instead, use SnapMirror to replicate the source volumes to a destination cluster and run the backups from the secondary storage system. Because EDA workloads are generally very high file count workloads, backups that walk the file system (such as NDMP or NAS-based backups) take a long time. In those cases, backup providers that use SnapDiff technologies or the SnapVault feature provide a better experience.

For backup considerations specific to FlexGroup volumes, see TR-4678: Data Protection and Backup with NetApp FlexGroup volumes.

Proactive resizing

ONTAP 9.8 introduces a new feature for capacity management, with the goal of taking capacity management tasks out of the hands of storage administrators and, instead, letting ONTAP manage FlexGroup member volume capacity.

The following should be considered regarding proactive resizing:

- Member volumes remain the same size if the member volume capacity is less than 60%, even if there is a large capacity disparity.
- Proactive resizing starts to adjust member volume sizes between 60-80% used capacity in small increments to attempt to maintain a relatively even balance of available space percentage.
- After 80% used capacity, the goal is to maintain even capacity usage by adjusting the total member volume sizes up or down.
- When a resize occurs, it is not massive; the range is between 10M and 10GB. But it also does not impact performance the way elastic sizing does, as there is no pause needed to check for free space — instead, resizing occurs well ahead of capacity issues.
- Volume autosize is implemented if a member volume reaches the autogrow threshold, provided you have enabled volume autosize.
- If you have very large member volumes (80–100TB), upgrade to ONTAP 9.8P6, ONTAP 9.9.1P1 or later to ensure the 100TB size limits are factored in to the ingest calculations. Otherwise, 100TB member volumes will fill at the same rate as other member volumes until they reach 99% and the member volume has no way to grow larger. For more information, see bug 1391793.
- Member volumes can be as large as 300TB in ONTAP 9.12.1

This free space buffer helps maintain even-file ingest across the volume, reduces capacity imbalance impact, and improves the capacity management story for FlexGroup volumes in ONTAP.

Proactive resizing behavior: Volume autosize disabled

The following example uses a 400GB FlexGroup volume with four 100GB member volumes. A client creates 32-10GB files in the FlexGroup volume across four folders. The FlexGroup volume has volume autosize **disabled**. This means that the FlexGroup volume size specified remains that size, even if it reaches 100% capacity.

At the start of the job (after the first files are written), the capacity balance appears as shown in the following figure:

Figure 28) Initial FlexGroup data balance—proactive resize, autosize disabled.

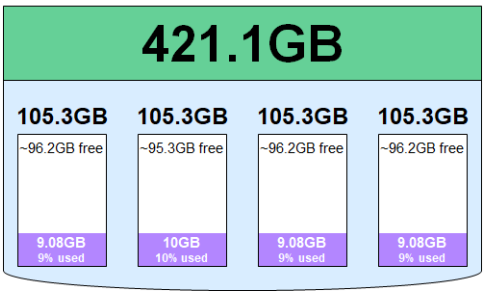
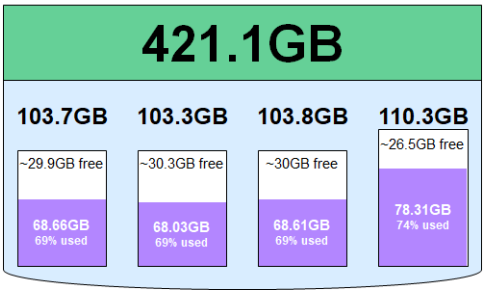
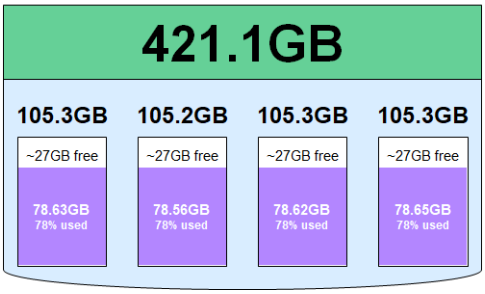


Figure 29) FlexGroup data balance, ~68% used—proactive resize, autosize disabled.



At around 70% capacity usage, you can start to see the member volumes resize a bit to maintain a balanced free space, but the total capacity remains the same.

Figure 30) FlexGroup data balance, job complete—proactive resize, autosize disabled.



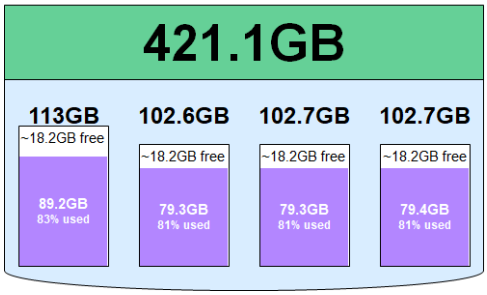
After the job finishes, ONTAP sees that the used space is even across all member volumes and proactive resizing shrinks the member volumes back down to their original sizes and makes them all the same because there is sufficient free space. The total FlexGroup volume size has not changed.

So, what happens when a new 10GB file is written after this?

When a file is written, it ends up in one of the member volumes. That creates a data imbalance, but ONTAP reacts accordingly by resizing the other member volumes to maintain an even amount of free space.

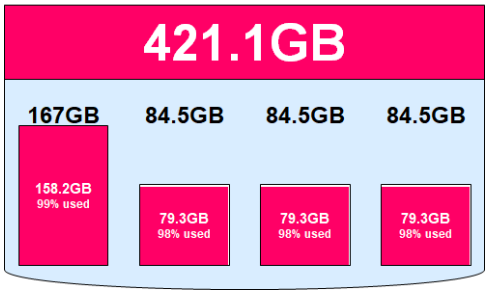
Figure 31 shows the data balance after the new 10GB file is written:

Figure 31) FlexGroup data balance, new large file—proactive resize, autosize disabled.



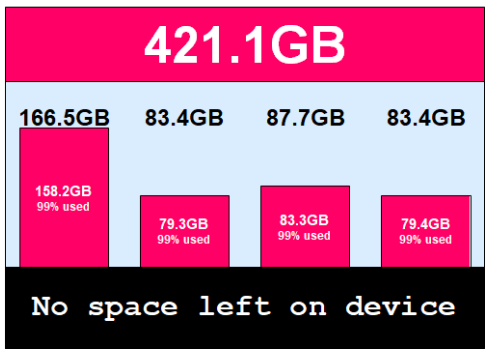
As you can see, the new file ends up in the first member volume. Elastic sizing grows that member volume to 113GB and shrinks the other member volumes while maintaining roughly the same amount of free space available and the total FlexGroup size.

Figure 32) FlexGroup data balance, 80GB file—proactive resize, autosize disabled.



Next, we write a new file to the FlexGroup again. This time, the file is too big to fit into a single member volume (80GB) and is almost too large to fit into the FlexGroup itself. When this happens, ONTAP uses proactive resizing, but every member volume does not have just 1GB of free space remaining. This means that the next 10GB file fails, because the entire FlexGroup is out of space and autosize is disabled, as shown in Figure 33.

Figure 33) FlexGroup data balance, out of space—proactive resize, autosize disabled.



As a result, the next file creation fails, but proactive resizing becomes much more aggressive in adding free space to the member volume to avoid an out of space error. But when a FlexGroup volume itself is out of space, the only remediation is growing the FlexGroup volume manually—or enabling volume autosize.

Proactive resizing behavior: Volume autosize enabled

The following example uses a 400GB FlexGroup volume with four 100GB member volumes. A client creates 32–10GB files in the FlexGroup volume across four folders. The FlexGroup volume has volume autosize enabled with the default settings, which means the following:

- The FlexGroup volume maintains the same capacity, even if proactive resizing occurs, until the 92% used-space threshold is reached.
- After the used-space threshold is reached, the volume increases no more than 20%, as per the default settings. In this case, 566.7GB is the maximum size the volume will grow, which is greater than 20% because this volume's size was increased and then later decreased.
- If the used capacity falls below 50%, then the volume shrinks back to the original size of 421.1GB.

These are the autosize settings for the FlexGroup volume:

```
cluster::> vol autosize -vserver DEMO -volume FG_SM_400G
Volume autosize is currently ON for volume "DEMO:FG_SM_400G".
The volume is set to grow to a maximum of 566.7g when the volume-used space is above 92%.
The volume is set to shrink to a minimum of 421.1g when the volume-used space falls below 50%.
Volume autosize for volume 'DEMO:FG_SM_400G' is currently in mode grow_shrink.
```

When a FlexVol or FlexGroup volume is smaller, the default grow threshold percentage is lower.

For example:

- A 100GB volume has a default grow threshold of 90% and a shrink threshold of 50%.
- A 10TB volume has a default grow threshold of 98% and a shrink threshold of 50%.

At the start of the job (after the first files are written), the capacity balance appears as shown in Figure 34.

Figure 34) Initial FlexGroup data balance—proactive resize, autosize enabled.

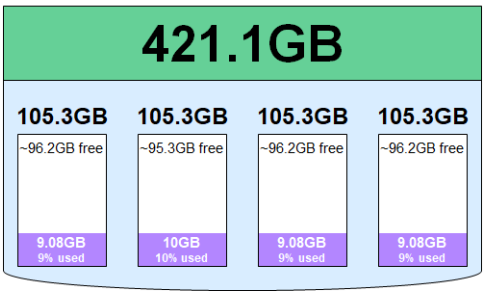
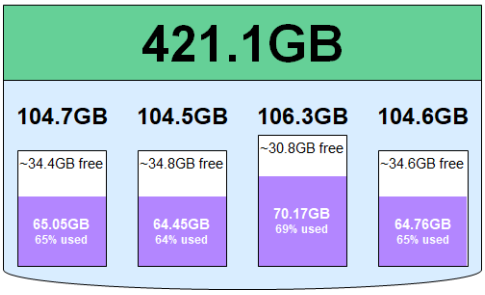
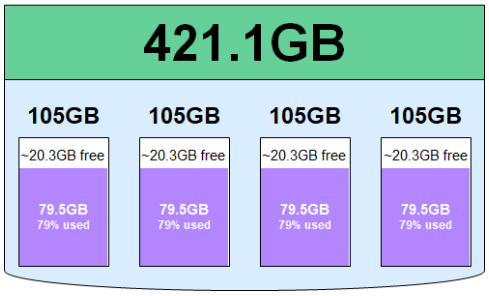


Figure 35) FlexGroup data balance, ~68% used—proactive resize, autosize enabled.



At around 66% capacity usage, the member volumes resize a bit to maintain balanced free space, but the total capacity remains the same, like when autosize is disabled.

Figure 36) FlexGroup data balance, job complete—proactive resize, autosize enabled.



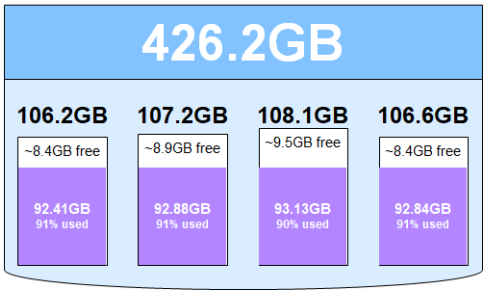
After the job finishes, ONTAP sees that the used space is even across all member volumes and proactive resizing shrinks the member volumes back down to their original sizes and makes them all the same size because there is sufficient free space available. The total FlexGroup size has not changed.

As you can see, FlexGroup volumes with autosize enabled act just like FlexGroup volumes when autosize is disabled when the free space thresholds are below where autosize kicks in.

In the preceding graphic, there is roughly ~81GB free space available in the entire FlexGroup volume. If we keep writing 10GB files, we eventually reach the autosize threshold and ONTA starts to react accordingly — this time with autosize growing the member volumes that need extra space, rather than by borrowing free space from other member volumes.

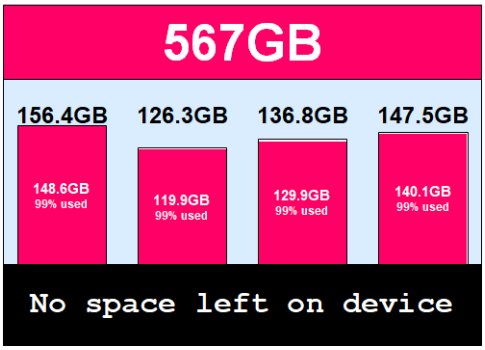
This results in an increase in the entire FlexGroup volume's capacity. In the next test run, we created a new folder in the same FlexGroup and re-ran the test that creates 32–10GB files in the FlexGroup volume across four folders.

Figure 37) FlexGroup data balance, second test run—proactive resize, autosize enabled.



After one of the FlexGroup member volumes reaches the 92% used-space threshold, autosize grows **only** that member volume. If other member volumes also need to be grown when they hit 92%, then those are also increased. This increases the overall capacity of the FlexGroup volume. Proactive resizing also adjusts the other member volume capacities up or down so that a relatively even amount of free space is available per member volume.

Figure 38) FlexGroup data balance, autosize limit—proactive resize, autosize enabled.



Autosize considerations—Smaller FlexGroup volumes

Because autosize capacity is based on percentage of total size, smaller FlexGroup volumes (such as a 420GB FlexGroup volume) have less runway for growth, by default, than a larger FlexGroup volume. The default autogrow maximum is capped to 120% of the total volume size. If the volume is ever grown manually and shrunk back down, then the autogrow value will reflect the larger volume size. Table 23 shows examples of autosize maximum sizes.

Table 23) Autosize maximum size examples.

FlexGroup volume size	Default maximum autosize	Default size delta
420GB	480GB	+80GB
100TB	120TB	+20TB
400TB	480TB	+80TB

As a result, if you are using volume autosize for FlexGroup volumes, use the following guidance:

- Use larger FlexGroup volumes and maintain the default autosize values.
- If you use smaller FlexGroup volumes, modify the default `-max-autosize` value to avoid outages.
- If you do not want your end users to have more capacity than you have provided, you can still use volume autosize if you use qtrees and quotas to limit the capacity seen and used by your end users.
- If you want to disable volume autosize, be aware that file writes fail when there is no more available space in the FlexGroup volume, even with proactive resizing in ONTAP 9.8.

Best practice 7: Combining ONTAP features for capacity management

The best way to approach capacity management involves a combination of larger FlexGroup volumes, volume autosize, ONTAP 9.8 or later, qtrees and quota enforcement. This story becomes more compelling when automatic tiering to cloud or S3 is done using FabricPool storage tiers. Using these features minimizes the capacity management overhead for storage administrators.

Qtrees

Qtrees allow a storage administrator to create folders from the ONTAP GUI or CLI to provide logical separation of data within a large bucket. Qtrees provide flexibility in data management by enabling unique export policies, unique security styles, and granular statistics.

Qtrees have multiple use cases and are useful for home directory workloads because you can name qtrees to reflect the user names of users accessing data, and you can create dynamic shares to provide access based on a user name.

The following points give more information about qtrees in FlexGroup volumes.

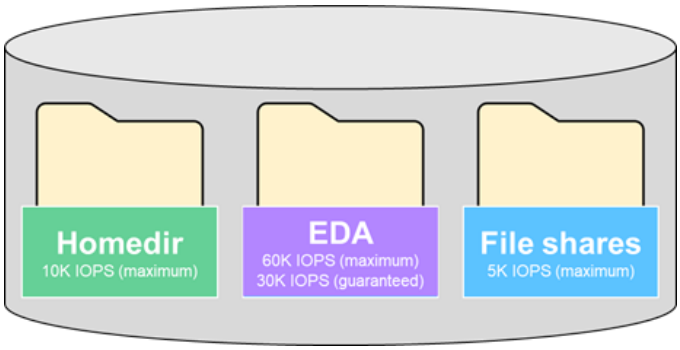
- Qtrees appear as directories to clients.
- You can create Qtrees at the volume level; you cannot currently create qtrees below directories to create qtrees that are subdirectories.
- You can create and manage Qtrees the same way a FlexVol qtree is managed.
- You cannot replicate Qtrees by using SnapMirror. Currently, SnapMirror operations are only performed at the volume level. If you want more granular replication with a FlexGroup volume, use a combination of FlexGroup volume and junction paths.
- A maximum of 4,995 qtrees is supported per volume. You can apply quota monitoring and enforcement at the qtree or user level.

  Note: ONTAP 9.5 adds quota enforcement support for FlexGroup volumes and qtree statistics. ONTAP 9.8 adds qtree QoS support for all volume types.

Qtree QoS

ONTAP 9.8 introduces the ability to apply QoS policies at the qtree level.

Figure 39) Qtree QoS use cases.



This means that you can provision a FlexGroup volume and manage performance in that volume with qtrees, rather than creating multiple FlexVol or FlexGroup volumes to divide that workload up.

Qtree QoS also provides a more granular level of statistics for the qtree than previous qtree statistics offered.

You can use qtree QoS in ONTAP 9.8 with FlexGroup volumes and FlexVol volumes, but it has the following limitations:

- NFS only
- CLI/REST API only; no current GUI support
- No adaptive QoS support

Qtree QoS also provides some enhanced statistics for performance monitoring, which aids in understanding specific workloads.

```
Policy Group             IOPS      Throughput    Latency
-------------------- -------- --------------- ----------
qtree                     113      113.00MB/s     2.82ms
```

Adaptive QoS

ONTAP 9.4 introduces adaptive QoS support for FlexGroup volumes, which allows ONTAP to adjust the IOPS and terabyte values of a QoS policy as the volume capacity is adjusted.

Note: Adaptive QoS is not supported with qtree QoS because qtrees are not objects you can grow or shrink.

Qtree statistics

Starting in ONTAP 9.5, qtree statistics are made available for FlexGroup volumes. These statistics provide granular performance information about FlexGroup volumes and their qtrees. The following example shows a statistics capture for a FlexGroup volume running a large NFS workload.

```
cluster::> statistics qtree show -interval 5  -iterations 1 -max 25 -vserver DEMO -volume
flexgroup_local

cluster : 11/7/2018 15:19:15

                                          NFS CIFS Internal *Total
               Qtree Vserver          Volume   Ops  Ops      Ops    Ops
--------------------- ------- --------------- ----- ---- -------- ------
DEMO:flexgroup_local/    DEMO flexgroup_local 22396    0        0  22396
DEMO:flexgroup_local/qtree
                         DEMO flexgroup_local     0    0        0
```

Qtrees and file moves

A qtree is considered a unique filesystem in ONTAP. Although it looks like a directory from a NAS client perspective, some operations might behave differently than if it were an actual directory. One example of this is moving a file between qtrees in the same volume.

When you move a file in a volume across directories, the file is simply renamed to a new name and it happens within seconds because it is a move inside of the same filesystem.

When a file move occurs between two qtrees, the file is copied to the new location rather than renamed. This causes the operation to take much longer.

This is a behavior that occurs whether the qtree lives in a FlexVol volume or a FlexGroup volume.

Qtree IDs and rename behavior

After a non-inherited export policy is applied to a qtree, NFS file handles change slightly when dealing with operations between qtrees. ONTAP validates qtree IDs in NFS operations, which impacts things like file renames/moves when moving to or from a qtree in the same volume as the source folder or qtree. This is considered a security feature, which helps prevent unwanted access across qtrees, such as in-home directory scenarios. However, simply applying export policy rules and permissions can achieve similar goals.

For example, a move or rename to or from a qtree in the same volume will result in an Access Denied error. The same move or rename to or from a qtree in a different volume results in the file being copied. With larger files, the copy behavior can make it seem like a move operation is taking an unusually long time, where most move operations are near-instantaneous, as they are simple file renames when in the same file system and volume.

This behavior is controlled by the advanced privilege option and is covered in detail in the NetApp Knowledge Base (KB) article Permission denied while moving files between qtrees when NFS option '-validate-qtree-export' is enabled.

From that KB, these are the behaviors of different operations.

```
Assuming that file permissions allow and that client is allowed by export policies to access both
source and destination volume/qtree, these are the current permutations with the 'validate-qtree-
export' flag enabled or disabled:

Enabled:
- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: EACCESS
- Rename between volumes where qtree IDs differ: EACCESS
- Rename between volumes where qtree IDs match: XDEV
```

Disabled:

```
- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: SUCCESS
- Rename between volumes where qtree IDs differ: XDEV
- Rename between volumes where qtree IDs match: XDEV
```

Note: NFS3ERR_XDEV and NFS3ERR_ACCESS are defined in RFC-1813.

To change the behavior of renames/moves across qtrees, modify `-validate-qtree-export` to `disabled`. See Validating qtree IDs for qtree file operations in the ONTAP 9 Documentation Center for more information.

Note: There is no known negative impact to disabling the `-validate-qtree-export` option, outside of allowing renames across qtrees.

File handle effects for qtree exports

Normally, the NFS export file handles that are handed out to clients are 32 bits or less in size. However, with qtree exports, an extra few bits are added to create 40 bit file handles. In most clients, this is not an issue, but older clients (such as HPUX 10.20, introduced in 1996) might have problems mounting these exports. Be sure to test older client connectivity in a separate test SVM before enabling qtree exports, because there is currently no way to change file handle behavior after qtree exports have been enabled.

Managing quotas with FlexGroup

NetApp FlexGroup volumes support user/group and tree quotas. The level of support for these can be broken down into the following.

- Support for quota reporting in ONTAP 9.3.
- Support for NetApp FPolicy, which can provide quota enforcement from third-party vendors, such as DefendX (formerly NTP) in ONTAP 9.4.
- Enforcement of quotas (that is, setting hard and soft limits for capacity and file count) is supported in ONTAP 9.5 and later.

User and group quota considerations

To implement user or group quotas, the cluster must be able to resolve the specified user name or group. This requirement means that the user or group must exist locally on the SVM or within a resolvable name service server, such as Active Directory, LDAP, or NIS. If a user or group cannot be found by the SVM, then the quota rule is not created. If a user or group quota fails to create because of an invalid user, the command line issues this error:

```
Error: command failed: User name user not found. Reason: SecD Error: object not found.
```

ONTAP System Manager delivers a similar message. Use the `event log show` command to investigate the issue further. For more information about configuring name services for identity management in ONTAP, see TR-4835: How to Configure LDAP in ONTAP and TR-4668: Name Services Best Practices Guide.

Creating a user or group quota

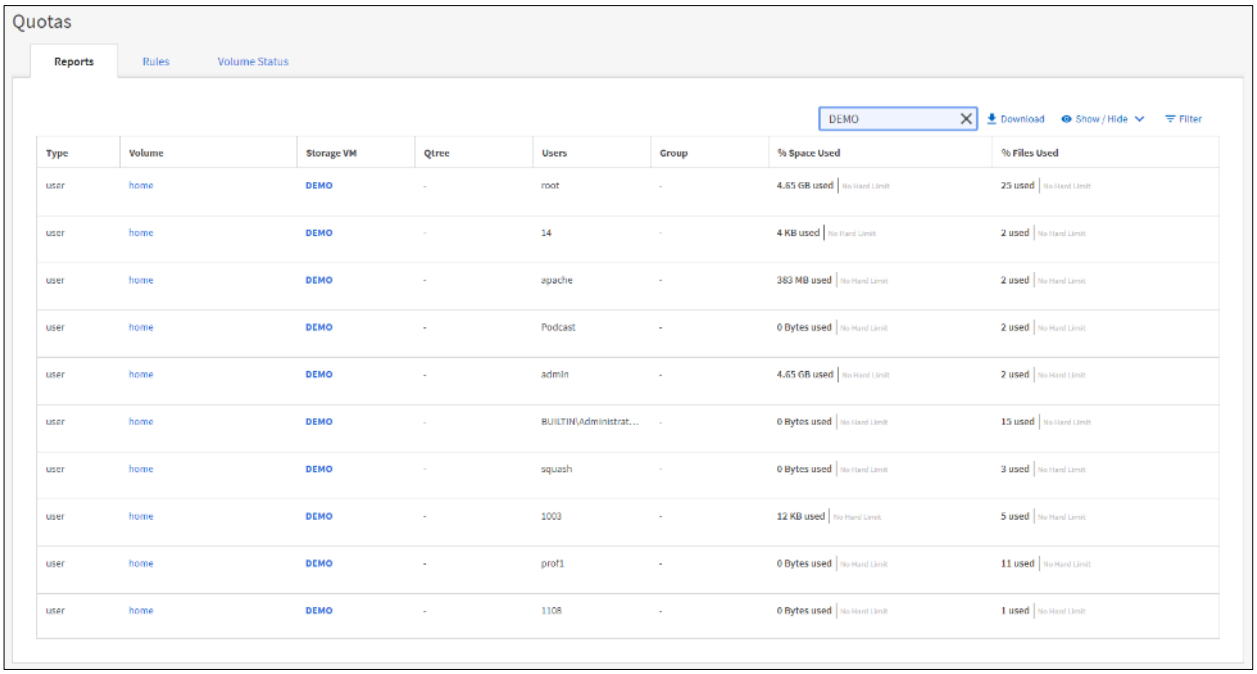
You can create user and group quotas to report or enforce capacity or file count limits on a per-user basis. These quotas are used in scenarios where multiple users or groups share the same namespace or qtree. These steps are the same for FlexVol volumes and FlexGroup volumes.

Creating a quota — ONTAP System Manager

To create user or group quota in ONTAP System Manager, on the left-hand menu, navigate to Storage > Quotas. This takes you to a page with three tabs: Reports, Rules, and Volume Status.

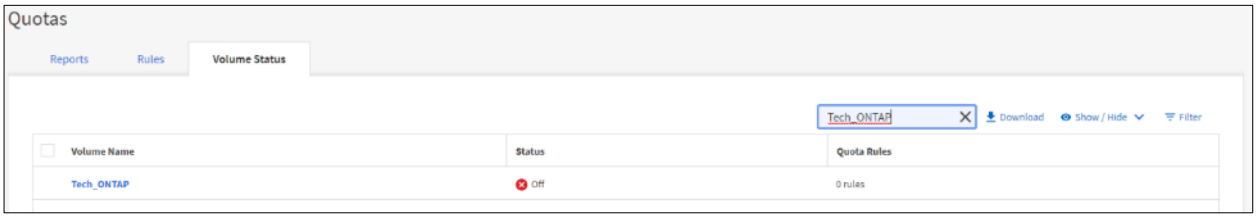
Reports shows you the current quota tracking for users, groups, and qtrees.

Figure 40) Quota reports—ONTAP System Manager.



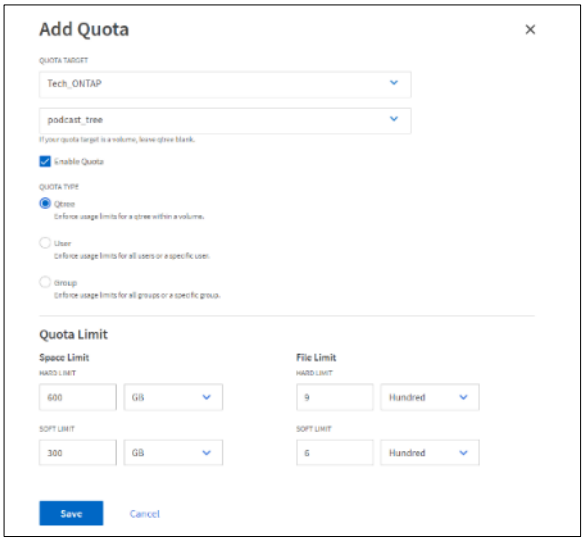
Volume Status shows whether quotas are on or off for the volume.

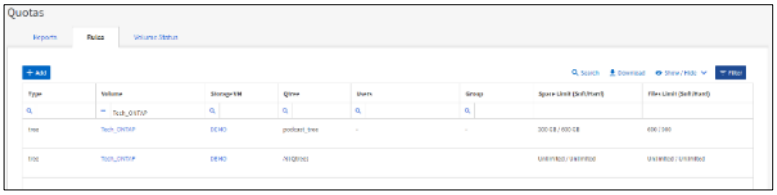
Figure 41) Quota volume status—ONTAP System Manager.

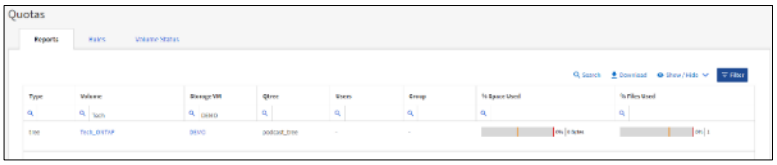


Rules is where you can create new quotas for users, groups, or qtrees. Click Add and enter the information for the user, group or qtree quota in the dialog box. After the rule is created, ONTAP System Manager performs all of the necessary steps to enable and activate the quota.

Figure 42) Quota rules—ONTAP System Manager.







Creating a user or group quota—command line

To create a user or group reporting quota with the command line for a specific user or group, use the following command at the admin privilege level:

```
cluster::> quota policy rule create -vserver SVM1 -policy-name default -volume flexgroup -type
[user|group] -target [username or groupname] -qtree ""
```

To create a user or group reporting quota with the command line for all users or groups, use the following command at the admin privilege level. The target is provided as an asterisk to indicate `all`:

```
cluster::> quota policy rule create -vserver SVM1 -policy-name default -volume flexgroup -type
[user|group] -target * -qtree ""
```

In releases earlier than ONTAP 9.5, quota enforcement is unsupported for use with FlexGroup volumes. As a result, you cannot set limits for files or disk space usage. ONTAP 9.5 enables you to set hard limits for files (`-file limit`) and capacity (`-disk-limit`) with quotas.

This example shows the `quota report` command in ONTAP 9.5 and later with FlexGroup volumes and quota enforcement:

```
cluster
cluster::> quota report -vserver DEMO
Vserver: DEMO
```

```
                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
flexgroup_local
         qtree     tree    1
                                  1.01GB    1GB       5      10   qtree
flexgroup              user    student1, NTAP\student1
                                    4KB    1GB      10      10   student1
```

Creating a tree reporting quota from the command line

To create a tree reporting quota with the command line for a specific user or group, use the following command at the admin privilege level:

```
cluster::> quota policy rule create -vserver DEMO -policy-name tree -volume flexgroup_local -type
tree -target qtree
```

To enable quotas, use `quota on` or `quota resize`.

```
cluster::> quota on -vserver DEMO -volume flexgroup_local
[Job 9152] Job is queued: "quota on" performed for quota policy "tree" on volume
"flexgroup_local" in Vserver "DEMO".

cluster::> quota resize -vserver DEMO -volume flexgroup_local
[Job 9153] Job is queued: "quota resize" performed for quota policy "tree" on volume
"flexgroup_local" in Vserver "DEMO".

cluster::> quota show -vserver DEMO -volume flexgroup_local

              Vserver Name: DEMO
               Volume Name: flexgroup_local
               Quota State: on
               Scan Status: -
          Logging Messages: -
          Logging Interval: -
          Sub Quota Status: none
  Last Quota Error Message: -
Collection of Quota Errors: -
       User Quota enforced: false
      Group Quota enforced: false
       Tree Quota enforced: true
```

The following example shows a `quota report` command on a FlexGroup volume with a tree quota specified:

```
cluster::> quota report -vserver DEMO -volume flexgroup_local
Vserver: DEMO

                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
flexgroup_local
         qtree     tree    1           0B      -       1       -   qtree
```

Files used and disk space used are monitored and increment as new files are created:

```
cluster::> quota report -vserver DEMO -volume flexgroup_local
Vserver: DEMO

                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
flexgroup_local
         qtree     tree    1
                                   13.77MB      -       4       -   qtree
```

Performance effect of using quotas

Performance effects are always a concern when enabling a feature. To alleviate performance concerns when using quotas, we ran a standard NAS benchmark test against FlexGroup volumes in ONTAP 9.5 with and without quotas enabled. We concluded that the performance effect for enabling quotas on a FlexGroup volume is negligible, as shown in Figure 43 and Figure 44.

Figure 43) ONTAP 9.5 performance (operations/sec)—quotas on and off.

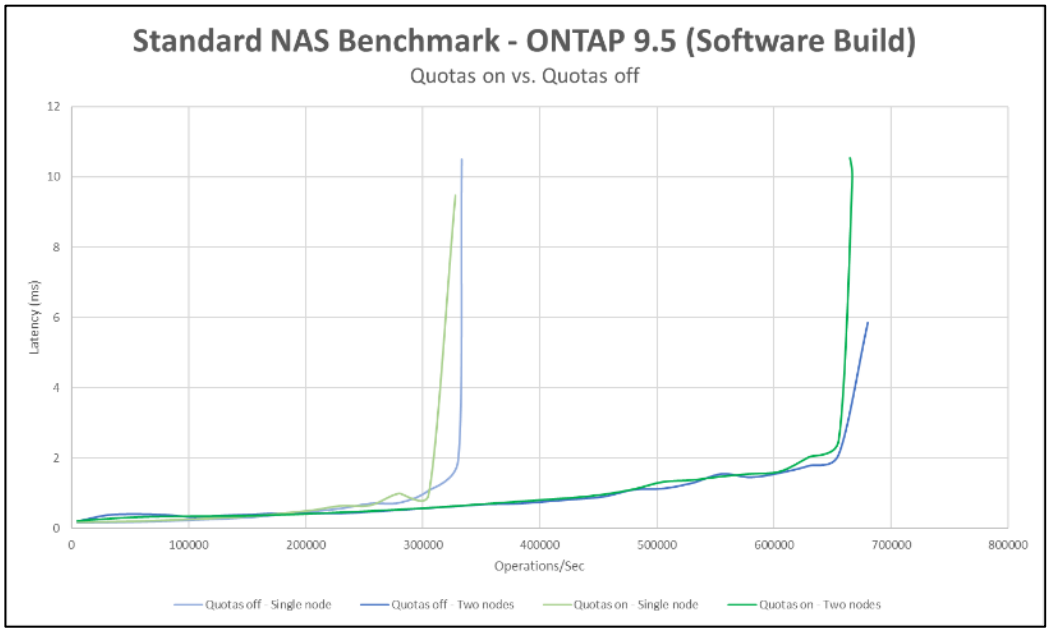
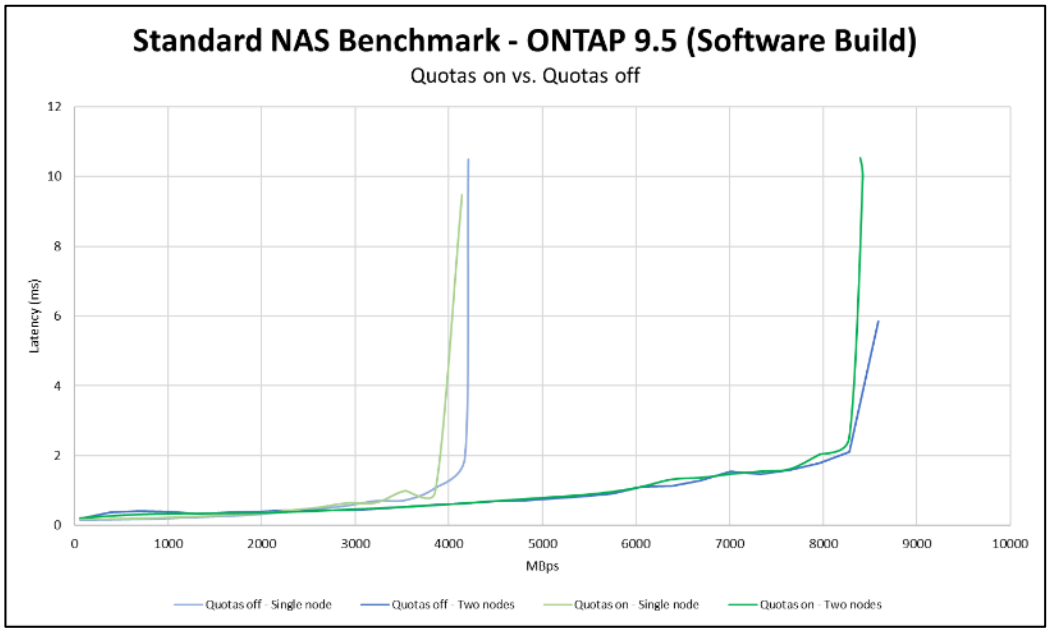


Figure 44) ONTAP 9.5 performance (MBps)—quotas on and off.



Quota scan completion times

When a quota initialization or resize takes place, ONTAP must perform some background tasks to complete the necessary work to reflect quota usage accurately. These tasks take time, which depend on a number of factors covered below.

Initialization completion time

The time it takes for quotas to initialize on a volume or qtree depends on the following factors:

- The number of files and folders in a volume. More files mean a longer initialization, while file size does not affect initialization time.
- Type of volume. FlexVol scans can take longer than FlexGroup scans, because FlexGroup quota scans are performed in parallel across the nodes on which a FlexGroup resides.
- Type of hardware and load on system. Heavily loaded systems with many files can result in scans that take hours.

You can check quota initialization status with the command `quota show -volume volname -instance`.

Quota resize completion time

Quota resize is used when a quota policy is changed. The resize operation performs a scan with the new limits. This process also has some considerations for time to completion.

- The resize operation only scans using the newly added rules, so it completes faster than an initialize operation.
- A resize operation typically completes in a matter of seconds because it has to do less than quotas on/off.
- Use the resize operation instead of toggling quotas on/off because resize completes faster.
- Quota resize can run up to 100 concurrent jobs; after 100 jobs, resize operations must wait in a queue.
- More concurrent scans can impact resize performance and add time to the job completion.

User-mapping considerations with quotas

User mapping in multiprotocol environments (data access from both SMB and NFS) for quotas occurs at the member volume level. Eventually, all member volumes agree on the user mapping. However, sometimes there might be a discrepancy, such as when user mapping fails or times out when doing a name mapping that succeeded on another member. This means that at least one member considers the user to be part of a user-mapped pair, and at least one other member considers it to be a discrete record.

At worst, enforcement of the quota rules can be inconsistent until the issue is resolved. For instance, a user might be able to briefly overrun a quota limit.

An event management system message is sent when user mapping results are coordinated.

```
cluster::*> event route show -message-name fg.quota.usermapping.result -instance

                                  Message Name: fg.quota.usermapping.result
                                      Severity: NOTICE
                             Corrective Action: (NONE)
                                   Description: This message occurs when the quota mapper
decides whether to map the Windows quota record and the UNIX quota record of a user into a single
multiuser record.
```

Tree quota considerations

SVMs in ONTAP can have a maximum of five quota policies, but only one policy can be active at a time. To see the active policy in an SVM, use the following command:

```
cluster::> vserver show -vserver DEMO -fields quota-policy
vserver quota-policy
------- ------------
DEMO    default
```

Note: Currently, you cannot view this information in ONTAP System Manager.

The default policy is adequate in most cases and does not need to be changed. When `quota on` is issued, the active policy is used—not the policy that was assigned to a volume. Therefore, it is possible to get into a situation where you think you have applied a quota and rules to a volume, but `quota on` fails.

The following example applies a quota policy to a volume:

```
cluster::*> quota policy show -vserver DEMO -policy-name tree

      Vserver: DEMO
  Policy Name: tree
Last Modified: 10/19/2017 11:25:20
    Policy ID: 42949672962

cluster::*> quota policy rule show -vserver DEMO -policy-name tree -instance

                Vserver: DEMO
            Policy Name: tree
            Volume Name: flexgroup_local
                   Type: tree
                 Target: tree1
             Qtree Name: ""
           User Mapping: -
             Disk Limit: -
            Files Limit: -
Threshold for Disk Limit: -
         Soft Disk Limit: -
        Soft Files Limit: -
```

Turning on quotas produces an error because the SVM has `default` assigned for quotas and does not contain any rules.

```
cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true

Error: command failed: No valid quota rules found in quota policy default for volume
flexgroup_local in Vserver DEMO.
```

When you add a rule to `default`, the `quota on` command works, but the SVM does not use the new tree policy.

```
cluster::*> quota policy rule create -vserver DEMO -policy-name default -volume flexgroup_local -
type tree -target ""

cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true
[Job 8063] Job succeeded: Successful

cluster::*> vserver show -vserver DEMO -fields quota-policy
vserver quota-policy
------- ------------
DEMO    default
```

To use the necessary policy, you must modify the SVM and then turn quotas off and back on.

```
cluster::*> vserver modify -vserver DEMO -quota-policy tree

cluster::*> quota off -vserver DEMO *
```

```
cluster::*> quota policy rule delete -vserver DEMO -policy-name default *
1 entry was deleted.

cluster::*> quota on -vserver DEMO -volume flexgroup_local -foreground true
[Job 8084] Job succeeded: Successful
```

This behavior is not unique to FlexGroup volumes; this happens with FlexVol volumes as well.

How clients see space when quotas are enabled

When quotas are enabled for a qtree in ONTAP, the clients only see the available space as reported by that quota. This provides a way to closely limit how much space is being used by EDA end users.

For example, this is a quota for qtree1:

```
cluster::*> quota report -vserver DEMO -volume flexgroupDS -tree qtree1
Vserver: DEMO

                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
flexgroupDS
         qtree1    tree    1           0B  500GB       1       -   qtree1
```

This is how much space that volume has:

```
cluster::*> vol show -vserver DEMO -volume flexgroupDS -fields size
vserver volume      size
------- ----------- ----
DEMO    flexgroupDS 10TB
```

This is what the client sees for space for that volume:

```
# df -h /mnt/nas2
Filesystem         Size  Used Avail Use% Mounted on
demo:/flexgroupDS  9.5T  4.5G  9.5T   1% /mnt/nas2
```

This is what is reported for that qtree:

```
# df -h /mnt/nas2/qtree1/
Filesystem         Size  Used Avail Use% Mounted on
demo:/flexgroupDS  500G     0  500G   0% /mnt/nas2
```

High file count considerations

An inode in ONTAP is a pointer to any file or folder within the file system, including Snapshot copies. Each FlexVol volume has a finite number of inodes and has an absolute maximum of 2,040,109,451 inodes.

Inodes can be increased after a FlexVol volume has been created and can be decreased only to a number that has not already been allocated.

Default and maximum inode counts

Default and maximum inode counts for volumes (both FlexVol and FlexGroup) are dependent on the total allocated capacity of the volume. For example, a 100GB FlexVol volume is able to hold as many inodes as an 8TB FlexVol volume.

Table 24 shows a sample of FlexVol volume sizes, inode defaults, and maximums.

Table 24) Inode defaults and maximums according to FlexVol size.

FlexVol size	Default inode count	Maximum inode count
20MB*	566	4,855
1GB*	31,122	249,030
100GB*	3,112,959	24,903,679
1TB	21,251,126	255,013,682
7.8TB	21,251,126	2,040,109,451
100TB	21,251,126	2,040,109,451
300TB	21,251,126	2,040,109,451

*FlexGroup member volumes should not be any smaller than 100GB in size.

Increasing maximum files—considerations

If you want to avoid monitoring and reacting to out of inode conditions, you can immediately configure high file count FlexGroup and FlexVols volumes with the maximum supported `files` value, with the following considerations in mind.

The default or maximum number of inodes on a FlexVol volume depends on the volume size and has a ratio of one inode to 4KB of capacity. This means that for every 4KB of allocated space to a volume, you can allocate one inode. Examples of these values are seen in Table 24.

In addition, each inode uses 288 bytes of capacity – this means that having many inodes in a volume can also use up a non-trivial amount of physical space in addition to the capacity of the actual data as well. If a file is less than 64 bytes, it is stored in the inode itself and will not use additional capacity.

This used space counts against the 10% aggregate reserve in ONTAP. Two billion files can use as much as ~585GB of space, and if you have many volumes set to the maximum files limit, then each volume's inode capacity will be allocated to that aggregate reserve. This capacity is only used when files are actually allocated to the volume and not by simply setting the maximum files value.

As a result, if you increase the files value to the maximum, you should pay attention to both the used inodes as well as the used aggregate space. Keeping both values in the 80% range gives the best results for high file count environments.

Other considerations:

- FlexGroup volumes are ideally the volume of choice to use for high file count environments due to their ability to nondisruptively scale when a limit has been reached.
- You can configure an approximate maximum of one inode per 4KB of allocated size, so a FlexVol or FlexGroup member volume must be approximately 7.8TB or larger in size in order to configure it with the maximum possible files setting of 2 billion.
  - In a FlexGroup volume, this means each member volume must be 7.8TB or greater.
- You should still monitor for out of inode conditions in case your environment hits the maximum supported values, and you might need to revisit the files setting any time that you grow or shrink a FlexVol volume or FlexGroup volume.
- If you choose to set the maximum files value in your volume, you should also consider setting your monitoring thresholds to 80% of the allocated inodes to give yourself ample time to plan and react before you run out of inodes.
- If the files value is set to the maximum amount on a FlexVol volume or on the individual member volumes in a FlexGroup volume and you run out of inodes, you cannot increase them further unless you are using a FlexGroup volume and add new member volumes. For this reason, avoid setting FlexVol volumes to 2 billion if possible; use FlexGroup volumes so that there is the option of adding member volumes in case you hit the 2 billion maximum value.

- Finally, keep in the mind that inode metadata is stored in the underlying aggregate, so you should monitor aggregate free space to make sure that the aggregate does not run out of space.

Easily increasing the files value to the maximum value

In many cases, you might be unsure about how many files is the maximum for a volume. In ONTAP 9.9.1, a new volume option was introduced to make setting the maximum file value simpler.

When you set the `files-set-maximum` value on a volume to true, ONTAP automatically adjusts the maxfiles to the largest possible value for you. You can only set this value to `true`—after it's set, you cannot unset it. Set the value to `true` only if you want to set the maxfiles to the largest possible value.

```
[-files-set-maximum {true|false}] - Set Total Files (for user-visible data) to the Highest Value
that the Volume can Hold (privilege: advanced)
This optionally specifies whether the volume's total number of files will be set to the highest
possible value. If true, the volume's total number of files is set to the highest value that the
volume can hold. Only <true> is a valid input. <false> is not permitted. To modify the total
number of files to a specific value, use option files.
```

For information about the implications of setting the maximum files value, see the "Increasing maximum files—considerations" section.

Default and maximum inode counts—FlexGroup volume considerations

When a default volume inode count reaches 21,251,126, it remains at that default value, regardless of the size of the FlexVol volume. This feature mitigates potential performance issues, but it should be considered when you design a new FlexGroup volume. The FlexGroup volume can handle up to 400 billion files (2 billion files x 200 FlexVol member volumes), but the **default** inode count for 200 FlexVol members in a FlexGroup volume is just 4,250,225,200.

This count is based on the following formula:

```
200 member volumes * 21,251,126 default inodes per member = 4,250,225,200 total default inodes
```

If the FlexGroup volume requires more inodes than what is presented as the default value, you must increase the inodes by using the `volume modify -files` command. As mentioned, this value can be increased to the absolute maximum value allowed, if desired, but you should follow the guidance in "Increasing maximum files—considerations".

When you use a FlexGroup volume, the total default inode count depends on both the total size of the FlexVol members and the number of FlexVol members in the FlexGroup volume.

Table 25 shows various examples of FlexGroup configurations and the resulting default inode counts.

Table 25) Inode defaults resulting from FlexGroup member sizes and member volume counts.

Member volume size	Member volume count	Default inode count (FlexGroup)
100GB	8	24,903,672
100GB	16	49,807,344
1TB	8	170,009,008
1TB	16	340,018,016
100TB	8	170,009,008
100TB	16	340,018,016

High file counts, low capacity needs

As mentioned, ONTAP allocates a default inode and maximum inode count based on volume capacity. In Table 24, member volumes smaller than 7.8TB are able to achieve the maximum 2 billion inodes. To get

2 billion inodes per member volume, the member volume capacity must be 7.8TB or greater. In a FlexGroup volume with eight member volumes and space guarantees enabled, this supports up to 16 billion files, but also provisions approximately 62.4TB of reserved storage.

If your dataset consists of very small files, you might never come close to approaching that reserved capacity and are wasting space that could be used for other workloads. For example, if all files in a workload are 288 bytes each in size, 16 billion files consume only approximately 4.6TB, which is well below the amount of capacity you need to get 16 billion files.

When deploying high file counts that use up little capacity, there are two main options for deploying the FlexGroup volume.

- **Deploy the FlexGroup volume with 7.8TB or greater member volumes with thin provisioning.** Thin provisioning a volume simply means that you are telling ONTAP a volume is a certain size, but that the size is not guaranteed in the file system. This provides flexibility in the file system to limit storage allocation to physical space. However, other volumes in the aggregate can impact the free capacity with their used space and if they have enabled space guarantees, so it is important to monitor available aggregate space when using thin provisioning. See TR-4571 for details on capacity monitoring in a FlexGroup volume.
- **Manually create the FlexGroup volume with more member volumes than the default.** If you want to keep space guarantees for the FlexGroup volume, another option for high-file-count/small capacity environments is to create more member volumes in a FlexGroup volume.

Because inode counts are limited per FlexVol member volume according to capacity, adding more smaller member volumes can provide for higher file counts at the same capacity. Table 26 shows some possible configurations. For more information about manual creation of FlexGroup volumes, see TR-4571.

Table 26) High-file-count/small capacity footprint examples—increasing member volume counts.

Total FlexGroup size	Member volume count (size)	Maximum inode count (entire FlexGroup)
80TB (no space guarantee)	8 (10TB)	16,320,875,608
64TB (space guarantee enabled)	32 (2TB)	16,320,875,608
64TB (space guarantee enabled)	64 (1TB)	16,320,875,608

Planning for high file counts in ONTAP

With utilities like the NetApp XCP Migration Tool (using the scan feature), you can evaluate your file count usage and other file statistics to help you make informed decisions about how to size your inode counts in your new FlexGroup volume. For more information about using XCP to scan files, see TR-4571 or contact ng-xcp-support@netapp.com.

Viewing used and total inodes

In ONTAP, you can view inode counts per volume by using the following command in **advanced privilege**:

```
cluster::*> volume show -volume flexgroup -fields files,files-used
vserver volume    files     files-used
------- --------- --------- ----------
SVM     flexgroup 170009008 823
```

You can also use the classic `df -i` command. To show all member volumes, use an asterisk with the volume name in **diag privilege**:

```
cluster::*> df -i Tech_ONTAP*
Filesystem               iused      ifree  %iused  Mounted on                Vserver
```

```
/vol/Tech_ONTAP/          10193  169998815      0%  /techontap                DEMO
/vol/Tech_ONTAP__0001/      923   21250203      0%  /techontap                DEMO
/vol/Tech_ONTAP__0002/     4177   21246949      0%  ---                       DEMO
/vol/Tech_ONTAP__0003/      878   21250248      0%  ---                       DEMO
/vol/Tech_ONTAP__0004/      848   21250278      0%  ---                       DEMO
/vol/Tech_ONTAP__0005/      750   21250376      0%  ---                       DEMO
/vol/Tech_ONTAP__0006/      972   21250154      0%  ---                       DEMO
/vol/Tech_ONTAP__0007/      879   21250247      0%  ---                       DEMO
/vol/Tech_ONTAP__0008/      766   21250360      0%  ---                       DEMO
```

What happens when you run out of inodes

When a volume runs out of inodes, no more files can be created in that volume until the number of inodes is increased or existing inodes are freed and the cluster triggers an event management system event (`callhome.no.inodes`). Additionally, a NetApp AutoSupport® message is triggered. Starting in ONTAP 9.3, a FlexGroup volume takes per-member inode numbers into account when deciding which member volumes are optimal for data ingest.

You can use event management system messages for monitoring, or for triggering scripts that automatically increase inode counts to help avoid space errors before they create production workload problems.

For information on increasing maximum files, see the section “Increasing maximum files—considerations”.

Async delete

ONTAP 9.8 introduces a new feature that allows storage administrators to delete entire directories from the cluster CLI, rather than needing to perform deletions from NAS clients. This provides a way to remove high file count folders much faster than by using NAS protocols, as well as removing network and client performance contention. This command works for both FlexVol and FlexGroup volumes.

In testing, `async-delete` performed almost ten times faster than single threaded `rm` commands and is slightly faster on FlexVol volumes.

Table 27) Async-delete performance.

A300 (24,000 files/folders)	rm -rf * seconds	async-delete seconds	Speed increase
FlexVol	18.3	2	9.1x
FlexGroup	32.1	3	10.7x

When a directory deletion occurs with `async delete`, a job runs and creates several tasks that run in parallel to delete the directory. By default, the job throttles to 5,000 concurrent tasks, but that amount can be decreased to a minimum of 50 or increased to a maximum of 100,000.

When a delete command is issued, ONTAP scans the specified directory. If subdirectories are found, the contents of those directories will be deleted first.

The following caveats apply:

- CLI only
- SVM and volumes must be valid
- Volume must be online and mounted
- Directory path must be valid
- Only one async-delete can be run at a time
- Must be run on a directory; cannot be run on single files

To run a delete job:

```
cluster::*> async-delete start -vserver DEMO -volume FlexGroup1 -path /files
[Job 34214] Job is queued: Asynchronous directory delete job.
```

To check the progress:

```
cluster::*> async-delete show -vserver DEMO -instance
```

Note: Because you can currently only run `async-delete` on one directory at a time, using XCP to delete files might be the better choice for some use cases.

Using XCP to delete files

XCP is a free tool offered by NetApp that can perform rapid data migration and scans but also has the capability to delete massive amounts of files in parallel from an NFS client.

For example, this is an `rm` operation performed on 37 million files. It took 20 hours:

```
# time rm -rf /flexgroup/*
real 1213m4.652s
user 1m39.703s
sys 41m16.978s
```

That same dataset took just 3.5 hours with XCP, using a single NFS client:

```
# time xcp diag -rmrf 10.193.67.219:/flexgroup_16
real 218m17.765s
user 149m16.132s
sys 40m47.427s
```

With larger clients and more robust networks (my client was a virtual machine (VM) with 8GB of RAM and just two CPUs on a 1Gb network), that completion time number can go even lower.

64-bit file identifiers

By default, NFS in ONTAP uses 32-bit file IDs. File system IDs (FSIDs) are unique identifiers in the file system that enable ONTAP to keep track of files. 32-bit file IDs are limited to 2,147,483,647 maximum signed integers, which is where the 2 billion inode limit for FlexVol volumes comes from.

FlexGroup volumes are able to support hundreds of billions of files in a single namespace by linking multiple member volumes together, but to get safely beyond the 32-bit signed integer limit of 2 billion (and remove the possibility of file ID collisions), 64-bit file IDs must be enabled.

ONTAP can hand out up to 4,294,967,295 file IDs (the 32-bit unsigned integer) in a FlexGroup volume when 32-bit file IDs are used before file ID collisions are guaranteed to occur. File ID collisions are mathematically impossible when there are 2,147,483,647 files, which is why that is the safest file count to use with 32-bit file IDs. After that value is exceeded, the likelihood of file ID collisions grows the closer the file count gets to the unsigned 32-bit integer value of 4,294,967,295. ONTAP does not prevent you from creating more than 2 billion files in a FlexGroup volume if you set the maxfiles value to a higher value. To learn more about what happens with file ID collisions, see the section “Impact of file ID collision”.

With 64-bit file IDs, ONTAP can allocate up to 9,223,372,036,854,775,807 unique file IDs to files (although, the stated supported limit for maximum files in a FlexGroup volume is 400 billion).

The 64-bit file identifier option is set to `off/disabled` by default. This was by design, to make certain that legacy applications and operating systems that require 32-bit file identifiers are not unexpectedly affected by ONTAP changes before administrators can properly evaluate their environments.

Note: Check with your application or operating system vendor for their support for 64-bit file IDs before enabling them or create a test SVM and enable it to see how applications and clients react with 64-bit file IDs. Most modern applications and operating systems can handle 64-bit file IDs without issue.

You can enable this option with the following **advanced privilege** level command, and it has NFSv3 and NFSv4 options.

```
cluster::> set advanced
cluster::*> nfs modify -vserver SVM -v3-64bit-identifiers enabled -v4-64bit-identifiers enabled
```

Alternately, you can use ONTAP System Manager to enable/disable these values.

What happens when I modify this option?

After enabling or disabling this option, you must remount all clients. Otherwise, because the FSIDs change, the clients might receive stale file handle messages when attempting NFS operations on existing mounts. For more information about how enabling or disabling FSID change options can affect SVMs in high-file-count environments, see the “Effects of file system ID changes in ONTAP” section later in this document.

Do I have to enable 64-bit file IDs?

You might notice that when you create a new FlexGroup volumes on an SVM that does not have 64-bit file IDs enabled, you get a warning that you should enable the option. However, because enabling the option forces you to remount volumes (and take an outage) and because some applications do not support 64-bit file IDs, you might not want to enable that option.

If your FlexGroup volumes do not exceed 2 billion files, you can leave this value unchanged. However, to prevent any FSID conflicts, you should also increase the inode maximum on the FlexGroup volume to no more than 2,147,483,647.

```
cluster::*> vol show -vserver SVM -volume flexgroup -fields files
```

Note: This option does not affect SMB operations and is unnecessary with volumes that use only SMB.

If your environment has volumes that need 32-bit and other volumes that require more than 2 billion files, then you can use different SVMs to host those volumes and enable or disable 64-bit file IDs as needed.

Best Practice 8: 64-bit file identifiers

NetApp strongly recommends enabling the NFS server option `-v3-64bit-identifiers` at the advanced privilege level before you create a FlexGroup volume, especially if your file system exceeds or might exceed the 2 billion inode threshold.

NFSv3 versus NFSv4.x – FSIDs

NFSv3 and NFSv4.x use different FSID semantics. Now that FlexGroup volumes support NFSv4.x, ONTAP 9.7 provides two different options for enabling/disabling 64-bit file IDs.

When you use both NFSv3 and NFSv4.x in an SVM and you want the 64-bit ID option to apply to both protocols, you must set both options.

If only one option is set and volumes are accessed by both protocols, you might see undesired behavior between protocols. For instance, NFSv3 might be able to create and view more than 2 billion files, whereas NFSv4.x sends an error when a file ID collision occurs.

The options are:

```
-v3-64bit-identifiers [enabled/disabled]
-v4-64bit-identifiers [enabled/disabled]
```

Note: If you upgrade to ONTAP 9.7 (the first release to support NFSv4.x on FlexGroup volumes), upgrade to 9.7P7 or later to avoid exposure to bug 1336512.

Using quota enforcement to limit file count

Starting with ONTAP 9.5, it is possible to set up a quota policy that prevents a FlexGroup volume from exceeding 2 billion files if 32-bit file handles are still being used by way of quota enforcement.

Because quota enforcement policies do not apply to files created below the parent volume (only monitoring/reporting policies), create a qtree inside the FlexGroup volume. Then create a quota tree rule for that qtree with 2 billion files as the limit to help reduce the risk of users overrunning the 32-bit file ID limitations. Alternately, you can create specific user or group quota rules if you know the user names/group names that will be creating files in the volume.

```
cluster::*> qtree create -vserver DEMO -volume FG4 -qtree twobillionfiles -security-style unix -
oplock-mode enable -unix-permissions 777
cluster::*> quota policy rule create -vserver DEMO -policy-name files -volume FG4 -type tree -
target "" -file-limit 2000000000
cluster::*> quota on -vserver DEMO -volume FG4
[Job 15906] Job is queued: "quota on" performed for quota policy "tree" on volume "FG4" in
Vserver "DEMO".
cluster::*> quota resize -vserver DEMO -volume FG4
[Job 15907] Job is queued: "quota resize" performed for quota policy "tree" on volume "FG4" in
Vserver "DEMO".
cluster::*> quota report -vserver DEMO -volume FG4
Vserver: DEMO

                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
FG4      twobillionfiles
                   tree    1           0B      -       1
                                                     2000000000   twobillionfiles
FG4                tree    *           0B      -       0
                                                     2000000000   *
2 entries were displayed.
```

After that is done, use file permissions and/or export policy rules to limit access and prevent users from creating files at the volume level. Apply SMB shares to the qtree rather than the volume, and NFS mounts should occur at the qtree level.

Then, as files are created in the qtree, they count against the limit.

```
[root@centos7 home]# cd /FG4/twobillionfiles/
[root@centos7 twobillionfiles]# ls
[root@centos7 twobillionfiles]# touch new1
[root@centos7 twobillionfiles]# touch new2
[root@centos7 twobillionfiles]# touch new3
[root@centos7 twobillionfiles]# ls
new1  new2  new3
cluster::*> quota report -vserver DEMO -volume FG4
Vserver: DEMO

                                    ----Disk----  ----Files-----   Quota
Volume   Tree      Type    ID        Used  Limit    Used   Limit   Specifier
-------  --------  ------  -------  -----  -----  ------  ------   ---------
FG4      twobillionfiles
                   tree    1           0B      -       4
                                                     2000000000   twobillionfiles
FG4                tree    *           0B      -       0
                                                     2000000000   *
```

For information on how to change the 64-bit file ID options in ONTAP, see TR-4571: NetApp ONTAP FlexGroup Volumes Best Practices and Implementation Guide.

Impact of file ID collision

If 64-bit file IDs are not enabled, the risk for file ID collisions increases. When a file ID collision occurs, the impact can range from a stale file handle error on the client, to the failure of directory and file listings, to

the entire failure of an application. Usually, it is imperative to enable the 64-bit file ID option when you use FlexGroup volumes.

You can check a file's ID from the client using the `stat` or `ls -i` command. When an inode or file ID collision occurs, it might look like the following. The inode is **3509598283** for both files.

```
# stat libs/
  File: `libs/'
  Size: 12288           Blocks: 24         IO Block: 65536  directory
Device: 4ch/76d Inode: 3509598283  Links: 3
Access: (0755/drwxr-xr-x)  Uid: (60317/   user1)   Gid: (10115/      group1)
Access: 2017-01-06 16:00:28.207087000 -0700
Modify: 2017-01-06 15:46:50.608126000 -0700
Change: 2017-01-06 15:46:50.608126000 -0700

# stat iterable/
  File: `iterable/'
  Size: 4096            Blocks: 8          IO Block: 65536  directory
Device: 4ch/76d Inode: 3509598283  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (60317/   user1)   Gid: (10115/      group1)
Access: 2017-01-06 16:00:44.079145000 -0700
Modify: 2016-05-05 15:12:11.000000000 -0600
Change: 2017-01-06 15:23:58.527329000 -0700

# ls -i libs
3509598283 libs

# ls -i iterable
3509598283 iterable
```

A collision can result in issues such as circular directory structure errors on the Linux client during `find` or `rm` command operations and an inability to remove files. In some cases, you might even see "`stale file handle`" errors.

```
rm: WARNING: Circular directory structure.
This almost certainly means that you have a corrupted file system.
NOTIFY YOUR SYSTEM MANAGER.
The following directory is part of the cycle:
  '/directory/iterable'

rm: cannot remove '/directory': Directory not empty
```

Note: File ID collisions impact NFS only. SMB does not use the same file ID structure.

Effects of file system ID changes in ONTAP

NFS uses a file system ID (FSID) when interacting between client and server. This FSID lets the NFS client know where data lives in the NFS server's file system. Because ONTAP can span multiple file systems across multiple nodes by way of junction paths, this FSID can change depending on where data lives. Some older Linux clients can have problems differentiating these FSID changes, resulting in failures during basic attribute operations, such as `chown` and `chmod`.

An example of this issue can be found in bug 671319. If you disable the FSID change option (for NFSv3 or NFSv4), be sure to enable the 64-bit file ID option on the NFS server (see "64-bit file identifiers"), as the total number of file IDs is now be shared across volumes in the SVM, so you run the risk of hitting file ID collisions sooner.

This FSID change option could also affect older legacy applications that require 32-bit file IDs. Perform the appropriate testing with your applications in a separate SVM before toggling FSID change.

How FSIDs operate with SVMs in high-file-count environments

The FSID change option for NFSv3 and NFSv4.x provides FlexVol and FlexGroup volumes with their own unique file systems, which means that the number of files allowed in the SVM is dictated by the number of

volumes. However, disabling the FSID change options causes the 32-bit or 64-bit file identifiers to apply to the SVM itself, meaning that the file limits with 32-bit file IDs applies to all volumes.

For example, if you have ten billion files in ten different volumes in your SVM, leaving the FSID change option enabled ensures that each volume can have its own set of unique file IDs. If you disable the FSID change option, then all ten billion files will share the pool of file IDs in the SVM. With 32-bit file IDs, you will likely see file collisions.

NetApp recommends leaving the FSID change option enabled with FlexGroup volumes to help prevent file ID collisions.

How FSIDs operate with Snapshot copies

When a Snapshot copy of a volume is created, a copy of a file's inodes is preserved in the file system for access later. The file theoretically exists in two locations.

With NFSv3, even though there are two copies of essentially the same file, the FSIDs of those files are not identical. FSIDs of files are formulated by using a combination of NetApp WAFL inode numbers, volume identifiers, and Snapshot IDs. Because every Snapshot copy has a different ID, every Snapshot copy of a file has a different FSID in NFSv3, regardless of the setting of the `-v3-fsid-change` option. The NFS RFC specification does not require FSIDs for a file to be identical across file versions.

Note: The `-v4-fsid-change` option does not apply to FlexGroup volumes in releases earlier than ONTAP 9.7, because NFSv4 is unsupported with FlexGroup volumes in those releases.

Directory size considerations: Maxdirsize

In ONTAP, there are limitations to the maximum directory size on disk. This limit is known as maxdirsize. The `maxdirsize` value for a volume is capped at 320MB, regardless of system. This means that the memory allocation for the directory size can reach a maximum of only 320MB before a directory can no longer grow larger. Directory sizes grow when file counts in a single directory increase. Each file entry in a directory counts against the allocated space for the directory. For information about how many files you can have in a single directory, see the "Number of files that can fit into a single directory with the default maxdirsize" section.

Best practice 9: Recommended ONTAP version for high-file-count environments

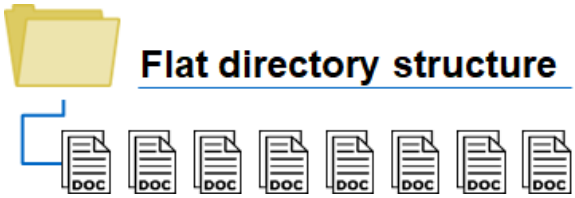
For high-file-count environments, use the latest ONTAP release available to gain the benefit of FlexGroup feature enhancements, WAFL enhancements and performance improvements for high file count workloads.

What directory structures can affect maxdirsize?

The `maxdirsize` value can be a concern when you are using flat directory structures where a single folder contains millions of files at a single level. Folder structures where files, folders, and subfolders are interspersed have a low impact on `maxdirsize`. There are several directory structure methodologies:

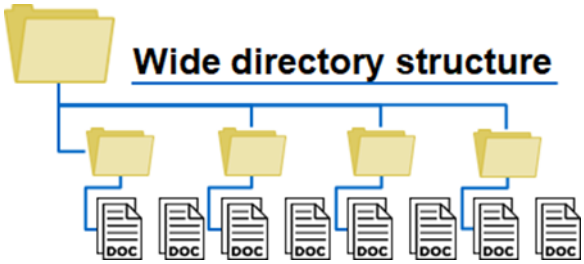
- **Flat directory structure:** a single directory with many files

Figure 45) Flat directory structure.



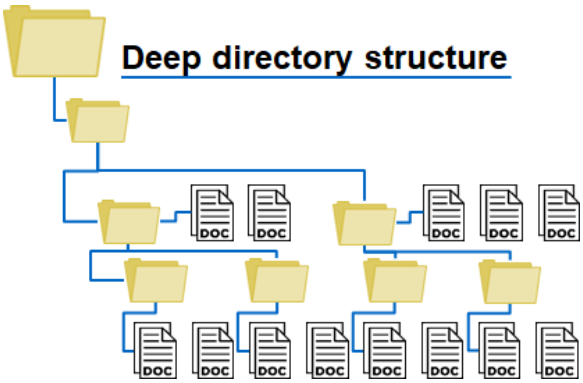
- **Wide directory structure.** Many top-level directories with files spread across directories

Figure 46) Wide directory structure.



- **Deep directory structures.** Fewer top-level directories, but with many subfolders; files spread across directories

Figure 47) Deep directory structure.



How flat directory structures can affect FlexGroup volumes

Flat directory structures (many files in a single/few directories) have a negative impact on a wide array of file systems, whether they are NetApp systems or not. Areas of impact can include, but are not limited to:

- Memory pressure
- CPU utilization
- Network performance/latency (particularly during mass queries of files, `GETATTR` operations, `READDIR` operations, and so on)

FlexGroup volumes can also have an extra impact on `maxdirsize`. Unlike a FlexVol volume, a FlexGroup volume uses remote hard links inside ONTAP to help redirect traffic. These remote hard links are what allow a FlexGroup volume to deliver scale-out performance and capacity in a cluster.

However, in flat directories, a higher ratio of remote hard links to local files is seen. These remote hard links count against the total `maxdirsize` value, so a FlexGroup volume might approach the `maxdirsize` limit faster than a FlexVol volume.

For example, if a directory has millions of files in it and generates roughly 85% remote hard links for the file system, you can expect `maxdirsize` to be exhausted at nearly twice the amount as a FlexVol volume.

Best practice 10: Directory structure recommendation
• For the best performance, avoid flat directory structures in ONTAP if at all possible. Wide or deep directory structures work best, as long as the path length of the file or folder does not exceed NAS protocol standards.

- If flat directory structures are unavoidable, pay close attention to the `maxdirsize` values for the volume and increase them as necessary with the guidance of NetApp support.
- NFS path lengths are defined by the client operating system.
- For information about SMB path lengths, see Microsoft Dev Center link.

Querying for used maxdirsize values

It is important to monitor and evaluate `maxdirsize` allocation in ONTAP. However, there are no commands for this specific to ONTAP. Bug 1336142 has been filed to add this functionality, so if you need this added to ONTAP, open a support case and have it attached to the bug.

Instead, `maxdirsize` allocation must be queried from the client.

The following command from an NFS client can retrieve the directory size information for a folder inside a FlexGroup volume for the ten largest directories in a given mount point, while omitting Snapshot copies from the search.

```
# find /mountpoint -name .snapshot -prune -o -type d -ls -links 2 -prune | sort -rn -k 7 | head
```

The following example took less than a second on a dataset in folders with millions of files:

```
[root@centos7 /]# time find /flexgroup/manyfiles/ -name .snapshot -prune -o -type d -ls -links 2
-prune | sort -rn -k 7 | head
787227871 328976 drwxr-xr-x   2 root     root     335544320 May 29 21:23
/flexgroup/manyfiles/folder3/topdir_8/subdir_0
384566806 328976 drwxr-xr-x   2 root     root     335544320 May 29 13:14
/flexgroup/manyfiles/folder3/topdir_9/subdir_0
3605793347 328976 drwxr-xr-x   2 root     root     335544320 May 29 21:23
/flexgroup/manyfiles/folder3/topdir_0/subdir_0
3471151639 328976 drwxr-xr-x   2 root     root     335544320 May 29 13:45
/flexgroup/manyfiles/folder3/topdir_4/subdir_0
2532103978 328976 drwxr-xr-x   2 root     root     335544320 May 29 14:16
/flexgroup/manyfiles/folder3/topdir_2/subdir_0
2397949155 328976 drwxr-xr-x   2 root     root     335544320 May 29 14:15
/flexgroup/manyfiles/folder3/topdir_1/subdir_0
1994984460 328976 drwxr-xr-x   2 root     root     335544320 May 29 13:43
/flexgroup/manyfiles/folder3/topdir_6/subdir_0
1860674357 328976 drwxr-xr-x   2 root     root     335544320 May 29 13:18
/flexgroup/manyfiles/folder3/topdir_5/subdir_0
1458235096 328976 drwxr-xr-x   2 root     root     335544320 May 29 14:25
/flexgroup/manyfiles/folder3/topdir_3/subdir_0
1325327652 328976 drwxr-xr-x   2 root     root     335544320 May 29 14:25
/flexgroup/manyfiles/folder3/topdir_7/subdir_0

real    0m0.055s
user    0m0.002s
sys     0m0.035s
```

Using XCP to check maxdirsize

The XCP Migration Tool is mostly considered a rapid data mover, but it also derives value in its robust file scanning capabilities. XCP is able to run `find` commands in parallel as well, so the previous examples can be run even faster on the storage system, as well as filter results to directories with specified file counts. The following XCP command example enables you to run `find` only on directories with more than two thousand entries:

```
# xcp diag find --branch-match True -fmt "'{size} {name}'.format(size=x.digest, name=x)"
localhost:/usr 2>/dev/null | awk '{if ($1 > 2000) print $1 " " $2}'
```

This XCP command helps you find the directory size values:

```
# xcp -match "type == d" -fmt "'{} {}'.format(used, x)" localhost:/usr | awk '{if ($1 > 100000)
print}' | sort -nr
```

When XCP looks for the directory size values, it scans the file system first. Here is an example:

```
[root@XCP flexgroup]#  xcp -match "type == d" -fmt "'{} {}'.format(used, x)"
10.193.67.219:/flexgroup_16/manyfiles | awk '{if ($1 > 100000) print}' | sort -nr

 660,693 scanned, 54 matched, 123 MiB in (24.6 MiB/s), 614 KiB out (122 KiB/s), 5s
 1.25M scanned, 58 matched, 234 MiB in (22.1 MiB/s), 1.13 MiB out (109 KiB/s), 10s
 …
 31.8M scanned, 66 matched, 5.83 GiB in (4.63 MiB/s), 28.8 MiB out (22.8 KiB/s), 7m52s

Filtered: 31816172 did not match
31.8M scanned, 66 matched, 5.83 GiB in (12.6 MiB/s), 28.8 MiB out (62.4 KiB/s), 7m53s.
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_9/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_8/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_7/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_6/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_5/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_4/subdir_0
336871424 10.193.67.219:/flexgroup_16/manyfiles/folder3/topdir_3/subdir_0
```

Number of files that can fit into a single directory with the default maxdirsize

To determine how many files can fit into a single directory with the default `maxdirsize` setting, use this formula:

- Memory in KB * 53 *25%

Because `maxdirsize` is set to 320MB by default on larger systems, the maximum number of files in a single directory is 4,341,760 for SMB and NFS on FlexVol volumes.

FlexGroup volumes use remote hard links to redirect I/O to member volumes. These hard links count against the total directory size, so the maximum number of files allowed with 320MB `maxdirsize` depends on the number of hard links that were created. The file count per directory might be in the 2-2.6 million range for directories in a FlexGroup volume.

NetApp strongly recommends that you keep the `maxdirsize` value at the default value.

Event management system messages sent when maxdirsize is exceeded

The following event management system messages are triggered when `maxdirsize` is either exceeded or close to being exceeded. Warnings are sent at 90% of the `maxdirsize` value and you can view them with the `event log show` command or in the ONTAP System Manager event section. You can use Active IQ Unified Manager to monitor `maxdirsize`, trigger alarms, and send a notification before the 90% threshold. These event management system messages also support SNMP traps.

```
wafl.dir.size.max
wafl.dir.size.max.warning
wafl.dir.size.warning
```

Impact of increasing the maxdirsize value

When a single directory contains many files, lookups (such as in a find operation) can consume large amounts of CPU and memory. Starting in ONTAP 9.2, directory indexing creates an index file for directory sizes exceeding 2MB to help offset the need to perform so many lookups and avoid cache misses. Usually, this helps large directory performance. However, for wildcard searches and `readdir` operations, indexing is not of much use. When possible, use the latest version of ONTAP for high file count environments to gain benefits from WAFL improvements.

Best practice 11: Maxdirsize maximums

Values for `maxdirsize` are hard coded not to be able to exceed 4GB. To avoid performance issues, NetApp recommends setting `maxdirsize` values no higher than 1GB.

Do FlexGroup volumes bypass maxdirsize limitations?

In FlexGroup volumes, each member volume has the same `maxdirsize` setting (which is configured at the FlexGroup level). Even though the **files** in a directory could potentially span multiple FlexVol member volumes and nodes, the **directory itself** resides on a single member volume. As a result, the same `maxdirsize` limitations you see in a FlexVol volume still come into play with a FlexGroup volume. This is because **directory size** is the key component, not the volume. In a FlexGroup volume, because a directory resides in a single FlexVol member volume, there is no relief for environments facing `maxdirsize` limitations.

Best practice 12: Avoiding maxdirsize issues

Newer systems offer more memory and CPU capacity, and AFF systems provide performance benefits for high file count environments. However, the best way to reduce the performance effect in directories with large numbers of files is to spread files across more directories in the file system.

Effect of exceeding maxdirsize

When `maxdirsize` is exceeded in ONTAP, an `out of space` error (`ENOSPC`) is issued to the client and an event management system message is triggered. This error can be misleading to storage administrators, as they imply an actual capacity issue when the problem in this case has to do with file count. Always check the ONTAP event log to narrow down problems when clients report seeing capacity issues.

To remediate a directory size issue, a storage administrator must increase the `maxdirsize` setting or move files out of the directory. For more information about remediation, see KB 000002080 on the NetApp Support site.

File System Analytics

ONTAP 9.8 introduces a new feature that provides a way for storage administrators to get instant access to file and directory information from ONTAP System Manager called NetApp ONTAP File System Analytics.

This initial release of File System Analytics includes information such as:

- File sizes
- Folder sizes
- Atime and mtime histograms
- File and folder listings
- Inactive/active data reporting
- File and directory counts

This information is gathered by ONTAP as the file system is updated after an initial scan is performed and takes minimal system resources to use. File System Analytics are off by default and can be enabled (and disabled) through ONTAP System Manager from the Explorer tab on the volume page for both FlexVol and FlexGroup volumes, regardless of the NAS protocol in use.

Figure 48) File System Analytics: Enable Analytics.

After analytics are enabled and the initial scan completes (completion time depends on file and folder count), you can browse the entire directory structure by clicking through the directory trees in ONTAP System Manager's Explorer tab.

Figure 49) File System Analytics: Directory and file information.

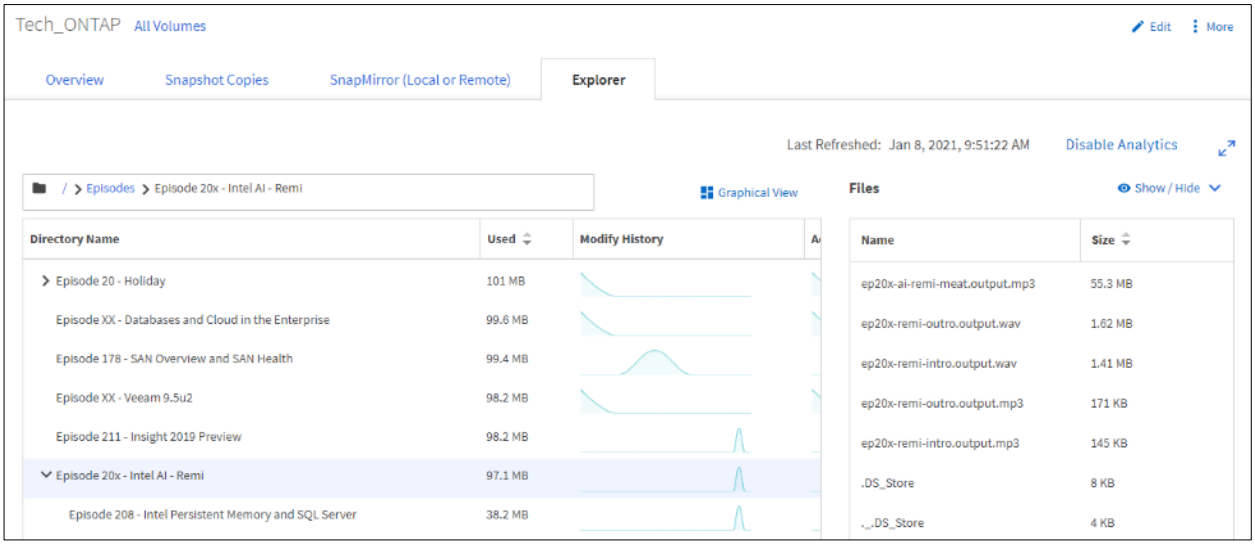
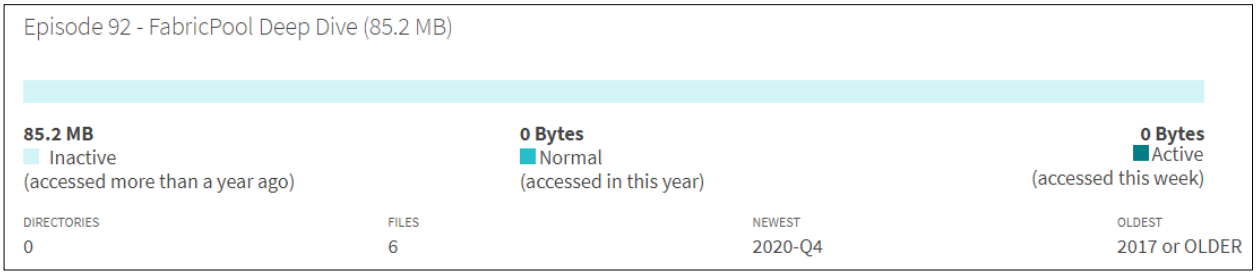


Figure 50) File System Analytics: Inactive/active data.



When files and folders are created or deleted, File System Analytics updates the tree in seconds with the new information. File System Analytics allows storage administrators to get file and folder information without the need to use off-box utilities or commands such as `du`, `find`, and `ls`, which can be time-intensive in high file count environments.

For more information on File System Analytics, including best practices and REST API examples, review the following resources:

- ONTAP File System Analytics: Better visibility for better management
- File System Analytics overview
- TR-4867: Best Practice Guide for File System Analytics
- TR-4863: Best Practice Guidelines for XCP
- Tech ONTAP Podcast Episode 270: NetApp ONTAP File System Analytics (audio podcast)

Special character considerations

Most common text characters in Unicode (when they are encoded with UTF-8 format) use encoding that is equal to or smaller than three bytes. This common text includes all modern written languages, such as Chinese, Japanese, and German. However, with the popularity of special characters such as the emoji,

some UTF-8 character sizes have grown beyond three bytes. For example, a trophy symbol is a character that requires four bytes in UTF-8 encoding.

Special characters include, but are not limited to, the following:

- Emojis
- Music symbols
- Mathematical symbols

When a special character is written to a FlexGroup volume, the following behavior occurs:

```
# mkdir /flexgroup4TB/🏆
mkdir: cannot create directory ‘/flexgroup4TB/\360\237\217\206’: Permission denied
```

In the preceding example, `\360\237\217\206` is hex `0xF0 0x9F 0x8F 0x86` in UTF-8, which is a trophy symbol.

ONTAP software does not natively support UTF-8 sizes that are greater than three bytes in NFS, as indicated in bug 229629. To handle character sizes that exceed three bytes, ONTAP places the extra bytes into an area in the operating system known as `bagofbits`. These bits are stored until the client requests them. Then the client interprets the character from the raw bits. FlexVol volumes support `bagofbits`, and FlexGroup volumes added support for `bagofbits` in ONTAP 9.2.

Best practice 13: Special character handling in FlexGroup volumes

For optimal special character handling with FlexGroup volumes, use ONTAP 9.5 or later and the utf8mb4 volume language.

Also, ONTAP has an event management system message for issues with `bagofbits` handling, which includes how to identify the offending file ID.

```
Message Name: wafl.bagofbits.name
Severity: ERROR

Corrective Action: Use the "volume file show-inode" command with the file ID and volume name
information to find the file path. Access the parent directory from an NFSv3 client and rename
the entry using Unicode characters.

Description: This message occurs when a read directory request from an NFSv4 client is made to a
Unicode-based directory in which directory entries with no NFS alternate name contain non-Unicode
characters.
```

To test `bagofbits` functionality in a FlexGroup volume, use the following command:

```
# touch "$(echo -e "file\xFC")"
```

In ONTAP 9.1, this command fails:

```
# touch "$(echo -e "file\xFC")"
touch: cannot touch `file\374': Permission denied
```

In ONTAP 9.2 and later, this command succeeds:

```
# touch "$(echo -e "file\xFC")"
# ls -la
-rw-r--r--.  1 root root    0 May  9  2017 file?
```

Support for utf8mb4 volume language

As mentioned previously, special characters might exceed the supported three bytes UTF-8 encoding that is natively supported. ONTAP then uses the `bagofbits` functionality to enable these characters to work.

This method for storing inode information is not ideal, so starting in ONTAP 9.5, utf8mb4 volume language support is added. When a volume uses this language, special characters that are four bytes in size is stored properly and not in `bagofbits`.

Volume language is used to convert names sent by NFSv3 clients to Unicode, and to convert on-disk Unicode names to the encoding expected by NFSv3 clients. In legacy situations in which NFS hosts are configured to use non-UTF-8 encodings, you will want to use the corresponding volume language. Use of UTF-8 has become almost universal these days, so the volume language is likely to be UTF-8.

NFSv4 requires use of UTF-8, so there is no need to use non-UTF-8 encoding for NFSv4 hosts. Similarly, CIFS uses Unicode natively, so it works with any volume language. However, use of utf8mb4 is recommended because files with Unicode names above the basic plane are not converted properly on non-utf8mb4 volumes.

Volume language can only be set on a volume at creation by using the `-language` option. You cannot covert a volume's language. To use files with a new volume language, create the volume and migrate the files by using a utility like the XCP Migration Tool.

Best practice 14: UTF-8 or utf8mb4?

If you are running ONTAP 9.5 or later, it is best to use the utf8mb4 volume language to help prevent issues with file name translation unless clients are unable to support the language.

Managing slow directory listings through NFS in high-file-count environments

Some workflows in high-file-count environments include running `find`, `ls`, or other read metadata-heavy operation on an existing dataset. This type of workload is inefficient and can take a long time to complete. If it is necessary to run these operations, there are a few things you can try to help speed things along.

Generally speaking, the issue with these types of operations is client, protocol, or network related. The storage rarely is the bottleneck for read metadata slowness. ONTAP is able to multithread read metadata operations. With `ls` operations, `GETATTR` requests are sent one at a time, in serial, which means for millions of `GETATTR` operations there might be millions of network requests to the storage. Each network request will incur `n` milliseconds of latency, which adds up over time.

As such, there are a few ways to speed these up:

- **Send more `GETATTR` requests at a time.** By itself, `ls` cannot send requests in parallel. But with utilities such as the XCP Migration Tool, it is possible to send multiple threads across the network to greatly speed up `ls` operations. Using XCP scan can help with speed, depending on what the `ls` output is being used for later. For example, if you need the user permissions/owners of the files, using `ls` by itself might be a better fit. But for sheer listing of file names, an XCP scan is preferable.
- **Add more network hardware (for example, 200GB instead of 10GB) to reduce round-trip time (RTT).** With larger network pipes, more traffic can be pushed over the network, thus reducing load, and potentially reducing overall RTT. With millions of operations, even shaving off a millisecond of latency can add up to a large amount of time saved for workloads.
- **Run `ls` without unnecessary options, such as highlighting/colors.** When running `ls`, the default behavior is to add sorting, colors, and highlighting for readability. These add work for the operation, so it might make sense to run `ls` with the `-f` option to avoid those potentially unnecessary features.
- **Cache `GETATTR` operations on the client more aggressively.** Client-side caching of attributes can help reduce the network traffic for operations, as well as bring the attributes local to the client for operations. Clients manage NFS caches differently, but in general, avoid setting `noac` on NFS mounts for high-file-count environments. Also, if possible, keep `actimeo` to a level no less than 30 seconds if your workload has a high number of `GETATTR` operations.

- **Create FlexCache volumes.** FlexCache volumes are able to create instant caches for read-heavy workloads. Creating FlexCache volumes for workloads that do a lot of read metadata operations, such as `ls`, can have the following benefits:
  - For local clusters, it can help offload the read metadata operations from the origin volume to the cache volumes, and, as a result, frees the origin volume up for regular reads and writes.
  - FlexCache volumes can reside on any node in a cluster, so creating FlexCache volumes makes the use of cluster nodes more efficient by allowing multiple nodes to participate in these operations, in addition to moving the read metadata operations away from the origin node.
  - For remote clusters across a WAN, FlexCache volumes can provide localized NFS caches to help reduce WAN latency, which can greatly improve performance for read-metadata-heavy workloads.

  When using FlexCache volumes to help read metadata workloads, be sure to disable `fastreaddir` on the nodes that use FlexCache volumes.

```
cluster::> node run "priv set diag; flexgroup set fastreaddir=false persist"
```

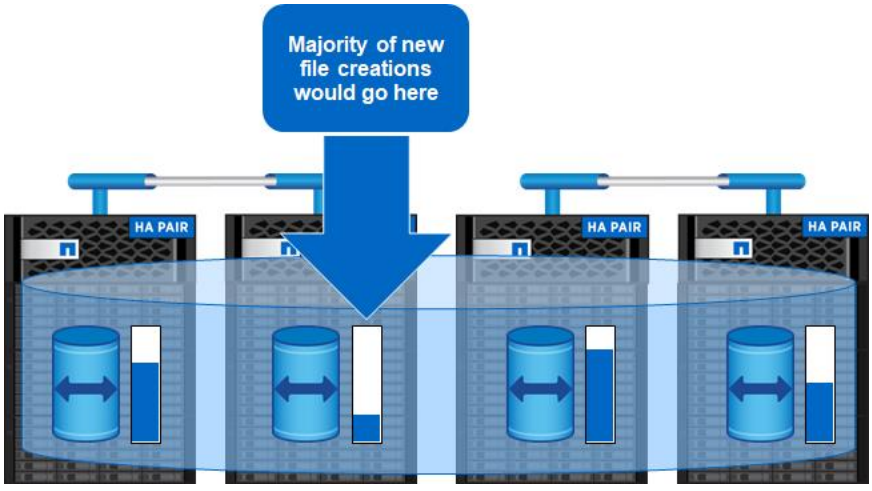
Note: For this to take effect, a reboot/storage failover is required.

Note: Starting in ONTAP 9.7, FlexGroup volumes can be origins for FlexCache volumes. For more information about FlexCache volumes, see TR-4743: FlexCache in NetApp ONTAP.

File deletions/FlexGroup member volume balancing

A FlexGroup volume spreads data across multiple member volumes relatively evenly on ingest of data. This data layout can help file deletions operate more efficiently on a FlexGroup volume as compared to a FlexVol volume, as the system is able to use more hardware and WAFL affinities to spread out the delete load more efficiently and use less CPU per node for these operations.

Figure 51) Capacity imbalance after deletion of larger files.



However, overall performance of file deletions might be slower because of remote access across the FlexGroup volume as compared to FlexVol volumes. In rare cases, the deletion of files (especially sets of large files) can create artificial hot spots in a FlexGroup volume by way of capacity imbalances.

You can view a FlexGroup volume's workload balance by using the following `diag-privilege-level` command:

```
cluster::*> set diag
```

```
cluster::*> node run * flexgroup show [flexgroup name]
```

This displays the following output:

- Member volume dataset ID (DSID)
- Member volume capacities (used and available, in blocks)
- Member volume used %
- Urgency, target, and probability percentages (used in ingest calculations)

Rebalancing data within a FlexGroup volume

Beginning with ONTAP 9.12.1, you can rebalance FlexGroup volumes by non-disruptively moving files from one constituent in a FlexGroup to another constituent, but in most cases, it is not necessary. ONTAP generally does a good job of balancing the ingest load so that new writes redirect to less full member volumes and with the proactive resizing feature available in ONTAP 9.8, ONTAP grows and shrinks member volumes as needed to maintain an even buffer of available free space so that a rebalance is not necessary. A data imbalance does not mean that there will be a performance issue unless the data imbalance is also accompanied by very full member volumes or localized hotspot files. FlexGroup volumes are optimized to maximize performance and reduce hotspots. Member volumes of different capacities are normal, should be expected, and do not indicate performance issues by themselves.

In the rare case in which a member volume grows significantly larger than other member volumes, you should analyze the workload to see if anything has changed (for instance, the workload went from creating 1MB files to 100GB files). You can use the XCP Migration Tool to scan folders and files to identify file sizes and anomalies. One common scenario that can over allocate a single member volume is when an end user zips up a large amount of data in the FlexGroup volume. The single zip file might grow to be very large and can fill up a member volume.

After the files are identified, either delete them, move them to other volumes, add space to the member volumes, or add new member volumes to help balance the ingest load in a FlexGroup volume. Ideally, upgrade the cluster to ONTAP 9.8 to gain the benefits of proactive resizing, which helps remove the management overhead for member volume capacity.

Why a FlexGroup volume does not need to rebalance existing data

As a FlexGroup volume ingests data, it has three goals:

- To encourage all its member FlexVol volumes to participate in hosting the workload in parallel. If only a subset of member volumes is active, the FlexGroup volume should distribute more new data toward the underactive members.
- To prevent any member FlexVol volume from running out of free space unless all other members are also out of free space. When one member has more data than others, the FlexGroup volume should align the underused members by placing new data on them at a higher-than-average rate.
- To minimize the performance losses caused by pursuing the previous two goals. If the FlexGroup volume placed each new file carefully and accurately where it was most beneficial, then the previous two goals could be easily achieved. However, the cost of all that careful placement appears as increased service latency. An ideal FlexGroup volume blends performance with capacity balance but favors performance.

Some of these goals are in conflict, so ONTAP employs a sophisticated set of algorithms and heuristics to maintain a balance in the FlexGroup volumes. However, in some scenarios, particularly on earlier ONTAP releases, imbalances such as the following might occur:

- Large files or files that grow over time might be present in a FlexVol member volume.
- A workload changes from smaller files to large files (such as a change in how video surveillance cameras record from 4K resolution to 8K resolution).

- Many files might be zipped or tarred into a single file in the same FlexGroup volume as the files themselves.
- A large amount of data might be deleted, and most of that data could be from the same member volume (rare).

Beginning in ONTAP 9.16.1, enabling advanced capacity balancing (-granular-data advanced) largely eliminates balance and capacity issues associated with large files monopolizing capacity in individual member volumes. When enabled, advanced capacity balancing splits large files into multiple 10GB parts (parts can be as small as 1GB for volumes under 100GB in size) that are written to multiple member volumes.

Note: You cannot revert to a release earlier than ONTAP 9.16.1 if advanced capacity balancing is enabled. If you need to revert, you must first restore from a snapshot created before advanced capacity balancing was enabled.

Customers who have not enabled advanced capacity balancing should be aware of the impact that individual large files might have on the capacity of individual member volumes.

Even when advanced capacity balancing is not enabled, when ONTAP detects that an individual member volume has an imbalance of capacity or files, it will take extra measures to provision new writes to under-allocated member volumes. In these edge-case scenarios, performance may be affected for new file writes as they are written to non-local member volumes. Data previously written to the local member volume should see little to no performance impacts.

When a FlexGroup volume needs to be rebalanced

FlexGroup rebalancing helps redistribute capacity when imbalances develop over time due to the addition or growth of large files. (By default, minimum file size considered for rebalancing is 100 MB.) After you manually start the rebalance operation, ONTAP selects the files and moves them automatically and non-disruptively.

Rebalancing is available only when all nodes in the cluster are running ONTAP 9.12.1 or later releases. You must enable granular data functionality on any FlexGroup volume that runs the rebalancing operation. Once that functionality is enabled, you cannot revert to ONTAP 9.11.1 and earlier versions unless you delete this volume or restore from a Snapshot copy that was created before the setting was enabled.

Beginning with ONTAP 9.14.1, ONTAP introduces an algorithm to non-disruptively and proactively move files in volumes that have granular data enabled without user interaction. The algorithm operates in very specific, targeted scenarios to alleviate performance bottlenecks. Scenarios where this algorithm might act include very heavy write load on a particular set of files on one node in the cluster or a continually growing file in a very hot parent directory.

To learn more about FlexGroup rebalancing, see: https://docs.netapp.com/us-en/ontap/flexgroup/manage-flexgroup-rebalance-task.html

You should be aware that FlexGroup rebalancing degrades system performance when large numbers of files are moved as part of a single rebalancing event or over multiple rebalancing events because of the creation of multi-part inodes. Every file moved as part of a rebalancing event has 2 multi-part inodes associated with that file. The larger the number of files with multi-part inodes as a percentage of the total number of files in a FlexGroup, the greater the performance impact. Certain use cases, such as a FlexVol to FlexGroup conversion, can result in a significant amount of multi-part inode creation.

FlexGroup rebalancing should not be used to rebalance FlexGroup volumes after a FlexVol to FlexGroup conversion. Instead, you can use the disruptive retroactive file move feature available in ONTAP 9.10.1 and later, by entering the volume rebalance file-move command. For command syntax, see the volume rebalance file-move start man page.

Listing files when a member volume is out of space

If a FlexGroup member volume runs out of space, the entire FlexGroup volume reports that it is out of space. Even read operations, such as listing the contents of a folder, can fail when a FlexGroup member is out of space.

Although `ls` is a read-only operation, FlexGroup volumes still require a small amount of writable space to allow it to work properly. ONTAP uses that storage to establish metadata caches. For example, suppose the name `foo` points to an inode with X properties, and the name `bar` points to an inode with Y properties. The amount of space used is negligible—a few kilobytes, or maybe a few megabytes on large systems—and this space is used and released repeatedly. Internally, this space is called the RAL reserve.

Under normal circumstances, even if you manage to fill up a member volume, a bit of space is left for the FlexGroup volume to use as it performs read-only operations like `ls`. However, ONTAP prioritizes other operations over the RAL reserve. If a member volume is 100% full, for example, and you create a Snapshot copy and then try to continue using the volume, the WAFL Snapshot reserve is used as you overwrite blocks and therefore consumes more space. ONTAP prioritizes the Snapshot space and takes space from things like the RAL reserve. This scenario rarely occurs, but it explains why an operation like `ls` might fail because of lack of space.

File rename considerations

FlexGroup volumes handle most high-metadata workloads well. However, with workloads that do a large amount of file renames at a time (for example, hundreds of thousands), performance of these operations suffers in comparison to FlexVol volumes. This is because a file rename does not move the `file` in the file system; instead, it just moves the `file name` to a new location. In a FlexGroup volume, moving this name likely takes place as a remote operation and creates a remote hard link. Subsequent renames create more remote hard links to the file's location, which keep adding latency to operations that occur on that file. If an application's workflow is mostly file renames, you should consider using FlexVol volumes instead of FlexGroup volumes. If the desired final landing spot is a FlexGroup volume after the rename occurs, consider moving the files from the FlexVol volume to the FlexGroup volume after the rename process.

Symlink considerations

If your workload contains many symlinks (that is, symlink counts in the millions) in a single FlexGroup volume, attempts to resolve that many symlinks might have a negative effect on performance. The negative effect is caused by creating remote hard links artificially in addition to the remote hard links ONTAP creates.

Best practice 15: Symlinks in FlexGroup volumes
Try to keep the number of symlinks below a few thousand per FlexGroup if possible.

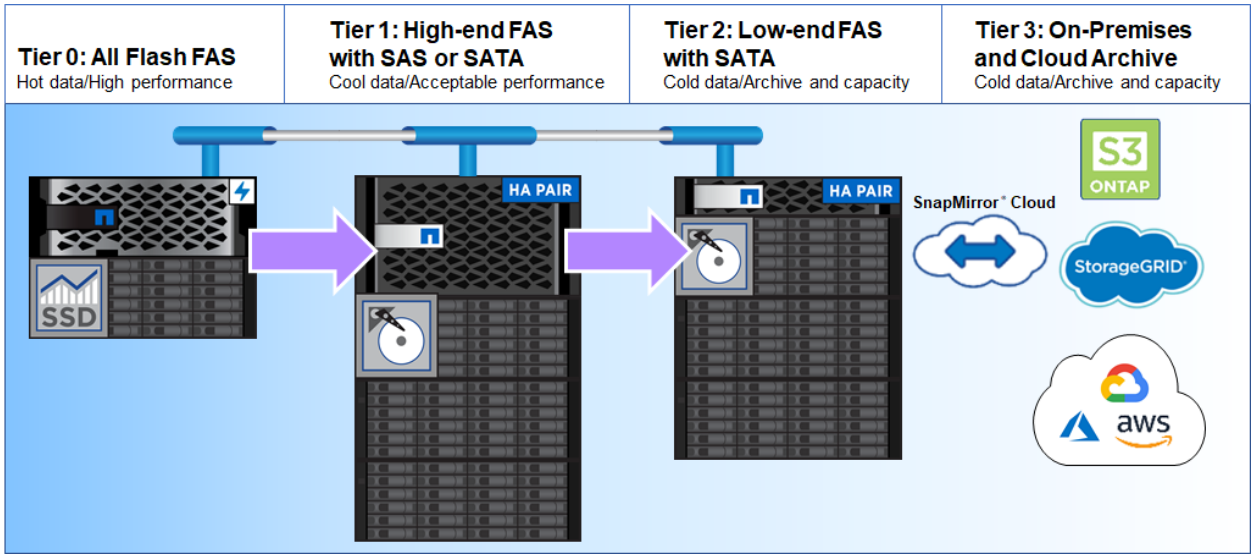
Project tiering considerations

An ONTAP cluster can consist of up to 24 nodes in NAS-only environments and up to 12 nodes when SAN is present. These nodes can be a mixture of high-performance nodes, such as AFF, and less-expensive nodes with spinning disk for capacity needs. You can architect clusters to tier workloads based on SLAs, performance needs, capacity requirements, and many other considerations.

For example, you can provision a project on capacity nodes to start, and as performance needs to be ramped up, move the project nondisruptively to the AFF nodes in the cluster for high-throughput, low-latency results. When the project lifecycle is complete, use `volume move` to relocate the data to the less expensive nodes, or replicate it to a disaster recovery site by using SnapMirror technology.

In ONTAP 9.2 and later, you can take your project tiering needs to the cloud with FabricPool and tier cold data from Snapshot copies or SnapMirror destinations to S3 buckets, whether in the cloud or on premises with NetApp ONTAP S3 or NetApp StorageGRID® using NetApp SnapMirror Cloud or SnapMirror to S3. For more information about FabricPool, see TR-4598: FabricPool Best Practices.

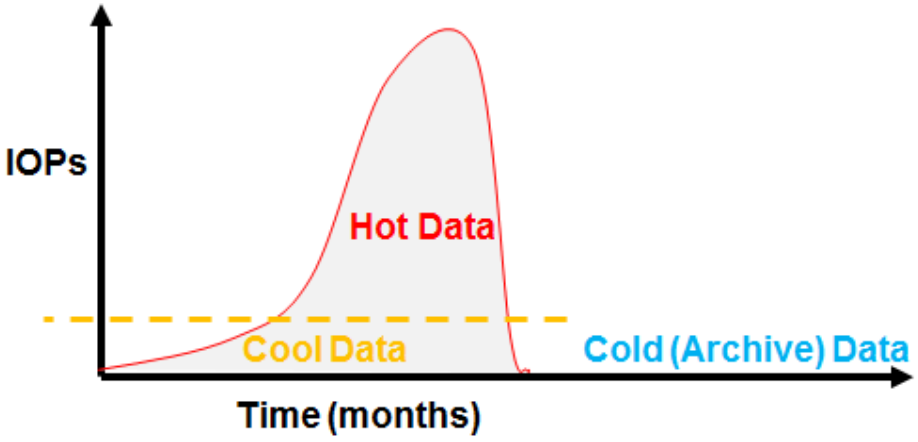
Figure 52) Cost benefits of project tiering.



Data lifecycle management

When considering data lifecycle, you can think in terms of hot, cool, and cold.

Figure 53) Project lifecycle.



Hot data is the latest project builds or the active jobs that are running. These workloads require the best possible performance, which is delivered with performance flash storage in ONTAP, ideally in parallel with FlexGroup volumes.

Cool data is a recent build or release that has just finished production. The data is still being actively accessed, but high performance is not necessarily a requirement anymore. These workloads can survive on capacity flash, AFF C-Series.

Cold data is archived projects and datasets. This type of data is accessed very rarely and can live on cheaper and deeper storage, such as AFF C-Series, Hybrid FAS, or live in S3 object storage, such as ONTAP S3, StorageGRID or cloud storage.

Table 28) Storage tiers.

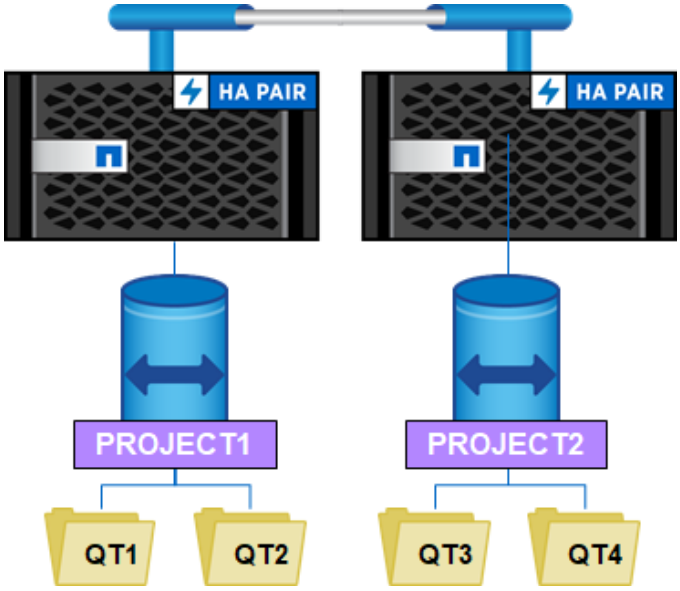
Value	Performance	Type
Hot	Fastest	AFF A-Series (Performance Flash)
	Faster	AFF C-Series (Capacity Flash)
	Fast	Hybrid FAS
Cool	Good	AFF C-Series (Capacity Flash)
	Good Enough	Hybrid FAS
Cold	Archive	AFF C-Series / Hybrid FAS / Object (S3)

Data lifecycle management challenges

In many cases, build releases are kept in individual directories. In ONTAP, you can set these directories up as qtrees, if desired, to leverage more granular export policy rules and quotas.

The downside of this approach is that data in directories and qtrees have gravity and is hard to move around easily. Tools such as rsync and/or XCP are necessary to migrate these large directories with many files. When dealing with live data, that challenge is even greater, because projects can ill afford downtime simply to migrate data. This makes project tiering difficult and untenable. Backups become time based rather than data driven, and IT teams end up backing up too much data, due to complications of the build tree structures.

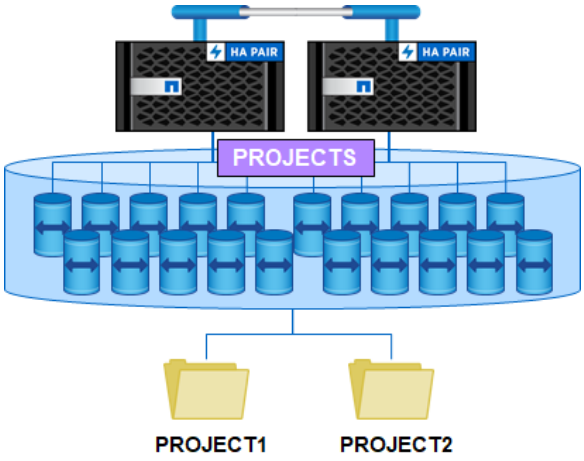
Figure 54) Build releases using qtrees with FlexVol volumes.



However, the benefits of using qtrees and FlexGroup volumes might overwhelm the downsides. Rather than worrying about a single node's performance and capacity, a FlexGroup volume automatically distributes data workloads across multiple nodes and provides more efficient use of your storage resources. When you add qtrees to the mix, you maintain the data distribution but add a number of benefits that might overwhelm any downsides due to data gravity and migration. In Figure 56, although

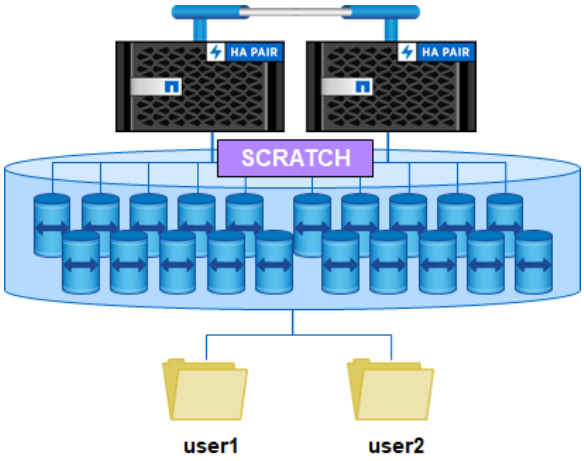
only two qtrees are shown, you can have as many as 4,995 qtrees in a single volume. See Table 29 for the pros and cons of using volumes compared to qtrees for project storage.

Figure 55) Build releases using qtrees with FlexGroup volumes.



When using scratch space with FlexGroup volumes, it is best to span multiple similar nodes and assign each user workspace a qtree. Then, enable quotas for space and file usage monitoring and enforcement so that a single user does not overrun other user's capacity. In addition, in ONTAP 9.8 you can set qtree QoS policies for these user qtrees to control performance for each user in the same FlexGroup volume.

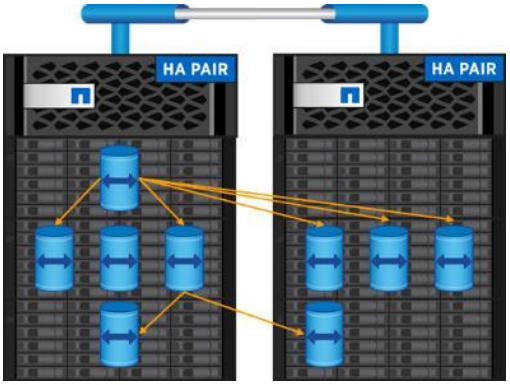
Figure 56) Scratch space workloads using qtrees with FlexGroup volumes.



Data lifecycle management solution: Volume-based project storage

ONTAP gives storage administrators the ability to present storage containers as folders to applications and projects in the form of FlexVol volumes.

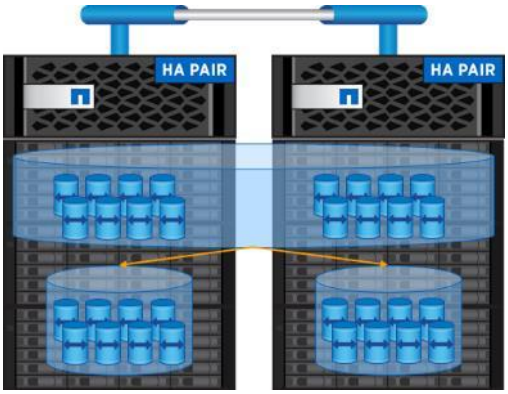
Figure 57) Volume-based multitenancy using junctioned FlexVol volumes.



This approach offers several benefits that traditional qtrees and directories cannot provide, including the ability to nondisruptively migrate data for individual projects based on performance or capacity needs. Storage administrators can also replicate individual projects by using SnapMirror or SnapVault software, rather than needing to replicate multiple projects (by using qtrees) to a destination site.

Additionally, you can create FlexClone volumes to create test/dev/scratch space scenarios for volumes, whereas with qtrees, you cannot. With the introduction of aggregate inline deduplication in ONTAP 9.2, this setup also allows storage efficiencies across multiple volumes in an aggregate.

Figure 58) Volume-based multitenancy using junctioned FlexGroup volumes.



You can also use junctioned volume architectures with FlexGroup volumes for many of the same reasons – granular data management and replication that volumes with qtrees cannot provide.

FlexGroup volumes are mounted to other volumes in the namespace, just like FlexVol volumes. The main caveat is total volume count. Each FlexGroup volume uses more overall FlexVol volumes in the system than a single FlexVol volume would.

Table 29) Pros and cons for volumes compared to qtrees for project storage.

	Pros	Cons
Using junctioned volumes for project-based storage	• Data mobility by way of volume moves. • Performance benefits in the form of multiple volume affinities/CPU threads. – This is not a concern when using FlexGroup volumes. • Ability to apply export policies to CIFS, if desired. • Ability to spread data volumes across multiple nodes. – This is greatly simplified when using FlexGroup volumes. • Ability to create qtrees inside volumes and provide even	• Volume limits per node are much lower than qtree limits. (node-specific, but 1-2K per node, 12K per cluster for volumes)

	more security granularity by using export policy rules. • Ability to take Snapshot copies of individual volumes and projects. • Ability to create FlexClone volumes for dev/test scenarios.	
Using qtrees for project-based storage	• Ability to create many more qtrees than volumes in a cluster (up to 4,995 qtrees per volume). • Ability to apply granular security at the qtree level through permissions and export policy rules. • Ability to apply monitoring and enforcement quotas. • Ability to apply granular QoS policies to qtrees (ONTAP 9.8 and later).	• Volume moves migrate entire directory structure; no qtree-based moves. • No granular Snapshot copies. • Cannot spread data across nodes when using qtrees (with FlexVol volumes). – FlexGroup volumes' qtrees will spread data across nodes when necessary to balance capacity. • No CIFS export policy support. • NFSv4 export policy support only in 8.3 and later.

Data lifecycle management solution: NetApp FabricPool

One issue with data lifecycle management is that it's very labor intensive, even with NetApp features such as volume move, qtrees, SnapMirror, and so on, and automation with things such as the NetApp Ansible modules.

One way to get yourself—and your end users—out of the business of managing individual volumes and qtrees is to use NetApp FabricPool to automatically tier cold data to capacity tiers in the cloud or on-premises (such as with NetApp StorageGRID or ONTAP S3) and automatically re-populate hot data tiers when datasets are accessed. For more information about NetApp FabricPool, see TR-4598: FabricPool Best Practices.

Security and ACL style considerations

In ONTAP, you can access the same data through NFS and SMB/CIFS by using multiprotocol NAS access. The same general guidance that applies to a FlexVol volume applies to a FlexGroup volume. That guidance is covered in the product documentation in the CIFS, NFS, and Multiprotocol Express Guides and the CIFS and NFS Reference Guides, which can be found with the product documentation for the specific ONTAP version being used. For multiprotocol NAS best practices, see TR-4887: Multiprotocol NAS in NetApp ONTAP.

In general, for multiprotocol access, you need:

- Valid users (Windows and UNIX)
- Valid name-mapping rules or 1:1 name mappings through local files and/or servers such as LDAP or NIS; ONTAP uses name mappings to coordinate access for clients
- Volume security style (NTFS, UNIX, or mixed); this can be configured for volumes or qtrees
- A default UNIX user (pcuser, created by default for Windows to UNIX name mappings); default Windows users (for UNIX to Windows name mappings) are not configured by default

When a volume is created, a security style is applied. If you create a volume without specifying a security style, the volume inherits the security style of the SVM root volume. The volume security style determines

the style of ACL that is used for a NAS volume and affects how users are authenticated and mapped into the SVM. When a FlexGroup volume has a security style selected, all member volumes will have the same security style settings.

Note: You can specify unique security styles in a single FlexVol or FlexGroup volume by using qtrees.

Basic volume security style guidance

The following is some general guidance on selecting a security style for volumes:

- With UNIX security style, Windows users must map to valid UNIX users. UNIX users only need to map to a valid user name if NFSv4.x is being used.
- In NTFS security style, Windows users must map to valid UNIX users, and UNIX users must map to valid Windows users to authenticate. If a valid UNIX user name exists, NFS clients see proper ownership on files and folders. Authorization (permissions) is processed by the Windows client after the initial authentication. If no valid UNIX user exists, the default UNIX user (pcuser) is used for authentication/UNIX ownership.
- UNIX security style allows some Windows clients to modify basic mode bit permissions (ownership changes, rwx), but does not allow NFSv4.x ACL management over SMB and does not understand advanced NTFS permissions.
- A mixed security style allows permissions to be changed from any type of client. However, it has an underlying effective security style of NTFS or UNIX, based on the last client type to change ACLs.
- A mixed security style requires proper name mapping to function properly due to the changing effective security styles.
- If granularity of ACL styles in a FlexGroup volume is desired, consider deploying qtrees, which are available starting in ONTAP 9.3. Qtrees enable you to set security styles per logical directory in ONTAP. If you want other home directory features such as NetApp FPolicy, antivirus, native file auditing, and quota enforcement, then use the most recent patched release of ONTAP 9.5 or later.
- NFSv4.x and NFSv4 ACL support for FlexGroup volumes is available in ONTAP 9.7.

Best practice 16: Volume security style recommendation

NetApp recommends a mixed security style only if clients need to be able to change permissions from both styles of clients. Otherwise, it is best to select either NTFS or UNIX as the security style, even in multiprotocol NAS environments.

More information about user mapping, name service best practices, and so on, can be found in the product documentation. You can also find more information in TR-4835: How to Configure LDAP in ONTAP, TR-4067: NFS Best Practice and Implementation Guide, and TR-4668: Name Services Best Practices Guide.

NFS considerations

In most cases, EDA workloads run on NFS, and predominantly on NFSv3 due its statelessness, which plays well in performance-driven workloads. This section covers NFS best practices and considerations as they pertain to EDA workloads.

NFS version considerations

When a client using NFS attempts to mount a volume in ONTAP without specifying the NFS version (that is, `-o nfsvers=3`), a protocol version negotiation takes place between the client and the server. The client asks for the highest versions of NFS supported by the server. If the server (in ONTAP, an SVM serving NFS) has NFSv4.x enabled, the client attempts to mount with that version.

However, because FlexGroup volumes do not currently support NFSv4.x, the mount request fails. This error usually manifests as “access denied,” which can mask the actual issue in the environment:

```
# mount demo:/flexgroup /flexgroup
mount.nfs: access denied by server while mounting demo:/flexgroup
```

To avoid issues with mounting a FlexGroup volume in environments where NFSv4.x is enabled, either configure clients to use a default mount version of NFSv3 by using `fstab` or specify the NFS version when mounting.

For example:

```
# mount -o nfsvers=3 demo:/flexgroup /flexgroup
# mount | grep flexgroup
demo:/flexgroup on /flexgroup type nfs (rw,nfsvers=3,addr=x.x.x.x)
```

Additionally, if a FlexGroup volume is junctioned to a parent volume that is mounted to a client through NFSv4.x, traversing to the FlexGroup volume fails because no NFSv4.x operations are currently allowed in FlexGroup volumes.

For example, FlexGroup volumes are always mounted to the vsroot (vserver root), which operates as (/) in the NFS export path. If a client mounts vsroot with NFSv4.x, and then attempts to access a FlexGroup volume from the NFSv4.x, the mount will fail. This includes `ls -la` operations, which require the ability to do NFSv4.x GETATTR operations.

In the following example, note that the information for the FlexGroup volumes is incorrect:

```
# mount demo:/ /mnt
# mount | grep mnt
demo:/ on /mnt type nfs (rw,vers=4,addr=10.193.67.237,clientaddr=10.193.67.211)
# cd /mnt/flexgroup
-bash: cd: /mnt/flexgroup: Permission denied
# ls -la
ls: cannot access flexgroup_4: Permission denied
ls: cannot access flexgroup_local: Permission denied
ls: cannot access flexgroup_8: Permission denied
ls: cannot access flexgroup_16: Permission denied
drwx--x--x. 12 root root   4096 Mar 30 21:47 .
dr-xr-xr-x. 36 root root   4096 Apr  7 10:30 ..
d??????????  ? ?    ?         ?            ? flexgroup_16
d??????????  ? ?    ?         ?            ? flexgroup_4
d??????????  ? ?    ?         ?            ? flexgroup_8
d??????????  ? ?    ?         ?            ? flexgroup_local
```

Compare that to the NFSv3 mount:

```
# ls -la
drwx--x--x. 12 root root 4096 Mar 30 21:47 .
dr-xr-xr-x. 36 root root 4096 Apr  7 10:30 ..
drwxr-xr-x.  6 root root 4096 May  9 15:56 flexgroup_16
drwxr-xr-x.  5 root root 4096 Mar 30 21:42 flexgroup_4
drwxr-xr-x.  6 root root 4096 May  8 12:11 flexgroup_8
drwxr-xr-x. 14 root root 4096 May  8 12:11 flexgroup_local
```

Therefore, be sure not to use NFSv4.x in any path where a FlexGroup volume resides.

Best practice 17: NFS version considerations for EDA environments

The primary protocol for EDA simulations is NFSv3. If NFSv4.x and its features are required, do not use NFSv4.0; instead, use NFSv4.1 or later. If you wish to use FlexGroup volumes with NFSv4.x, use ONTAP 9.8 or later.

Note: Whenever using NFSv4.x, plan to use the latest software releases for the NFS client and ONTAP.

General NAS networking considerations for ONTAP

A NAS connection in ONTAP follows these general guidelines:

- When a client connects to a data LIF in ONTAP, a unique ID is assigned to that TCP connection to keep track of the operations.
- If the same NAS client makes subsequent NAS connection requests to the same data LIF, that unique ID is re-used, even if a different data volume is accessed.
- If the same NAS client makes a subsequent NAS connection request to a *different* data LIF (even if it is on the same node), then a new unique ID is assigned for that client.
- Each node in a cluster has a finite limit of possible TCP connections for NAS operations. Once this limit is reached, new connections are denied until resources are freed up. The ONTAP EMS logs will show errors if this happens.
- Using a single data LIF in a cluster provides only one access point for incoming NAS connections and adheres to the TCP connection limits on the single node. As a result, if you mount 1,000 clients to a single node, you run the risk of exhausting system resources faster than if you mount 1,000 clients to four different cluster nodes. Avoid pointing all NAS clients to a single data LIF in large scale environments - such as EDA – to mitigate resource exhaustion.
- In an HA pair, if each node has more than half of the maximum CIDs allowed and a storage failover occurs then you may hit the CID limit on the surviving node as clients attempt to re-establish connectivity to the NFS mounts and connections will fail. If possible, try to keep NAS connections around 50% of the total allowed limit in case a failover occurs,
- NFSv3 mounts have ancillary protocols that are used during the connection process, such as mount and portmapper. These UDP-based protocols also get assigned unique IDs during the initial connection process and will age out after 60 seconds. In mount storm scenarios (thousands of clients mounting at the same time), it may be possible to exhaust connection resources artificially and prematurely due to the temporary ID assignments for the ancillary protocols. Consider staggering client mounts and spreading your automounter workloads across multiple nodes in a cluster, or, if using Cloud Volumes ONTAP, multiple CVO instances that utilize FlexCache volumes.
- Unmounts also will generate CIDs for the mount/portmap protocols.
- Both UDP and TCP connections count against the total network connections on a node.
- Specifying TCP as a mount option eliminates those extra UDP calls and reduces the total number of connections generated on each mount/unmount. In environments that generate a lot of mounts/unmounts, use TCP as a mount option.
- NFSv4.x always uses TCP and does not use mount/portmapper for mounts/unmounts, so no extra CIDs will be generated for that NFS version.
- When the maximum connection limits are exceeded, an EMS will be generated (maxCID.threshold.exceeded or maxCID.limit.exceeded).
- Modern NAS clients have features that attempt to add more parallel network connections to a single NAS connection, such as SMB multichannel and NFS nconnect. When using features that provide more TCP threads per NAS share, it generally improves performance, but they will also use up more unique IDs in the NAS stack of ONTAP. For example, a normal NFSv3 mount may only use a single unique ID, but an NFSv3 mount using nconnect=4 may use up to 4 unique IDs per mount. Keep this in mind when designing for scale in EDA environments.
- Each unique ID in ONTAP has a limit of 128 concurrent NAS operations. If this limit is exceeded by the client sending more concurrent operations than ONTAP is able to handle, a form of flow control is enacted on the NAS stack in ONTAP until resources are freed up. You can mitigate this behavior with client-side configuration or by using more unique IDs per client. See “Network connection concurrency and TCP slots: NFSv3” for more information on network concurrency with NAS.
- In addition to per-node CID limits and per-connection concurrent NAS operation limits, there are also node-level limits for total available NAS operations at any given time. Each time a NAS operation is performed, a resource context is reserved until that operation is completed. At that time, the resource is released back to the system. If too many resources are requested at once on a single node, then performance issues can be seen. See “Exec context throttling” for information on these resources and how to best prevent issues.

Examples of mount/unmount connection behavior

The following shows examples of how CIDs get generated with NFSv3 mount/unmounts with and without the `tcp mount` option. These CIDs can be seen with the command `network connections active show -remote-ip x.x.x.x.`

Best practice 18: Use the `tcp mount` option for NFSv3 mounts to minimize total connections

To reduce the total number of connections for NFSv3 mounts and reduce the chances of exceeding the node's connection limits, specify the `tcp mount` option.

CIDs generated with mounts/unmounts not using `tcp mount` option

Before mount:

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

Client mounts without TCP option:

```
# mount -o vers=3 DEMO:/home /mnt/client1
```

After mount:

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: cluster-02
DEMO       data2:111              centos83-perf.ntap.local:56131
                                                               UDP/port-map
DEMO       data2:635              centos83-perf.ntap.local:44961
                                                               UDP/mount
DEMO       data2:635              centos83-perf.ntap.local:1022
                                                               UDP/mount
DEMO       data2:2049             centos83-perf.ntap.local:879 TCP/nfs
4 entries were displayed.
```

After approximately 60 seconds, the UDP CIDs disappear and only the NFS connection remains:

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: cluster-02
DEMO       data2:2049             centos83-perf.ntap.local:879 TCP/nfs
```

Client unmounts:

```
# umount /mnt/client1
```

NFS CID is gone, but now we have mount/portmap UDP connections for approximately 60 seconds:

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: cluster-02
DEMO       data2:111              centos83-perf.ntap.local:36775
                                                               UDP/port-map
DEMO       data2:635              centos83-perf.ntap.local:33867
                                                               UDP/mount
DEMO       data2:635              centos83-perf.ntap.local:966 UDP/mount
3 entries were displayed.

cluster::> network connections active show -remote-ip x.x.x.x
```

```
There are no entries matching your query.
```

CIDs generated with mounts/unmounts with the TCP mount option

Before mount:

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

Client mounts with the TCP option:

```
# mount -o vers=3,tcp DEMO:/home /mnt/client1
```

After mount:

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                   Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: cluster-02
DEMO        data2:2049             centos83-perf.ntap.local:931 TCP/nfs
```

Client unmounts:

```
# umount /mnt/client1
```

NFS CID is gone and we have no UDP mount/portmapper:

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

Behavior of NFS connections with containers

Containerized environments (such as Docker or RedHat OpenShift) are becoming more prevalent in EDA environments. As a result, it's important to understand how containers impact NFS connections in ONTAP to ensure proper sizing and scale-out considerations when implementing containers using NFS mounts.

If there are NFS mounts on the container host, they will have their own CIDs per data LIF. Containers on that host using NFS mounts will often share the same network IP address as the host, but when initiating an NFS mount, it will get its own CID.

For example, this container host has a mount to a volume:

```
[root@centos7-docker ~]# mount | grep scripts
demo:/scripts/dockerfiles on /dockerfiles type nfs
(rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec
=sys,mountaddr=x.x.x.x,mountvers=3,mountport=635,mountproto=udp,local_lock=none,addr=x.x.x.x)

cluster::> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node               cid         vserver local-address remote-ip     proto service
------------------ ----------- ------- ------------- ------------- ----- -------
clutster-01        1011516163 DEMO    x.x.x.x       x.x.x.y       TCP   nfs

cluster::> nfs connected-clients show -node * -client-ip x.x.x.y -data-lif-ip x.x.x.x

     Node: cluster-01
  Vserver: DEMO
  Data-Ip: x.x.x.x
Client-Ip       Volume-Name      Protocol Idle-Time     Local-Reqs Remote-Reqs
--------------- ---------------- -------- ------------- ---------- -----------
x.x.x.y         scripts          nfs3     55s           73         0
```

If a container mount is initiated to the same data LIF, a new CID is generated for NFS, along with new UDP CIDs for mount and port-map:

```
[root@centos7-docker ~]# docker exec -it centos bash
[root@f8cac0b471dc /]# mount -o vers=3 10.193.67.237:/scripts /mnt

cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node               cid        vserver local-address remote-ip     proto service
------------------ ---------- ------- ------------- ------------- ----- -------
cluster-01         1011516163 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01         1011516166 DEMO    x.x.x.x       x.x.x.y UDP   port-map
cluster-01         1011516167 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01         1011516168 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01         1011516170 DEMO    x.x.x.x       x.x.x.y TCP   nfs
```

When we start a new container and mount within it, an additional NFS CID and UDP CIDs are created from the same client IP, even though we are using the same data LIF, volume and node:

```
[root@centos7-docker ~]# docker run --name centos2 --rm -it --cap-add SYS_ADMIN -d
parisi/centos7-secure
16dec486692dbc1133b4c1f74c6e78aa7aab875c0aed71d0d087461a6bed8060
[root@centos7-docker ~]# docker exec -it centos2 bash
[root@16dec486692d /]# mount -o vers=3 10.193.67.237:/scripts /mnt

cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node               cid        vserver local-address remote-ip     proto service
------------------ ---------- ------- ------------- ------------- ----- --------
cluster-01         1011516163 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01         1011516170 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01         1011516177 DEMO    x.x.x.x       x.x.x.y UDP   port-map
cluster-01         1011516178 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01         1011516179 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01         1011516183 DEMO    x.x.x.x       x.x.x.y TCP   nfs
```

As a result, CIDs can begin to add up quickly in environments that generate a lot of NFS mounts in a short period of time – particularly if containers are involved.

For an idea of how scaling container environments can create potential impact on the NFS server, consider the following scenario: if a container host starts 1,000 containers and each container makes an NFS mount request to the same data LIF in a cluster, then the total number of CIDs that will get generated on that single data LIF will range between 1,000 (if specifying the `tcp mount` option) and 4,000 (3,000 UDP CIDs for mount/portmap that will age out after approximately 60 seconds if the `tcp mount` option is not specified).

If those containers mount to different data LIFs on the same node, the same CID dispersal will apply.

If those containers mount to different data LIFs on different nodes, then the CID will be distributed across nodes and it will take longer for the CID limits to be reached. The more cluster nodes and data LIFs used, the more CID dispersal will take place. For 2 nodes, 1,000 CIDs can divide into 500 per node. For a 4-node cluster, 1,000 CIDs can divide into 250 per node.

Impact of nconnect on total connections

When an NFS mount is established, a single NFS CID is used. However, with the new NFS option nconnect, it is possible to open multiple TCP connections per NFS mount. For instance, using nconnect=8 with your NFS mount option creates as many as eight NFS CIDs to the data LIF in the SVM. While this can deliver extra performance benefits, it can also use up more CIDs than expected.

For example:

```
# mount -o nconnect=8 DEMO:/home /mnt/client1

cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node               cid        vserver local-address remote-ip     proto service
------------------ ---------- ------- ------------- ------------- ----- -------
```

```
cluster-02          2328843073 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843076 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843077 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843078 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843079 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843080 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843081 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-02          2328843082 DEMO    x.x.x.y       x.x.x.x TCP   nfs
```

If you plan on using nconnect in environments that generate a large number of mounts, be mindful of the CID limits per node for your platform and plan to distribute connections across multiple cluster nodes to balance connections appropriately. In cases where a single node/data LIF is being used for NFS mounts, CIDs will run out much faster than if multiple nodes are used. CID limits should be factored into scale discussions when architecting the solutions.

Impact of NetApp XCP on total connections

When using NetApp XCP for data migrations, it behaves similarly to nconnect when establishing CIDs. While a single mount point may be specified, the `-parallel` option determines the total number of CIDs established to the ONTAP node. If the `-parallel` option is not defined, XCP will default to 8 parallel connections.

Note: With XCP scans, only a single CID is used.

In the following example, an XCP host was used to copy data from an NFS mount with the following syntax:

```
# xcp copy -parallel 16 IP1,IP2:/files destination:/files
```

In the above command, XCP is creating 16 network threads per IP address specified. In this case, both IP1 and IP2 are on the same node of the source cluster. As a result, that node gets 34 total CIDs established for this job (18 on the first IP specified; 16 on the second IP):

```
cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node                cid        vserver local-address remote-ip     proto service
------------------- ---------- ------- ------------- ------------- ----- -------
cluster-01          1011516323 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516327 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516328 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516329 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516330 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516331 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516332 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516333 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516334 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516335 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516336 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516337 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516338 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516339 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516340 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516342 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516343 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516344 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516345 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516346 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516347 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516348 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516349 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516350 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516351 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516352 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516353 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516354 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01          1011516355 DEMO    x.x.x.x       x.x.x.y TCP   nfs
```

```
cluster-01          1011516356 DEMO    x.x.x.z        x.x.x.y TCP   nfs
cluster-01          1011516357 DEMO    x.x.x.z        x.x.x.y TCP   nfs
cluster-01          1011516358 DEMO    x.x.x.z        x.x.x.y TCP   nfs
cluster-01          1011516359 DEMO    x.x.x.x        x.x.x.y TCP   nfs
cluster-01          1011516360 DEMO    x.x.x.z        x.x.x.y TCP   nfs
34 entries were displayed.
```

When using XCP, try to spread the connections across multiple IP addresses on multiple nodes and understand that the number of parallel threads means more total network CIDs will be in use.

Impact of SMB multichannel on total connections

Most EDA environments do not use SMB for workloads, but in environments that do leverage SMB, CIDs still are used for NAS connectivity.

Unlike NFSv3, SMB does not make use of UDP for connections – instead, TCP is always used, so CIDs are not kept around for transient operations such as tree connect/disconnect. However, ONTAP does provide support for the SMB multiplexing feature called multichannel. This operates in similar fashion to nconnect and XCP – multiple network CIDs are generated for a single SMB/CIFS share.

In the following example, an SMB share connection is established from a Windows 2019 client that supports SMB multichannel. We can see that multichannel is being used with the command `cifs session show`:

```
cluster::*> cifs session show -address x.x.x.x

Node:    cluster-02
Vserver: DEMO
Connection Session                                      Open         Idle       Connection
ID         ID      Workstation      Windows User       Files        Time            Count
---------- ------- ---------------- ---------------- --------- ------------ ---------------
2328843549 6491094437924438318                                        3s                4
                   x.x.x.x          NTAP\                    2
                                    Administrator
```

In the above, we can see connection count is four, which means the SMB share connected with SMB multichannel. `network connections active show` verifies that we have four active CIDs for the service `cifs-srv`:

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields service
node                cid        vserver service
------------------ ---------- ------- --------
cluster-02          2328843549 DEMO    cifs-srv
cluster-02          2328843550 DEMO    cifs-srv
cluster-02          2328843551 DEMO    cifs-srv
cluster-02          2328843552 DEMO    cifs-srv
4 entries were displayed.
```

These CIDs will remain as long as the SMB/CIFS share is in use. Once the share is closed, the CIDs will be released back to ONTAP for use with new NAS connections. Every SMB client that supports SMB multichannel connects to data LIFs on this node will use four CIDs, so incorporate this knowledge into your NAS architecture.

For more information on SMB multichannel, see the following resources:

- Manage SMB Multichannel – Azure
- Deploy SMB Multichannel

Impact of Robocopy multithreading on CIDs

Similar to XCP, Robocopy can be used to create multithreaded data copies over SMB. However, unlike XCP, Robocopy does not create its own SMB server, but instead uses the SMB functionality of the SMB client. With Robocopy, the multithreading is via CPU threads. The SMB share being used will still

leverage the same network CIDs as a normal share operation, such as what you would see with SMB multichannel.

For example, this robocopy job uses `/MT:16` to perform the data copy from one SMB share to another. In this case, both shares are in the same cluster and SVM.

```
PS C:\> robocopy /MT:16 \\demo\files \\demo\flexgroup\files
```

The CIDs used for this job were only the connections used for SMB multichannel.

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields service
node               cid        vserver service
------------------ ---------- ------- --------
cluster-02         2328843568 DEMO    cifs-srv
cluster-02         2328843569 DEMO    cifs-srv
cluster-02         2328843570 DEMO    cifs-srv
cluster-02         2328843571 DEMO    cifs-srv
4 entries were displayed.
```

Viewing CID maximums and allocations

In ONTAP, it is possible to view how many CIDs are available for a node, as well as how many CIDs are currently in use and which clients are using those connections.

To view CID information, keep the following commands in mind:

network connections active (admin privilege)

```
cluster::> network connections active ?
  delete                      *Delete an active connection in this cluster
  show                        Show the active connections in this cluster
  show-clients                Show a count of the active connections by client
  show-lifs                   Show a count of the active connections by logical interface
  show-protocols              Show a count of the active connections by protocol
  show-services               Show a count of the active connections by service
```

statistics start/show -object cid (diag privilege)

```
cluster::*> statistics start -object cid
```

node run netstat -na (diag privilege)

```
cluster::*> node run -node node1 netstat -na
```

Example of CID behavior and maximums using statistics

To gather statistics for this object, run the following command in diag privilege:

```
cluster::*> statistics start -object cid
```

Statistics collection is being started for sample-id: `sample_55`.

To view the output, run the following command:

```
cluster::*> statistics show -object cid
```

The following is a sample of how those results might appear before an NFS mount is established:

```
Object: cid
Instance: cid
Start-time: 6/25/2021 16:54:27
End-time: 6/25/2021 16:56:37
Elapsed-time: 130s
Scope: cluster-01
```

```
    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                     0
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        352
    in_use_max                                                      0
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

In the above example, the node has a `cid_max` of 115,904. Of those, 10,526 are `reserved_cid` for ONTAP system operations, which means the total number of available CIDs for client operations would be 115,904 – 10,526 = 105,378. If a node exceeds that limit, EMS will trigger a `maxCID.limit.exceeded` message.

Currently, there are 352 `in_use` CIDs.

After an NFS mount with nconnect=8 is established on node1, this is how those numbers change.

```
Object: cid
Instance: cid
Start-time: 6/25/2021 16:54:27
End-time: 6/25/2021 17:04:54
Elapsed-time: 627s
Scope: cluster-01

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                    14
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        360
    in_use_max                                                      0
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

Now, instead of 352 CIDs `in_use`, we used 360. This aligns with the eight active network connections created with the nconnect mount.

When an NFSv3 mount is issued without the `tcp mount` option specified, four new CIDs will be `in_use` and we'll see it in the statistics.

```
    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                    28
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        364
    in_use_max                                                      1
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

These CIDs include the mount and port-map UDP entries seen in network connections active show.

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node                cid        vserver local-address remote-ip     proto service
------------------- ---------- ------- ------------- ------------- ----- -------
cluster-01          1011516253 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516256 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516257 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516258 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516259 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516260 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516261 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516262 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516268 DEMO    x.x.x.z       x.x.x.x UDP   port-map
cluster-01          1011516269 DEMO    x.x.x.z       x.x.x.x UDP   mount
cluster-01          1011516270 DEMO    x.x.x.z       x.x.x.x UDP   mount
cluster-01          1011516272 DEMO    x.x.x.z       x.x.x.x TCP   nfs
```

Once those UDP entries age out (around ~60 seconds), the CIDs are released for use by new connections and we only have one extra CID `in_use` by the NFS connection on data LIF IP **x.x.x.z**.

```
    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                    28
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        361
    in_use_max                                                      1
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526

cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node                cid        vserver local-address remote-ip     proto service
------------------- ---------- ------- ------------- ------------- ----- -------
cluster-01          1011516253 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516256 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516257 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516258 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516259 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516260 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516261 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516262 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01          1011516272 DEMO    x.x.x.z       x.x.x.x TCP   nfs
```

The rootonly options: nfsrootonly and mountrootonly

The `rootonly` options are added to help prevent untrusted client access. Untrusted clients (those not part of the export rules) can potentially access data by using SSH tunneling to trusted clients. However, those requests would come from untrusted ports (ports greater than 1,024). This can provide a back door for clients not intended to have access.

Therefore, the enabling or disabling of the `rootonly` options hinges upon need. Does the environment require more ports to allow NFS to function properly? Or is it more important to prevent untrusted clients from accessing mounts?

One potential compromise is to make use of NFSv4.x and/or Kerberos authentication for a higher level of secured access to NFS exports. TR-4616: NFS Kerberos in ONTAP covers how to use NFS Kerberos.

In these scenarios, using the `mount-rootonly` and/or `nfs-rootonly` options can alleviate these issues.

To check port usage on the client, run the following command:

```
# netstat -na | grep [IP address]
```

To check port usage on the cluster, run the following command:

```
cluster::> network connections active show -node [nodename] -vserver [vservername] -service nfs*
```

For example, we can specify that client should use a port outside of the reserved port range with the mount option `noresvport` (`resvport` is the default if not specified and uses source ports between 1–1,024). When we do that and `mount-rootonly` is enabled for the NFS SVM, the mount fails.

```
cluster::*> nfs show -vserver DEMO -fields mount-rootonly,nfs-rootonly
vserver mount-rootonly nfs-rootonly
------- -------------- ------------
DEMO    enabled        disabled

# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
mount.nfs: access denied by server while mounting demo:/scripts
```

From a packet trace, we can see that the client's source port is outside of the allowed range (36643).

```
129     x.x.x.x        x.x.x.y        MOUNT   138     V3 MNT Call (Reply In 130) /scripts
User Datagram Protocol, Src Port: 36643, Dst Port: 635
```

When we use resvport, the mount succeeds:

```
# mount -o resvport,vers=3 demo:/scripts /mnt/client1
#
```

A packet trace shows that the source port is within the 1,024 range (703).

```
44      x.x.x.x        x.x.x.y        MOUNT   138     V3 MNT Call (Reply In 45) /scripts
User Datagram Protocol, Src Port: 703, Dst Port: 635
45      x.x.x.y        x.x.x.x        MOUNT   150     V3 MNT Reply (Call In 44)
```

When the `mount-rootonly` option is set to disabled:

```
cluster::*> nfs modify -vserver DEMO -mount-rootonly disabled
```

The mount using noresvport succeeds:

```
# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
#
```

And the trace shows the source port is outside of the 1,024 range (58323).

```
101     x.x.x.x        x.x.x.y        MOUNT   138     V3 MNT Call (Reply In 102) /scripts
User Datagram Protocol, Src Port: 58323, Dst Port: 635
102     x.x.x.y        x.x.x.x        MOUNT   150     V3 MNT Reply (Call In 101)
```

Note: Since NFSv4.x mounts don't use the ancillary mount protocols for NFS mounts, the `mount-rootonly` port does not factor in to those operations and only will impact NFSv3 mounts.

Mount port exhaustion with a large number of NFS clients

In environments with a large number of clients connecting through NFS, it is important to keep in mind that, by default, the number of mount ports are limited to 1,024. In a mount storm scenario (such as thousands of clients mounting or unmounting at around the same time), 1,024 ports could be exhausted quickly and create connectivity issues.

- Mount operations (only applicable to NFSv3) are limited through the NFS option `mount-rootonly`, which is set to **enabled** by default. This limits the incoming port range for mounts to 1–1,024.

- For NFS operations over port 2049, the default number of incoming ports allowed is 65,534. This is controlled by the `nfs-rootonly` NFS option and is set to **disabled** by default. To limit the number of incoming NFS client source ports to 1,024, set this option to **enabled**.

In some circumstances, the number of ports used to mount or for NFS operations might be exhausted, which then causes subsequent mount and NFS operations to hang or fail until a port is made available.

If an environment has thousands of clients that are mounted through NFS and generating I/O (such as the container example in "Behavior of NFS connections with containers"), it is possible to exhaust all ports on an NFS server. For example, one scenario seen was with ESX using NFS datastores, because some legacy best practices would call for a data LIF/IP address per datastore. In environments with many volumes/datastores, this created a situation where the NFS ports were overrun. The remediation for this situation was to disable the `mount-rootonly` and/or the `nfs-rootonly` options on the NFS server. Performing that action removed the 1 to 1,024 port range limit and allowed up to 65,534 ports to be used in a NFS server for mounts and NFS operations. On the client side, you may also need to use the mount option `noresvport` to use nonprivileged source ports, as NFS mounts default to the `resvport` mount option when not specified.

Mount port usage with UDP versus TCP

By default, NFSv3 mounts use UDP for the MOUNT protocol. Because UDP has no acknowledgements, ONTAP will maintain the CID for UDP mounts for 60 seconds to ensure the mount has a chance to succeed or fail before ONTAP will remove the CID. This also applies to unmount operations.

Because of this, a large number of mount/unmount requests at the same time can use up all of the available source ports in the 1–1,024 range until the 60-second expiration is reached and the existing mount CIDs are cleared. Even unsuccessful mounts using UDP will maintain the CIDs for 60 seconds.

For example, this mount using a port outside of the allowed range failed.

```
# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
mount.nfs: access denied by server while mounting demo:/scripts
```

From ONTAP, we can see there are three CIDs generated from that failed mount request that will expire after 60 seconds. In a mount storm, 60 seconds can be an eternity.

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields remote-port,cid,service
node               cid        vserver remote-port service
------------------ ---------- ------- ----------- --------
cluster-02         2328843541 DEMO    35470       port-map
cluster-02         2328843542 DEMO    60414       mount
cluster-02         2328843543 DEMO    33649       mount
3 entries were displayed.
```

TCP uses acknowledgements in its conversations, so ONTAP does not need to maintain CIDs after the mount request is successful. As a result, using `-o tcp` for your NFS mount option means that ports in the 1–1,024 range will be freed up faster and will prevent most cases where port exhaustion for mounts may occur.

Best practices for environments with a large number of NFS clients

The following section covers some of the best practices for environments with a large number of NFS clients, such as EDA compute farms.

Best practice 19: Considerations for environments with mount storms

A mount storm is where a large number of NFS clients mount or unmount NFS exports to the same NFS server or cluster in a short period of time. NFS environments that heavily utilize automounters can experience mount storms without administrators being aware.

In a mount storm scenario, system resources can be used up quickly. The following best practices should be considered when designing an environment where it's possible to encounter mount storms.

- More cluster nodes and data LIFs on each node means there are more available system resources to use to balance out NFS mounts, connections, etc. When designing for large scale NFS mount environments, consider using as many nodes as possible for incoming NFS connections. ONTAP supports up to 24 nodes for NAS-only clusters.
- When using multiple data LIFs in a cluster for load balancing of connections, use a DNS load balancer to simplify the environment by masking many IP addresses behind a single hostname. This allows easy addition/removal of IP addresses to the DNS FQDN without needing to notify end users of a change in connectivity and helps maintain an even balance of incoming NAS connections from clients whether automounters are used or not.
- Monitor data LIF locations and storage failovers closely. If a data LIF migrates to another node (due to port failure or manual migration), then that data LIF will be using the current node's resources for NAS connections and could contribute to faster resource exhaustion. The same is true for storage failovers (planned or unplanned), as the failed node's data LIF will migrate to the surviving partner and now that node will need to maintain all of the NAS connections. If a data LIF migration occurs, resolve the issue that caused the migration and revert the data LIF back to its home node.
- Monitor CID usage in the cluster using `statistics start -object cid` or the equivalent REST API functions. Try to keep CID `in_use` values around 50% of the `max_cid` values for the node in case of storage failover (roughly 50,000 `in_use` CIDs per node). This can be accomplished by adding more nodes with new data LIFs to the cluster.
- Monitor EMS for the events `maxCID.limit.exceeded`, `maxCID.threshold.exceeded`, `nblade.execsOverLimit` and contact NetApp technical support if these events are generated.
- Multiplexing of NFS connections by way of more data LIFs per node or via nconnect can increase performance for some workloads, but it can also use more available resources per storage node (such as CIDs and exec contexts). If you use multiplexing, be aware of the potential side effects (as covered in "Impact of nconnect on total connections" and "Exec context throttling").
- If possible, use the `tcp mount` option for NFS mounts (`-o tcp`) – this reduces the total number of CIDs generated per mount/unmount operation and helps prevent resource exhaustion (CIDs and NFS mount ports) in mount storm scenarios.
- If you require more than 1,024 available incoming NFS mount ports (for instance, if 2,000 clients are mounting at the same time), you may need to consider disabling the NFS server option `mount-rootonly`, using the `noresvport` mount option on the NFS clients/automounters and using the mount option TCP to reduce the number of CIDs generated per mount. (UDP mount connections will remain in cache for up to 60 seconds; TCP mount connections will be removed upon client ACK).
- NFSv4.x is not susceptible to issues with UDP or the mount protocol, but has its own challenges as covered in "NFS version considerations."

NFSv4.x performance impact

NetApp is constantly striving to improve performance for each ONTAP release. NFSv4.x performance is a priority because it is the future for NFS.

The following performance enhancements are listed in this section, along with the ONTAP release in which they are introduced.

Note: As of ONTAP 9.8, NFSv4.1 performance for high metadata workloads is roughly 34% worse than NFSv3 at peak IOPS on an AFF A700 system (about 5% better than ONTAP 9.7). At lower loads, the performance impact is less, with NFSv3 and NFSv4.1 achieving similar IOPS, with NFSv4.1 experiencing roughly 1ms higher latency than NFSv3. For the best possible performance with NFS, always run the latest patched ONTAP release available.

NFSv4.x fastpath (introduced in ONTAP 8.2)

Starting in Data ONTAP 8.2, NFS fastpath is available to potentially improve NFSv4 performance for READs and WRITES. This improvement is made by bypassing the internal processing of NFSv4 packets into clustered Data ONTAP-centric packets when the data request is made on a LIF that is local to the node hosting the volume. When combined with other features, such as pNFS or referrals, localized data can be guaranteed for each READ and WRITE request, thus allowing consistent use of the NFSv4 fastpath. NFSv3 has always had an NFS fastpath concept. NFS fastpath is enabled by default.

NFSv4.x multithreaded operations (introduced in ONTAP 8.2)

In Data ONTAP 8.2 and later, multiprocessor support is added for NFSv4.x read and write operations. Metadata operations, however, still use a single threaded approach. In earlier releases, NFSv4.x read and write operations are single threaded, thus allowing a potential bottleneck at the CPU for the protocol domain. Using multiple processors for read and write operations can greatly increase throughput on NetApp systems that contain more than one CPU for NFSv4.x workloads that are read and write heavy.

Note: NFSv3 has always used multiple processors for reads and writes. NFSv3 also uses multiple processors for metadata operations.

NFSv4.x performance improvements for streaming workload types (introduced in ONTAP 9.0)

ONTAP 9.0 introduces an improvement to how streaming workload types like VMware, Oracle and SAP HANA perform with NFSv4.1 by adding large I/O support. This enables NFS (both v3 and 4.x) to use up to 1MB for both reads and writes.

NFSv4.x performance improvements for metadata workloads (introduced in ONTAP 9.5)

Many improvements have been added to ONTAP 9.5 to improve metadata workloads, including:

- NFSv4.0 cache I/O support
- Optimizations of NFSv4.x metadata operations
- Increased caching
- Improved locking performance
- Increased storePool limits

NFSv4.x FlexGroup volume support (introduced in ONTAP 9.7)

In addition to the metadata performance improvements added in ONTAP 9.5, FlexGroup volume support can help increase metadata workload performance as well, by way of parallelization of file ingest operations. For more information about FlexGroup volumes, see TR-4571: NetApp FlexGroup Volumes Best Practices and Implementation.

NFSv4.x performance enhancements for metadata operations (ONTAP 9.8)

Some enhancements have been added to ONTAP 9.8 to improve the overall performance for NFSv4.x when dealing with high metadata workloads, such as what is seen with the SPEC SFS2014_swbuild benchmark. These changes have resulted in a marked improvement for these workloads, bringing the peak IOPS achieved at 1ms latency closer to what NFSv3 provides.

These enhancements include:

- Improved write lock usage
- Optimized QoS performance
- Improved handling of replay operations

- Consolidation of compound operations
- storePool optimizations
- OPEN/CLOSE enhancements
- Nconnect support (see “Nconnect” for details)
- Parallelization of UNLINK operations
- Parallelization of READDIR operations

NFSv4.x performance optimizations in ONTAP 9.9.1

Additional optimizations were added to ONTAP 9.9.1 to improve the overall performance for NFSv4.x when dealing with high metadata workloads, such as what would be seen with standard NAS benchmark tests for software builds. These changes resulted in a marked improvement over ONTAP 9.8 for these workloads, bringing the peak IOPS achieved at 1ms latency within ~22% of what NFSv3 offers:

- Prefetch of attributes during LOOKUP and CLOSE operations
- Prefetch of ACCESS calls during OPEN to reduce metadata operations
- GETATTR after WRITE optimizations
- Memory pool to store prefetched attributes
- FlexGroup optimizations (cache FlexGroup volume details to optimize LOOKUP)
- Path length reduction for export checks

Coexisting NFSv3 and NFSv4.x in the compute farm?

Newer versions of Linux support both NFSv4.x and NFSv3. However, some EDA workloads might include a mix of older clients that do not support both NFS versions and newer clients that do. Rather than taking maintenance windows to upgrade older clients to newer Linux releases, EDA workload environments can choose to use both NFSv3 and NFSv4.x in the same environment, on the same datasets.

ONTAP supports NFSv3, NFSv4.0, and NFSv4.1/pNFS, and you can use all three in the same storage VM concurrently for one or more exported file systems. Before leveraging newer NFS versions, check with the application vendor for their statement of support, as well as their recommended client operating system versions.

The benefits of having NFSv3 and NFSV4.1/pNFS protocols coexist in the compute farm are:

- No change is required to existing compute nodes that mount the file systems over NFSv3. There is no disruption to existing clients in the compute farm as more nodes on newer client versions are added to scale the number of jobs. The same file system can also be mounted over NFSv3 or NFSv4.1/pNFS from new pNFS-supported clients.
- NFSv4.1/pNFS can provide significant performance improvement in job completion times. Critical chip designs can be isolated from the rest faster job completion and better SLO.

Note: If you are using NetApp FlexGroup volumes, use ONTAP 9.8 or later.

NFSv3 vs. NFSv4.x: Performance Comparisons

To accurately understand the impact on performance between NFSv3 and NFSv4.x in your environment, testing is essential. Not all workloads will perform the same with each protocol, and NFSv4.x is substantially different from NFSv3 in a variety of ways – locking, statefulness, metadata handling, compound operations, to name a few. A workload in NFSv3 may have a much different profile than the same workload using NFSv4.x. For example, a high file create workload (1000 directories, 1 million files across those directories) using ONTAP 9.9.1 NFSv3 produced statistics that looked like this – roughly an even split of creates, lookups and writes.

```
Object: nfsv3
```



```
Instance: DEMO
    Counter                                                     Value
    -------------------------------- --------------------------------
    access_total                                                 1014
    create_percent                                                33%
    create_total                                              1000000
    fsinfo_total                                                    4
    getattr_total                                                   3
    lookup_percent                                                33%
    lookup_total                                              1000003
    mkdir_total                                                  1003
    null_total                                                      4
    pathconf_total                                                  2
    total_ops                                                   31936
    write_percent                                                 33%
    write_total                                               1000000
```

The same exact workload using NFSv4.1 produces a very different statistical profile. Since NFSv4.x operates much differently than NFSv3 for file creates, we see:

- Higher metadata (15%)
- Fewer CREATE operations (1000 vs. 1000000 with NFSv3)
- OPEN/CLOSE operations for each file creation
- Several other metadata operation types (COMPOUND, SEQUENCE, GETFH, PUTFH)
- Write totals stayed the same, but percentage wise, were much lower (7%)

```
Object: nfsv4_1
Instance: DEMO
    Counter                                                     Value
    -------------------------------- --------------------------------
    access_percent                                                7%
    access_total                                              1001014
    close_percent                                                 7%
    close_total                                               1000000
    compound_percent                                             23%
    compound_total                                            3002078
    create_session_total                                            2
    create_total                                                 1003
    exchange_id_total                                               8
    getattr_percent                                              15%
    getattr_total                                             2002064
    getfh_percent                                                 7%
    getfh_total                                               1001012
    lookup_total                                                    7
    open_percent                                                  7%
    open_total                                                1000000
    putfh_percent                                                23%
    putfh_total                                               3002064
    putrootfh_total                                                 2
    reclaim_complete_total                                          2
    sequence_percent                                             23%
    sequence_total                                            3002068
    total_ops                                                    6417
    write_percent                                                 7%
    write_total                                               1000000
```

As a result, just by using NFSv4.1 for this workload, there are more overall metadata operations, which NFSv4.1 tends to perform poorly with. That is illustrated in a side-by-side comparison of various performance metrics.

Table 30) NFSv3 vs. NFSv4.1 performance – High file creation workload.

Test	Average IOPS	Average MBps	Average Latency	Completion Time	Average CPU %
NFSv3	55118	71.9	1.6ms	54.7 seconds	51%

NFSv4.1	25068	13.9	11.5ms	283.5 seconds	24%

As you can see, the latency is higher, IOPS and throughput is lower and the completion time is nearly 5x for this workload when using NFSv4.1. The CPU utilization is lower because the storage isn't being asked to do as much (such as, process as many IOPS).

That doesn't mean NFSv4.x always performs worse than NFSv3; in some cases, it can perform as well or better. It all depends on the type of workload being used.

For a highly sequential write workload, NFSv4.1 was able to compete with NFSv3, since there is less metadata to process. Using a multithreaded dd operation to create eight 10GB files, this was the NFSv3 workload profile:

```
Object: nfsv3
Instance: DEMO
    Counter                                                 Value
    ------------------------------ --------------------------------
    access_total                                               18
    create_total                                                8
    fsinfo_total                                                4
    getattr_total                                               7
    lookup_total                                               11
    mkdir_total                                                11
    null_total                                                  4
    pathconf_total                                              2
    total_ops                                                5357
    write_percent                                             99%
    write_total                                           1248306
```

In this case, we are at nearly 100% writes. For NFSv4.1, the write percentage is lower, but the accompanying metadata operations are not the types that incur performance penalties (COMPOUND and PUTFH).

```
Object: nfsv4_1
Instance: DEMO
    Counter                                                 Value
    ------------------------------ --------------------------------
    access_total                                               25
    close_total                                                 4
    compound_percent                                          33%
    compound_total                                        1238160
    create_session_total                                        4
    create_total                                               11
    destroy_clientid_total                                      3
    destroy_session_total                                       3
    exchange_id_total                                          16
    getattr_total                                              72
    getdeviceinfo_total                                         8
    getfh_total                                                28
    layoutget_total                                             8
    layoutreturn_total                                          1
    lookup_total                                                7
    open_total                                                  8
    putfh_percent                                             33%
    putfh_total                                           1238107
    putrootfh_total                                             2
    reclaim_complete_total                                      4
    sequence_percent                                          33%
    sequence_total                                        1238134
    total_ops                                               15285
    write_percent                                             33%
    write_total                                           1238032
```

This results in a much better performance comparison for NFSv4.1. IOPS, throughput and latency are nearly identical and NFSv4.1 takes just 8 seconds longer (~3.7% more) than NFSv3.

Table 31) NFSv3 vs. NFSv4.1 performance – High sequential writes.

Test	Average IOPS	Average MBps	Average Latency	Completion Time	Average CPU %
NFSv3	6085	383.1	5.4ms	216.6 seconds	28%
NFSv4.1	6259	366.3	4.4	224.7 seconds	26%

With a more standard benchmarking tool (vdbench), we see the differences between read and write performance with different workload types. NFSv3 performs better in most cases, but the gap isn't especially wide for workloads that read/write existing datasets. Writes tend to have a wider performance disparity, but read performance is nearly identical. Sequential writes actually performed a bit better for NFSv4.x in these tests as well.

Figure 59) Random reads, 4K, NFSv3 vs. NFSv4.x – IOPS/Latency.

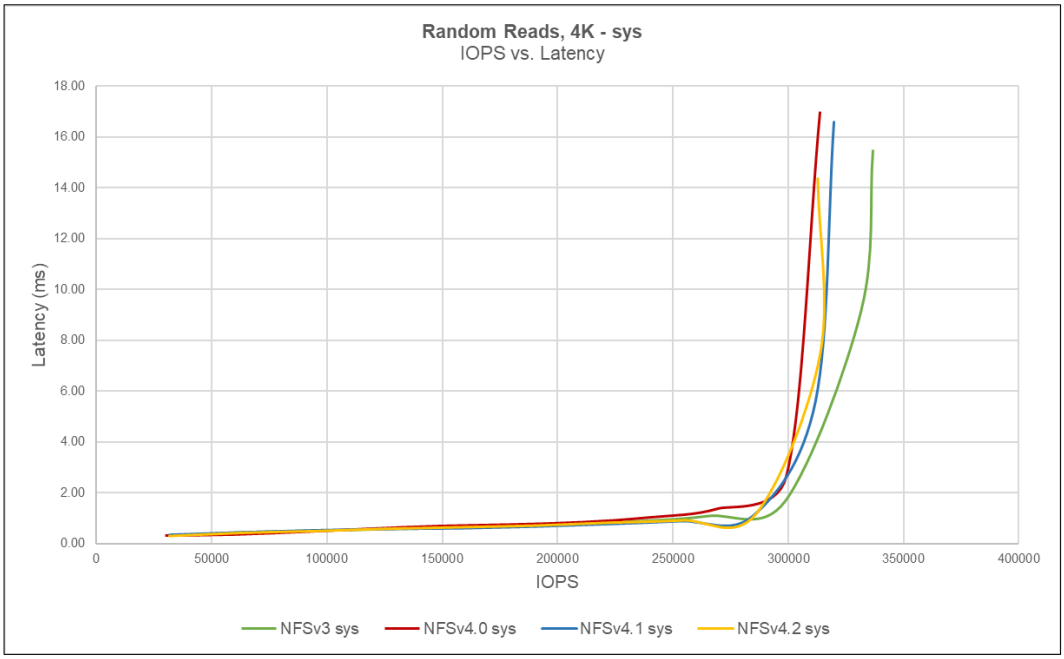


Figure 60) Random writes, 4K, NFSv3 vs. NFSv4.x – IOPS/Latency.

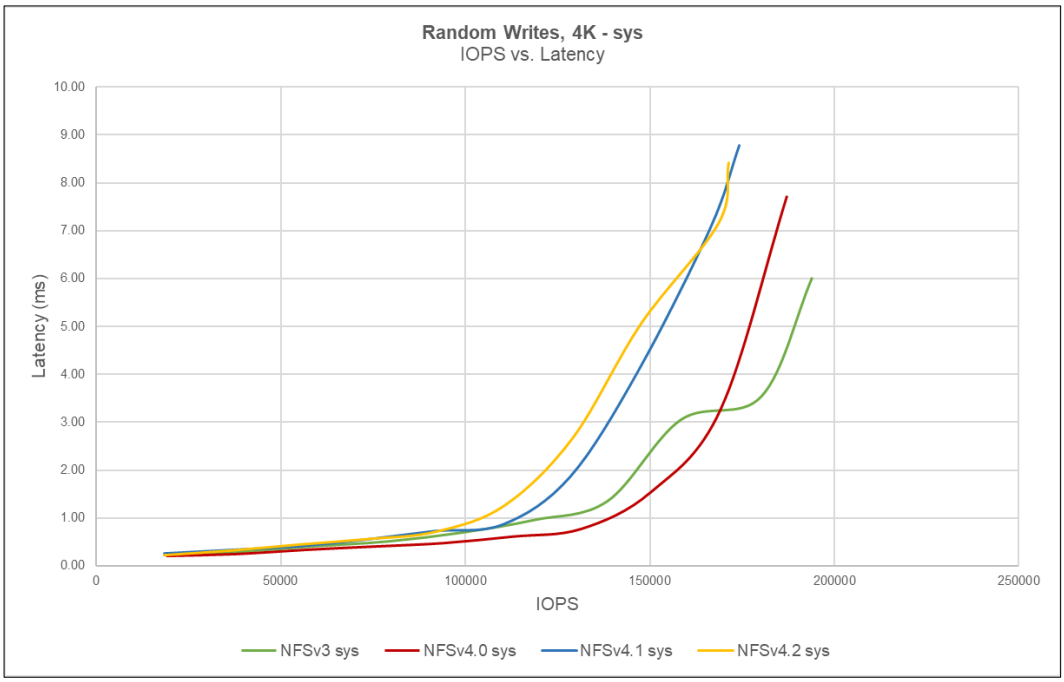


Figure 61) Sequential reads, 32K, NFSv3 vs. NFSv4.x – IOPS/Latency.

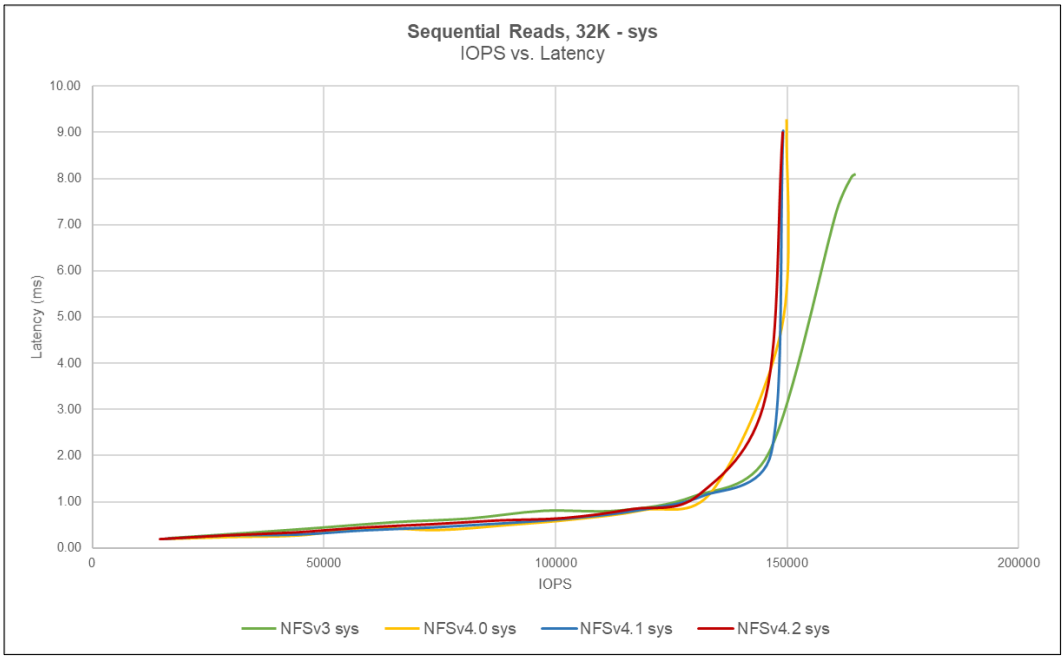
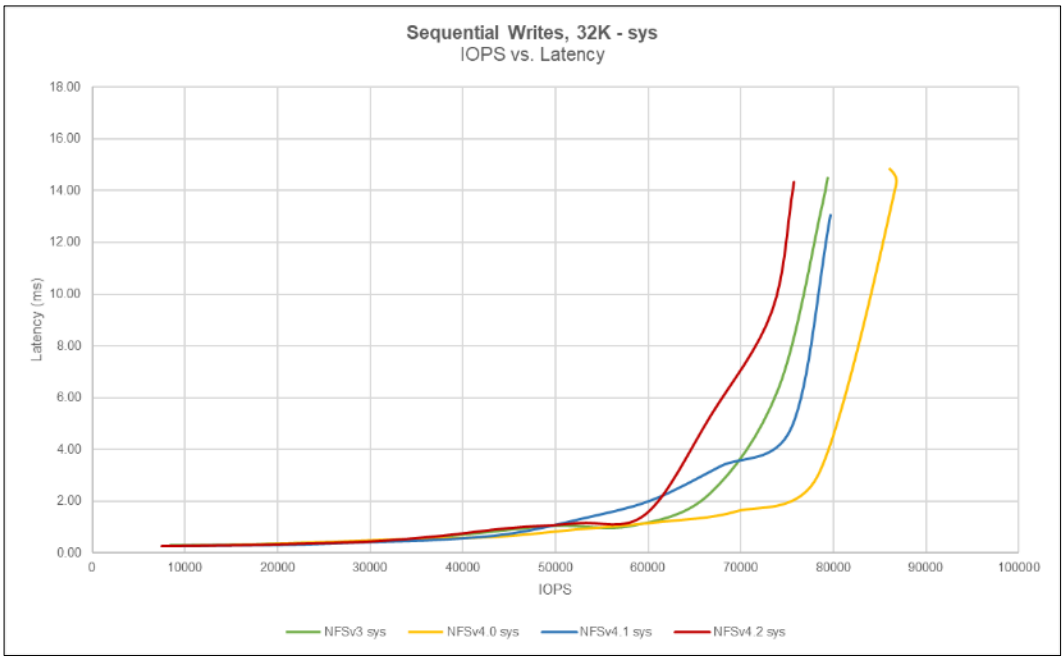


Figure 62) Sequential writes, 32K, NFSv3 vs. NFSv4.x – IOPS/Latency.



NFS write appends

In some cases, older releases of ONTAP might experience performance issues with file write appends over NFS based on how large the appends were. Later ONTAP releases provide parallel processing of these write appends to improve performance on write appends regardless of the file sizes involved.

See bug 1256520 for more information and guidance on which ONTAP releases have fixed this problem.

Nconnect

Nconnect is a mount option available in some Linux distributions. This option specifies how many TCP connections you should use per mount and offers substantial performance benefits in some workloads per client – generally only when the network threads are the bottleneck in a workload. This also provides benefits to ONTAP by allowing clients to leverage more RPC slot tables per mount. See the section “Network connection concurrency and TCP slots: NFSv3” for details on RPC slot tables.

ONTAP 9.8 offers support for the use of nconnect with NFS mounts, provided the NFS client also supports it. If you wish to use nconnect, check to see if your client version provides it and use ONTAP 9.8 or later.

Table 32 shows results from a single Ubuntu client using different nconnect thread values.

Table 32) Nconnect performance results.

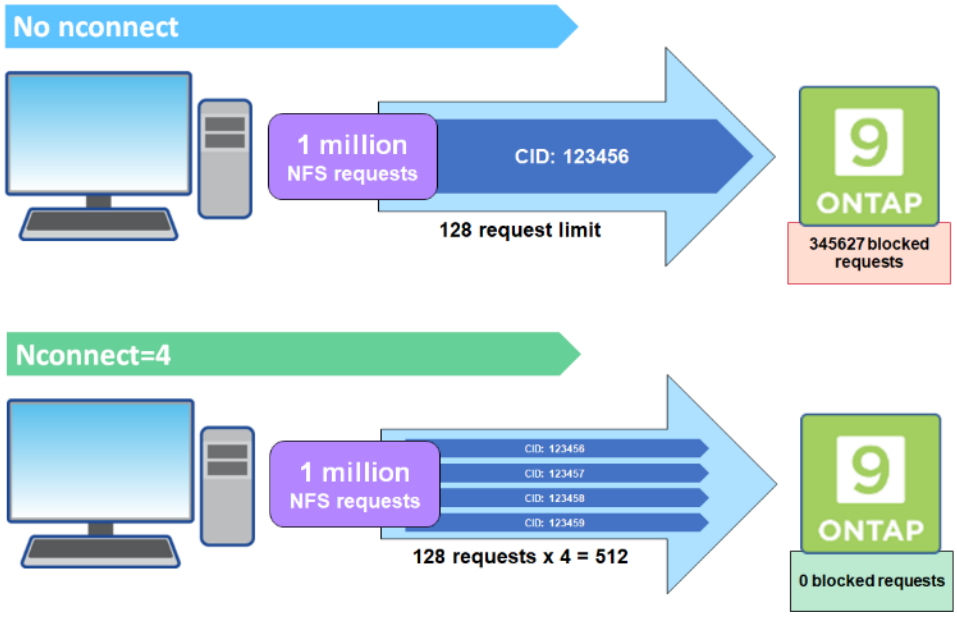
Nconnect value	Threads per process	Throughput	Difference
1	128	1.45GB/s	-
2	128	2.4GB/s	+66%
4	128	3.9GB/s	+169%
8	256	4.07GB/s	+181%

How can I tell nconnect is working?

Nconnect is designed to allocate more sessions across a single TCP connection. This helps to better distribute NFS workloads and add some parallelism to the connection, which helps the NFS server handle the workloads more efficiently. In ONTAP, when an NFS mount is established, a CID is created. That CID provides up to 128 concurrent in-flight operations. When that number is exceeded by the client, ONTAP enacts a form of flow control until it can free up some available resources as other operations complete. These pauses usually are only a few microseconds, but over the course of millions of operations, those can add up and create performance issues. Nconnect can take the 128 limit and multiply it by the number of nconnect sessions on the client, which provides more concurrent operations per CID and can potentially add performance benefits, as seen in Table 33.

Figure 63 illustrates how mounts without nconnect handle concurrent operations and how nconnect works to distribute operations to NFS mounts.

Figure 63) NFS mounts with and without nconnect.



To determine whether nconnect is indeed working in your environment, you can verify a few things.

When nconnect is not being used, a single CID is established per client mount. You can verify those CIDs by running the following command:

```
cluster::> network connections active show -node [nodes] -service nfs* -remote-host [hostname]
```

For example, this is the output from an active NFS connection without nconnect:

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                    Protocol/Service
----------- ---------------------- ---------------------------- ----------------
Node: node1
DEMO        data1:2049             centos83-perf.ntap.local:1013
                                                                TCP/nfs
```

When nconnect is in use, more CIDs per mount are present. In this example, we used nconnect=8.

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
```

```
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: node1
DEMO       data1:2049             centos83-perf.ntap.local:669 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:875 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:765 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:750 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:779 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:773 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:809 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:897 TCP/nfs
```

Another way to determine whether nconnect is being used is through a statistics capture for the CID object. You can start the statistics for that object by running the following command:

```
cluster::> set diag
cluster::*> statistics start -object cid
When that object runs, it tracks the number of total allocated cids (alloc_total).
```

For example, this is the number of `alloc_total` for a mount without nconnect:

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    11
This is from a mount with nconnect=4:
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    16
This is the alloc_total from nconnect=8:
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    24
```

Mapping NFS connected clients to volume names

To check what version of NFS is being mounted from the cluster, use the `nfs connected-clients show` command available in ONTAP 9.7:

```
cluster::> nfs connected-clients show -node * -vserver DEMO

     Node: node1
  Vserver: DEMO
  Data-Ip: 10.x.x.x
Client-Ip       Volume-Name      Protocol Idle-Time     Local-Reqs Remote-Reqs
--------------- ---------------- -------- ------------- ---------- -----------
10.x.x.x        CIFS             nfs4.1   2d 0h 9m 3s   153        0
10.x.x.x        vsroot           nfs4.1   2d 0h 9m 3s   0          72
10.x.x.x        flexgroup_16__0001
                                 nfs3     0s            0          212087
10.x.x.x        flexgroup_16__0002
                                 nfs3     0s            0          192339
10.x.x.x        flexgroup_16__0003
                                 nfs3     0s            0          212491
10.x.x.x        flexgroup_16__0004
                                 nfs3     0s            0          192345
10.x.x.x        flexgroup_16__0005
                                 nfs3     0s            212289     0
```

To avoid issues with mounting a FlexGroup volume in environments in which NFSv4.x is enabled, either configure clients to use a default mount version of NFSv3 through `fstab` or explicitly specify the NFS version when mounting.

If a FlexGroup volume is junctioned to a parent volume that is mounted to a client with NFSv4.x, traversing to the FlexGroup volume in ONTAP 9.6 and earlier fails because no NFSv4.x operations are allowed on FlexGroup volumes.

For example, FlexGroup volumes are always mounted to the vsroot (vserver root), which operates as (/) in the NFS export path. If a client mounts vsroot with NFSv4.x, attempts to access a FlexGroup volume in ONTAP 9.6 or earlier from the NFSv4.x mount fail. This includes `ls -la` operations because they require the ability to do NFSv4.x `GETATTR` operations.

Note in the following example that the information for the FlexGroup volumes is incorrect because of the lack of NFSv4.x support:

```
# mount demo:/ /mnt
# mount | grep mnt
demo:/ on /mnt type nfs (rw,vers=4,addr=10.193.67.237,clientaddr=10.193.67.211)
# cd /mnt/flexgroup
-bash: cd: /mnt/flexgroup: Permission denied
# ls -la
ls: cannot access flexgroup_4: Permission denied
ls: cannot access flexgroup_local: Permission denied
ls: cannot access flexgroup_8: Permission denied
ls: cannot access flexgroup_16: Permission denied
drwx--x--x. 12 root root   4096 Mar 30 21:47 .
dr-xr-xr-x. 36 root root   4096 Apr  7 10:30 ..
d??????????  ? ?    ?         ?            ? flexgroup_16
d??????????  ? ?    ?         ?            ? flexgroup_4
d??????????  ? ?    ?         ?            ? flexgroup_8
```

Compare that to the NFSv3 mount:

```
# ls -la
drwx--x--x. 12 root root 4096 Mar 30 21:47 .
dr-xr-xr-x. 36 root root 4096 Apr  7 10:30 ..
drwxr-xr-x.  6 root root 4096 May  9 15:56 flexgroup_16
drwxr-xr-x.  5 root root 4096 Mar 30 21:42 flexgroup_4
drwxr-xr-x.  6 root root 4096 May  8 12:11 flexgroup_8
```

As a result, be sure to avoid using NFSv4.x in any path where a FlexGroup volume resides in ONTAP 9.6 or earlier. If NFSv4.x is desired, upgrade ONTAP to 9.7 or later.

NFS server tuning

In ONTAP, most of the server tuning is done dynamically, such as window sizing and NAS flow control, as described in TR-4067. This section covers NFS server-specific tuning recommended for EDA workloads.

Max TCP transfer size/read and write size

Before ONTAP 9.0, NFS mounts negotiated rsize and wsize values based on the following options:

```
v3-tcp-max-read-size
v3-tcp-max-write-size
```

This value is 64k (65536), by default. ONTAP 9.0 and later versions deprecate these options and consolidate read and write sizes under the single option `tcp-max-xfer-size`. When a client mounts, if no rsize and wsize is specified, the client negotiates the read and write sizes to the value specified in `tcp-max-xfer-size`. The recommended value for EDA workloads is 64K (65536).

NFS/compute node considerations

When mounting NFS to a client running EDA workloads, there are some mount considerations you should review to achieve the best possible results. You should use the following mount options in most cases unless the need to deviate arises:

```
vers=3,rw,bg,hard,rsize=65536,wsize=65536,proto=tcp,intr,timeo=600
```

In many cases, these mount options auto-negotiate or use default values from the client. Check your client version and the default mount options in the NFS client configuration files to verify. If desired, change the default mount options for your client to ensure that these mount options are always used, or use the `/etc/fstab` file to specify mount options.

Mount options

NFS mount option recommendations depend solely on the workload and application being used. There is general advice for specific mount options, but the decision on which NFS options to use is dependent on the client operating system administrators and application vendor recommendations. There is no catch all recommendation for mount options. The following sections covers only a subset of NFS mount options. For a complete list of supported NFS mount options, use `man nfs` on your NFS client.

Default mount options

Linux clients set default mount options out of the box. These defaults depend on client operating system version and NFS configuration files found on the clients. Default mount options are what is set during a mount operation where no options are specified with the `-o` flag.

In some cases, mount options are negotiated with the NFS server. Specifically, in newer Linux kernels, the rsize/wsize values and the NFS versions are based on what the NFS server is set to.

For example, if NFSv4.1 is enabled and no NFS version is specified in configuration files or with the mount command, then the client will use NFSv4.1 as it is the highest supported NFS version enabled.

The following shows output from a mount command that issued no specific options. The ONTAP NFS server was set to 1MB TCP max transfer size (`-tcp-max-xfer-size`) and has NFSv4.1 enabled.

```
# mount DEMO:/flexgroup_16 /flexgroup
# mount | grep flexgroup
DEMO:/flexgroup_16 on /flexgroup type nfs4
(rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=0,timeo=600,retr
ans=2,sec=sys,clientaddr=10.x.x.x,local_lock=none,addr=10.x.x.y)
```

Wsize and rsize

The mount options wsize and rsize determine how much data is sent between the NFS client and server for each packet sent. This might help optimize performance for specific applications but should be set as per application vendor best practices because what is best for one application might not be best for other applications.

Newer NFS clients will autonegotiate the wsize and rsize values to what the `-tcp-max-xfer-size` value is set to on the ONTAP NFS server if the mount command does not explicitly set the values. ONTAP defaults `-tcp-max-xfer-size` to 64K and can be set to a maximum of 1MB.

Note: The general recommendation for `-tcp-max-xfer-size` is to set the value in ONTAP to 262144 (256K) for SSDs and either 65536 (64K) or 131072 (128K) for spinning disk.

For examples of some performance testing with different workload types and different wsize/rsize values, see TR-4067.

For examples of wsize/rsize recommendations for specific applications, see:

- TR-3633: Oracle on NetApp ONTAP

- TR-4435: SAP HANA on NetApp AFF Systems using NFS

Actimeo and nocto

NFSv3 manages shared file system consistency and application performance by using cached file/directory data and cached file/directory attributes. It is a loose consistency because the application does not have to go to shared storage and fetch data every time to use it. This can have a tremendous impact on application performance. The cached information has timers that set the period to trust the cache data and at timeout, a light weight, fast `GETATTR`/access call to revalidate the data until the next time out.

There are two mechanisms that manage this process:

- Close to open consistency assures getting the latest data for a file, regardless of cache (cto)
- Attribute cache timer (actimeo; default 3s for file, 30s for directory)

If the client has complete ownership of data, that is, it is not shared, there is guaranteed consistency. You can reduce `GETATTR`/access operations to storage and speed up the application by turning off cto consistency (nocto as a mount option) and by turning up the timeouts for the attribute cache management (actimeo=600 as a mount option changes the timer to 10m versus the defaults mentioned above). In some testing, nocto reduces ~65-70% of the `GETATTR`/access calls and actimeo reduces another ~20-25%.

There are other cases that can benefit from a similar set of mount options, even though there is not complete ownership by the clients. For applications that use grids of clients like EDA, web hosting and movie rendering and have relatively static data sets (like the tools/libraries in EDA, like the web content for the web hosting, like the textures for rendering), the typical behavior is the data set is largely cached on the clients (very few reads; no writes). Many `GETATTR`/access calls come back to storage in these cases. These data sets are typically updated through either another client mounting the file systems and periodically pushing content updates, or in some cases SnapMirror relationships, to multiple file systems for update.

In these cases, there is a known lag in picking up new content and the application still works with potentially out of date data. For these scenarios, you can use nocto and actimeo to control the time period where out of data date can be managed. For example, in EDA with tools and libraries and other static content, actimeo=600 works well because this data is typically updated infrequently. For small web hosting where clients need to see their data updates timelier as they are editing their sites, actimeo=10 might be acceptable. For large scale web sites where there is content pushed to multiple file systems, actimeo=60 might be more effective. As always, test with your individual environments.

Using these mount options reduces the workload to storage significantly in these cases (for example, a recent EDA experience reduced IOPs to the tool volume from >150K to ~6K) and applications can run significantly faster because they can trust the data in memory, rather than needing to query the NFS storage. This also helps reduce overall CPU % and load on the ONTAP nodes.

Actimeo

The actimeo mount option controls the attribute cache timeout on NFS clients. The actimeo option covers the entire range of available attribute caches, including:

```
acregmin=n
The minimum time (in seconds) that the NFS client caches attributes of a regular file before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 3-second minimum.

acregmax=n
The maximum time (in seconds) that the NFS client caches attributes of a regular file before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 60-second maximum.
```

```
acdirmin=n
The minimum time (in seconds) that the NFS client caches attributes of a directory before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 30-second minimum.

acdirmax=n
The maximum time (in seconds) that the NFS client caches attributes of a directory before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 60-second maximum.
```

Attribute caching provides some relief on networks by reducing the number of metadata calls. This also helps reduce latency to some workloads, as these metadata operations can now occur locally on the client. Attribute caching generally has no impact on the number of overall operations, unless all operations to the storage are metadata – specifically `ACCESS` calls.

For example, in our Customer Proof of Concept labs, actimeo was set to 10 minutes (600 seconds) and saw latency cut in half with an EDA workload generated by vdbench (from ~2.08ms to ~1.05ms).

Figure 64) Default actimeo latency—vdbench.

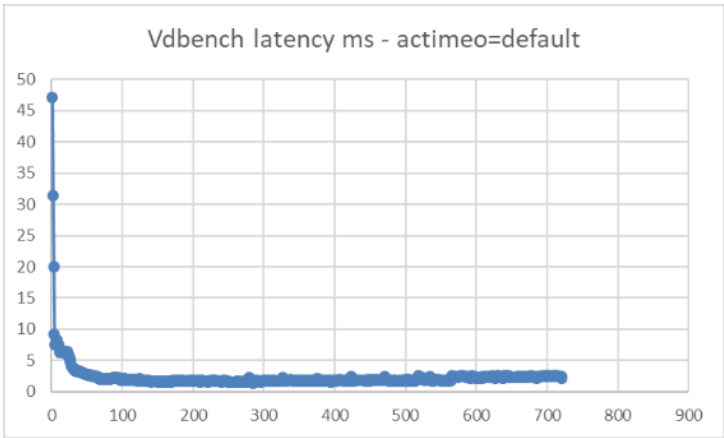
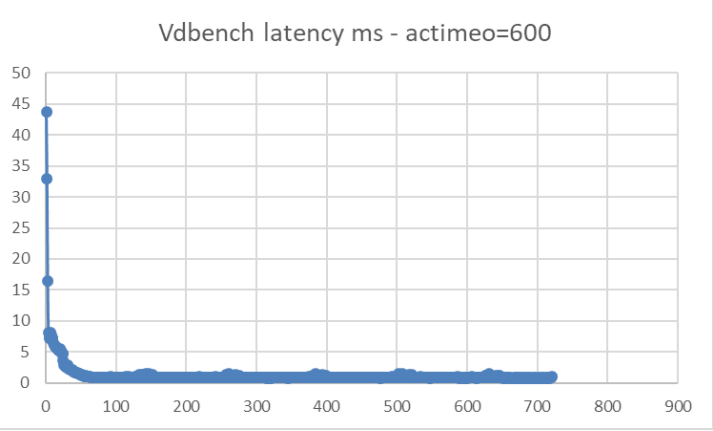


Figure 65) Actimeo=600 latency—vdbench.



The downside of setting the actimeo value too high is that attributes that change might not be reflected properly until the cache timeout occurs, which could result in unpredictable access issues.

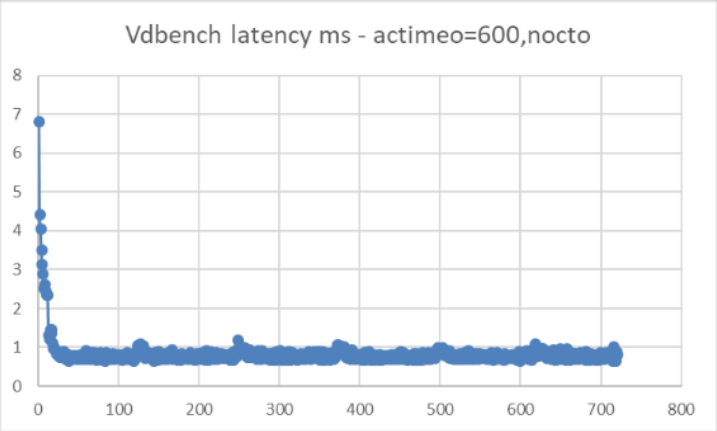
Note: The recommendation for attribute caching is to leave the defaults, unless otherwise directed by the application vendor or if testing shows a vast improvement in performance.

Nocto

Nocto stands for “no close-to-open,” which means that a file can close before a write has completed to save time. In NFS environments, this means that other clients that have that file open for reading do not get consistent updates to the file. By default, nocto is not set on NFS mounts, which means that all files wait to finish writes before allowing a close.

The nocto option is used primarily to increase raw performance. For example, in the same vdbench tests run in our Customer Proof of Concept Labs, the nocto mount option reduced latency by an additional .35ms to .7ms.

Figure 66) Actimeo=600,nocto latency—vdbench.



Note: The recommendation for use of the nocto option is to use only with read-heavy/read-mostly workloads.

NFSv4.1/pNFS client tuning recommendations

- Turn off hyperthreading on the BIOS of each of the Linux nodes
- Use the following mount options:

```
vers=4,rsize=65536,wsize=65536,hard,proto=tcp,timeo=600,minorversion=1
```

- Set up or configure NTP or time services on all compute nodes
- Set the `tuned-adm profile latency performance` for compute-intensive workloads. The following parameters are changed at the kernel level:
  - For `/sys/block/sdd/queue/scheduler`, set to `[deadline]` ; the default is `[cfq]`
- For `/etc/sysconfig/cpuspeed`, `governor` should be set to `performance`; default is `governor` set to nothing. This uses the performance governor for p-states through cpuspeed
- In RHEL 6.5 and later, the profile requests a `cpu_dma_latency` value of `1`
- Disable irqbalance
- Set `net.core.netdev_max_backlog = 300000`

Nconnect

A NFS mount option called nconnect is in its nascent stages for use with NFS mounts. nconnect is only available on most Linux clients. Be sure to verify with the OS vendor documentation if the option is supported in your kernel.

The purpose of nconnect is to provide multiple transport connections per TCP connection or mount point on a client. This helps increase parallelism and performance for NFS mounts.

ONTAP 9.8 and later offers official support for the use of nconnect with NFS mounts, provided the NFS client also supports it. If you would like to use nconnect, check to see if your client version provides it and use ONTAP 9.8 or later. ONTAP 9.8 and later supports nconnect by default with no option needed.

Note: Nconnect is not recommended for use with NFSv4.0.
NFSv3, NFSv4.1, and NFSv4.2 should work fine with nconnect.

Table 33 shows results from a single Ubuntu client using different nconnect thread values.

Table 33) Nconnect performance results.

Nconnect Value	Threads per process	Throughput	Difference
1	128	1.45GB/s	-
2	128	2.4GB/s	+66%
4	128	3.9GB/s	+169%
8	256	4.07GB/s	+181%

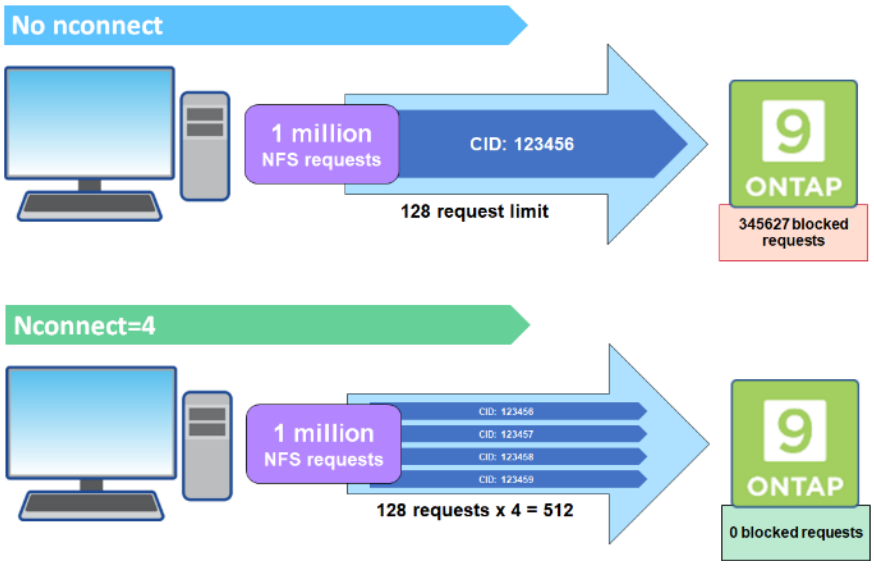
Note: The recommendation for using nconnect depends on client OS and application needs. Testing this new option is highly recommended before deploying in production.

How can I tell if nconnect is working?

Nconnect is designed to allocate more sessions across a single TCP connection. This helps to better distribute NFS workloads and add some parallelism to the connection, which helps the NFS server handle the workloads more efficiently. In ONTAP, when an NFS mount is established, a CID is created. That CID provides up to 128 concurrent in-flight operations. When that number is exceeded by the client, ONTAP will enact a form of flow control until it can free up some available resources as other operations complete. These pauses usually are only a few microseconds, but over the course of millions of operations, those can add up and create performance issues. Nconnect can take that 128 limit and multiply it by the number of nconnect sessions on the client, which provides more concurrent operations per CID and can potentially add performance benefits, as seen in Table 33.

Figure 67 illustrates how mounts without nconnect handle concurrent operations and how nconnect works to distribute operations to NFS mounts.

Figure 67) NFS mounts with and without nconnect.



To see if nconnect is indeed working in your environment, you can check a few things.

When nconnect is not being used, a single CID is established per client mount. You can see those CIDs with the following command:

```
cluster::> network connections active show -node [nodes] -service nfs* -remote-host [hostname]
```

For example, this is the output from an active NFS connection without nconnect:

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: node1
DEMO       data1:2049             centos83-perf.ntap.local:1013
                                                               TCP/nfs
```

When nconnect is in use, more CIDs per mount will be present. In this case, we used `nconnect=8`.

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
Vserver    Interface              Remote
Name       Name:Local Port        Host:Port                    Protocol/Service
---------- ---------------------- ---------------------------- ----------------
Node: node1
DEMO       data1:2049             centos83-perf.ntap.local:669 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:875 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:765 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:750 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:779 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:773 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:809 TCP/nfs
DEMO       data1:2049             centos83-perf.ntap.local:897 TCP/nfs
```

Another way to see if nconnect is being used is through a statistics capture for the CID object. You can start the statistics for that with the following command:

```
cluster::> set diag
cluster::*> statistics start -object cid
```

When that object runs, it will track the number of total allocated CIDs (`alloc_total`).

For example, this is the number of `alloc_total` for a mount without nconnect:

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    11
```

This is from a mount with `nconnect=4`:

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    16
```

And this is the `alloc_total` from `nconnect=8`:

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_total                                                    24
```

NFSv4.1/pNFS3—ONTAP considerations

- Enable read and write file delegations for NFSv4.1 to promote aggressive caching on single writer applications.
- pNFS provides data locality. You can access a volume over a direct path from anywhere in a cluster. Consider using pNFS for workloads that are mostly sequential I/O; high metadata workload performance might not see benefits from pNFS.
- If a volume is moved for capacity or workload balancing, there is no requirement to move or migrate the LIF around in the cluster namespace to provide local access to the volumes. pNFS handles the path redirection.
- NFSv4.1 is a stateful protocol, unlike NFSv3. If there is ever a requirement to migrate a LIF, the I/O operations stall for up to 45 seconds to migrate the lock states over to the new location.

NFS Readahead

Readahead in NFS is a way for NFS clients to predictively request blocks of a file to improve performance and throughput for sequential I/O workloads. Until recently, the readahead value for NFS mounts was set to 15 times the rsize value of the mount. For example, if you set rsize to 64KiB, then readahead size would be 960KiB.

In modern NFS clients (such as RHEL 8.3 and later or Ubuntu 18.04 and later), the readahead value is no longer determined by the mount rsize, but is globally defaulted to 128KiB. This can cause severe negative performance consequences on reads. In newer Linux client versions that use the default 128KiB readahead value, it is recommended to set the value to a higher limit. Testing read performance with different values is the preferred method, but internal NetApp testing has found that the value can be safely set to as high as 15360KiB for sequential read workloads.

For more information about setting and viewing readahead values, consult with your client OS vendor. For example, this SUSE KB describes readahead for those OS client flavors: https://www.suse.com/support/kb/doc/?id=000017019.

For a CentOS/RedHat client, the process is similar.

```
# cat /etc/redhat-release
CentOS Linux release 7.8.2003 (Core)
```

Find the BDEV information for the mount with this command (BDEV format is N:NN):

```
# cat /proc/self/mountinfo | grep /mnt/client1
125 39 0:46 / /mnt/client1 rw,relatime shared:107 - nfs DEMO:/files
rw,vers=3,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mount
addr=10.193.67.219,mountvers=3,mountport=635,mountproto=udp,local_lock=none,addr=10.193.67.219
```

Use the BDEV information to find the readahead value (BDEV for that mount point is 0:46):

```
# cat /sys/class/bdi/0:46/read_ahead_kb
15360
```

In the above example, readahead is set to 15360KiB (15x the rsize) for the `/mnt/client1` mount and rsize is set to 1MB on my CentOS 7.8 client.

On CentOS 8.3, this is the value my mount is set to by default:

```
# cat /sys/class/bdi/0:50/read_ahead_kb
```

NFS in ONTAP resources

The following technical reports provide more information about NFS in ONTAP:

- TR-4067: NFS Implementation and Best Practice Guide

- TR-4668: Name Services Best Practices
- TR-4571: NetApp ONTAP FlexGroup Volumes Best Practices
- TR-4616: NFS Kerberos in ONTAP
- TR-4835: How to Configure LDAP in ONTAP
- TR-4887: Multiprotocol NAS Overview and Best Practices

Export policy considerations

ONTAP maintains a cache of export policies and rules to reduce the number of requests needed each time a client mounts the cluster. The contents of the cache can be viewed with the `export-policy access-cache show` command, which requires the SVM and client IP address to be specified.

Example:

```
::> export-policy access-cache show -node node-01 -vserver DEMO -policy default -address
10.193.67.225

                                        Node: node-01
                                     Vserver: DEMO
                                 Policy Name: default
                                  IP Address: 10.193.67.225
                    Access Cache Entry Flags: has-usable-data
                                 Result Code: 0
                First Unresolved Rule Index: -
                     Unresolved Clientmatch: -
              Number of Matched Policy Rules: 1
         List of Matched Policy Rule Indexes: 1
                                Age of Entry: 3s
                Access Cache Entry Polarity: positive
Time Elapsed since Last Use for Access Check: 3s
      Time Elapsed since Last Update Attempt: 3s
               Result of Last Update Attempt: 0
                List of Client Match Strings: 0/0
```

The export policy access cache has a default configuration that can be seen with the `export-policy access-cache config show` command.

Example:

```
::> export-policy access-cache config show -vserver DEMO

                          Vserver: DEMO
TTL For Positive Entries (Secs): 3600
TTL For Negative Entries (Secs): 3600
         Harvest Timeout (Secs): 86400
             Is Dns TTL Enabled: false
```

The above means that positive entries (clients that were found in the policy and allowed access) will stay cached for 3600 seconds. Negative entries (clients that were *not* found in the policy and were denied access) will also stay cached for 3600 seconds. The “harvest” timeout of 86400 seconds is how often ONTAP will purge unused entries from the cache.

When an export policy is changed to add or remove rules, the access cache is automatically refreshed to ensure ONTAP has the most up to date information to prevent unwanted access from clients that may have been removed from the policy and to prevent unwanted denials of access to clients that may have been added.

In the event where there are many clients connected to the cluster, with many entries in the cache, flushing all entries may create a scenario where CPU on the cluster is taxed when policies are re-populated, which can introduce a performance impact to production workloads.

If you have many clients connecting to a cluster (think thousands), consider contacting NetApp technical support before updating export policy rules for potential workarounds to caches updating upon changes.

Security considerations

NetApp ONTAP provides multiple ways to secure your EDA environment, aside from the basic permissions and export policies offered by NAS environments. Security concepts for NAS take two main forms – in-flight and at-rest.

- **In-flight encryption** provides a way to protect data packets from being intercepted or sniffed in transit between clients and storage.
- **At-rest encryption** provides a way to protect data on the storage system by leveraging security keys that are unique to the system, so that if physical hardware is taken, the data is inaccessible.

For details about security hardening in ONTAP, see TR-4569: Security Hardening Guide for ONTAP.

At-rest encryption

ONTAP has three FIPS 140-2–compliant data-at-rest encryption solutions:

- NetApp Storage Encryption (NSE) is a hardware solution that uses self-encrypting drives.
- NetApp Volume Encryption (NVE) is a software solution that enables encryption of any data volume on any drive type where it is enabled with a unique key for each volume.
- NetApp Aggregate Encryption (NAE) is a software solution that enables encryption of any data volume on any drive type where it is enabled with unique keys for each aggregate.

NSE, NVE, and NAE can use either external key management or the onboard key manager (OKM). Use of NSE, NVE, and NAE does not affect ONTAP storage efficiency features. However, NVE volumes are excluded from aggregate deduplication. NAE volumes participate in and benefit from aggregate deduplication.

The OKM provides a self-contained encryption solution for data at rest with NSE, NVE, or NAE. NVE, NAE, and OKM use the ONTAP CryptoMod. CryptoMod is now listed on the CMVP FIPS 140-2 validated modules list. See FIPS 140-2 Cert# 3387.

NetApp Volume Encryption (NVE) FlexGroup volume considerations

ONTAP 9.2 introduced support for NetApp Volume Encryption (NVE) for FlexGroup volumes. Implementing this feature with FlexGroup volumes follows the same recommendations and best practices as stated for FlexVol volumes, except that NVE cannot be enabled on existing FlexGroup volumes. Currently, only new FlexGroup volumes can use NVE. To encrypt existing FlexGroup volumes, you must create a new volume with encryption enabled and then copy the data to the volume at the file level, for example with XCP.

Generally, NVE requires:

- A valid NVE license
- A key management server
- A cluster-wide passphrase (32 to 256 characters)
- FAS or AFF hardware that supports AES-NI offloading

For information about implementing and managing NVE with FlexGroup and FlexVol volumes, see the "NetApp Encryption Power Guide" and "Scalability and Performance Using FlexGroup Volumes Power Guide" on the support site for your release of ONTAP. For FlexGroup information, see TR-4571: NetApp ONTAP FlexGroup Volumes Best Practices.

In-flight encryption

ONTAP supports NAS-protocol specific encryption (such as SMB3 encryption and signing/sealing or NFS Kerberos5p) for SMB/NFS communication, which uses AES encryption to wrap client/server communication to prevent man-in-the-middle attacks and packet sniffing. These encryption methods require client and protocol version support to function, as well as proper infrastructure to manage the ticket exchanges.

Note: For details on configuring NFS Kerberos, see TR-4616: NFS Kerberos in ONTAP.

Additionally, ONTAP 9.8 introduced support for Internet Protocol Security (IPsec). IPsec provides end-to-end encryption support for all IP traffic between a client and an ONTAP SVM. IPsec data encryption for all IP traffic includes NFS, iSCSI, and SMB/CIFS protocols. IPsec provides the only encryption in flight option for iSCSI traffic.

Providing NFS encryption over the wire is one of the main use cases for IPsec. Before ONTAP 9.8, NFS over-the-wire encryption required the setup and configuration of Kerberos to use krb5p to encrypt NFS data in flight. This is not always simple or easy to accomplish in every customer environment.

Customers who use data-at-rest encryption technologies such as NSE or NVE and Cluster Peering Encryption (CPE) for data replication traffic can now use end-to-end encryption between client and storage across their hybrid multi-cloud data fabric by upgrading to ONTAP 9.8 or later and using IPsec.

For details on IPsec, see TR-4569: Security Hardening Guide for ONTAP.

Cloud considerations

More and more EDA workloads are either considering a move or have already moved into the cloud for the cost benefits of renting compute (and spending only operational expenditures (OPEX)) for EDA jobs like kernel verifications and memory compiles, rather than spending capital expenditure (CAPEX) and OPEX on your own hardware and data center space.

Leveraging the cloud for these workloads also provides additional benefits, such as having the ability to localize these jobs anywhere in the world by spinning up an instance in a nearby cloud region. This becomes even more apparent in today's prevalent work from home climate.

ONTAP offers several cloud-resident features and options for storing EDA datasets, including Cloud Volumes ONTAP, Amazon FSx for NetApp ONTAP, Azure NetApp Files, and Google Cloud NetApp Volumes.

In addition, ONTAP offers features such as NetApp FabricPool for tiering cold data to the cloud, as well as NetApp FlexCache volumes for fast, local sparse caches of existing EDA datasets that can be deployed from on-premises or cloud instances. Using a combination of FlexCache and FlexGroup volumes can remove some of the complexities found in other EDA solutions, such as AWS on Lustre, as there is no need to constantly synchronize new data to the cloud, and creating a single large namespace is as simple as clicking a few buttons. FlexCache volumes fetch data automatically when a client requests it, but only when the client requests it—space is only consumed when it is actively being used. FlexGroup volumes manage the balancing of load across provisioned cluster resources and deliver parallel performance for high ingest workloads like EDA.

Check out EDA Cloud Challenges and How Cloud Volumes ONTAP Can Solve Them and Chip Design and the Azure Cloud: An Azure NetApp Files Story for more information about EDA in the cloud on NetApp Cloud Volumes ONTAP.

Object storage/S3

As EDA companies move closer to putting their workloads into the cloud, the concept of object storage is being given closer consideration as a way to serve grid workloads. This SemiWiki article touts the benefits of cloud for EDA compute, as well as noting some reasons why you might want to potentially move some of those datasets to an object store.

ONTAP 9.8 and later offers native S3 object support, as well as being able to serve both NFS and object workloads from the same cluster (though not to the same volumes). TR-4814: S3 in ONTAP Best Practices details how the ONTAP 9.8 S3 solution works and where it fits.

In addition, you can use ONTAP S3 as a tiering destination for cold data off of higher cost flash storage by using NetApp FabricPool. You can learn more about NetApp FabricPool in TR-4598: NetApp FabricPool Best Practices, as well as TR-4826: NetApp FabricPool with StorageGRID.

If you are looking for a larger scale object storage platform that can span multiple global data centers and offer a distributed namespace with more robust policy-driven object storage system than ONTAP currently offers with its S3 implementation, NetApp StorageGRID is a good fit.

Automation

Many EDA organizations are looking for ways to cut management and provisioning costs by using automation tasks that provide repeatable, simple, and scalable ways to present storage to end users. NetApp offers a wide array of Ansible certified storage modules for automation tasks. Automation using Ansible can help your organization save time and money on their EDA project deployments.

For more information and examples, see:

- Ansible and NetApp
- NetApp IT Insider Tips on ONTAP Management Using Ansible
- Getting Started with NetApp and Ansible: First Playbook Example
- NetApp.io: Configuration Management & Automation
- Tech ONTAP Podcast Episode 274: NetApp Ansible Updates - Winter 2020
- Tech ONTAP Podcast Episode 285: NetApp Trident Updates - Winter 2021
- Tech ONTAP Podcast Episode 285: Project Astra? Project No More!
- Tech ONTAP Podcast Episode 295: NetApp DataOps Toolkit

Migrating to NetApp FlexGroup volumes

One challenge in having many files or a massive amount of capacity is deciding how to move the data as quickly and as nondisruptively as possible. This challenge is greatest in high-file-count, high-metadata-operation workloads. Copies of data at the file level require file-system crawls of the attributes and the file lists, which can greatly affect the time that it takes to copy files from one location to another. That is not even considering aspects such as network latency, WANs, system performance bottlenecks, and other things that can make a data migration painful.

With NetApp ONTAP FlexGroup volumes, the benefits of performance, scale, and manageability are apparent. But how do you get there?

Data migrations with NetApp FlexGroup volumes can take three general forms:

- Migrating from third-party storage to NetApp FlexGroup volumes
- Migrating from NetApp Data ONTAP operating in 7-Mode to NetApp FlexGroup volumes

- Migrating from FlexVol volumes or Infinite Volume in ONTAP to NetApp FlexGroup volumes

The following sections discuss these use cases and how to approach them.

Migration by using NDMP

In ONTAP 9.7 and later, FlexGroup volumes support NDMP operations. These include the `ndmpcopy` command, which can be used to migrate data from a FlexVol volume to a FlexGroup volume. For information about setting up `ndmpcopy`, see: https://kb.netapp.com/app/answers/answer_view/a_id/1032750.

In the following example, `ndmpcopy` was used to migrate around five million folders and files from a FlexVol volume to a FlexGroup volume. The process took around 51 minutes (output shortened):

```
cluster::*> system node run -node node1 ndmpcopy -sa ndmpuser:AcDjtsU827tputjN -da
ndmpuser:AcDjtsU827tputjN 10.x.x.x:/DEMO/flexvol/nfs 10.x.x.x:/DEMO/flexgroup_16/ndmpcopy

Ndmpcopy: Starting copy [ 2 ] ...
Ndmpcopy: 10.x.x.x: Notify: Connection established
…
Ndmpcopy: 10.x.x.x: Log: RESTORE: Thu Jan  9 12:35:04 2020 : We have processed 4814787 files and
directories.
Ndmpcopy: 10.x.x.x: Log: RESTORE: RESTORE IS DONE
Ndmpcopy: 10.x.x.x: Notify: restore successful
Ndmpcopy: Transfer successful [ 0 hours, 50 minutes, 53 seconds ]
Ndmpcopy: Done
```

The same dataset using `cp` over NFS took 316 minutes—six times as long as `ndmpcopy`:

```
# time cp -R /flexvol/nfs/* /flexgroup/nfscp/

real    316m26.531s
user    0m35.327s
sys     14m8.927s
```

Using the NetApp XCP Migration Tool, that dataset took just under 20 minutes—or around 60% faster than `ndmpcopy`:

```
#  xcp copy 10.193.67.219:/flexvol/nfs 10.193.67.219:/flexgroup_16/xcp
Sending statistics...
5.49M scanned, 5.49M copied, 5.49M indexed, 5.60 GiB in (4.81 MiB/s), 4.55 GiB out (3.91 MiB/s),
19m52s.
```

Note: This XCP copy was done on a VM with a 1GB network and not much RAM or CPU; more robust servers will perform even better.

FlexVol to FlexGroup conversion

In ONTAP 9.7 and later, you can convert a single FlexVol volume to a FlexGroup volume containing a single member volume, in place, with less than 40 seconds disruption. This is regardless of the data capacity or number of files that reside in the volume. There is no need to remount clients, copy data, or make any other modifications that could create a maintenance window. After the FlexVol volume is converted to a FlexGroup volume, you can add new member volumes to the converted FlexGroup volume to expand the capacity.

When to convert a FlexVol volume to a FlexGroup volume

FlexGroup volumes offer a few advantages over FlexVol volumes, such as:

- Ability to expand beyond 100TB and 2 billion files in a single volume
- Ability to scale out capacity or performance nondisruptively
- Multi-threaded performance for high-ingest workloads

- Simplification of volume management and deployment

For example, perhaps you have a workload that is growing rapidly, and you do not want to have to migrate the data but still want to provide more capacity. Or perhaps a workload's performance is not good enough on a FlexVol volume, so you want to provide better performance handling with a FlexGroup volume. In this case, converting can help.

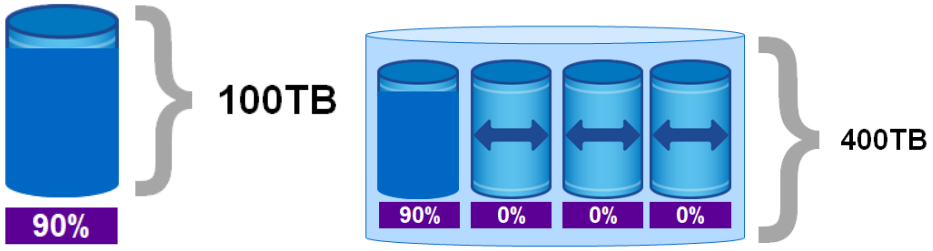
When not to convert a FlexVol volume

Converting a FlexVol volume to a FlexGroup volume might not always be the best option. If you require FlexVol features that are not available in FlexGroup volumes, then you should hold off. For example, SnapLock and SnapMirror Synchronous are not currently supported for FlexGroup volumes, so if you need them, you should stay with FlexVol volumes.

Also, if you have a FlexVol volume that is already very large (80–100TB) and already very full (80–90%), you might want to copy the data rather than convert, because the converted FlexGroup volume will have a very large, very full member volume. This can create performance issues and does not fully resolve your capacity issues, particularly if that dataset contains files that grow over time.

If you convert this 90% full volume to a FlexGroup volume, you will have a 90% full member volume. If you add new member volumes, they will be 100TB each and 0% full, so they will take on a majority of new workloads. The data does not rebalance and if the original files grow over time, you can still run out of space with nowhere to go (because 100TB is the maximum member volume size).

Figure 68) Converting a FlexVol volume that is nearly full and at maximum capacity.



Things that block a conversion

ONTAP can block FlexGroup conversion for a number of reasons, mainly to protect against converting a volume to an unsupported configuration. To learn more about the reasons ONTAP blocks a FlexGroup conversion, see TR-4571.

How it works

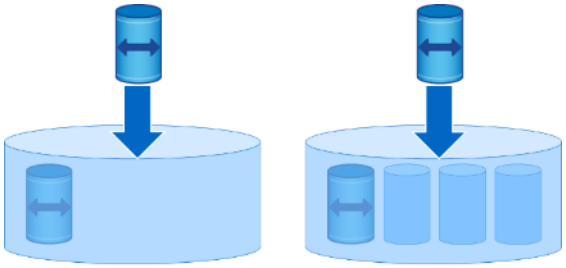
To convert a FlexVol volume to a FlexGroup volume in ONTAP 9.7 and later, run the following single, simple command at the advanced privilege level.

```
cluster::*> volume conversion start ?
-vserver <vserver name> *Vserver Name
[-volume] <volume name> *Volume Name
[ -check-only [true] ] *Validate the Conversion Only
[ -foreground [true] ] *Foreground Process (default: true)
```

When you run this command, ONTAP converts a single FlexVol volume into a FlexGroup volume with one member. You can even run a validation of the conversion before you do the real thing.

The process is 1:1, so you cannot currently convert multiple FlexVol volumes into a single FlexGroup volume. When the conversion is done, you have a single-member FlexGroup volume to which you can then add more member volumes of the same size to increase capacity and performance.

Figure 69) Converting a FlexVol volume to a FlexGroup and adding member volumes.



High file count impact on the conversion process

FlexGroup volume conversion does not convert any files; instead, it converts the volume's file identifiers, which are used when handing out new file handles to clients. This process takes less than a minute whether there are 300,000 files or 500,000,000 files. For more information and examples of FlexGroup conversions, see TR-4571.

Note: For a video example, see Statistics show-periodic during FlexVol - FlexGroup convert

Migrating from third-party storage to FlexGroup

When migrating from third-party storage (SAN or NAS), the migration path is a file-based copy. Various methods are available to perform this migration; some are free, and some are paid through third-party vendors.

For NFSv3-only data, NetApp strongly recommends the NetApp XCP Migration Tool. XCP is a free, license-based tool that can vastly improve the speed of data migration of high-file-count environments. XCP also offers robust reporting capabilities. XCP 1.5 and later versions also offer NFSv4.x and NFSv4.x ACL support, as well as being officially supported by NetApp. XCP 1.6 and later offers File System Analytics functionality.

Note: XCP is supported only for migration to a NetApp storage system. Use the latest available XCP release for best results.

For CIFS/SMB data, XCP for SMB is available. Robocopy is a free tool, but the speed of transfer depends on using its multithreaded capabilities. Third-party providers can also perform this type of data transfer.

NetApp XCP File Migration and Analytics

NetApp XCP is free and was designed specifically for scoping, migration, and management of large sets of unstructured NAS data. The initial version was NFSv3 only, but a CIFS version is now available. To use the tool, download it and request a free license (for software tracking purposes only).

XCP addresses the challenges that high-file-count environments have with metadata operation and data migration performance by using a multicore, multichannel I/O streaming engine that can process many requests in parallel.

These requests include the following:

- Data migration
- File or directory listings (a high-performance, flexible alternative to `ls`)
- Space reporting (a high-performance, flexible alternative to `du`)

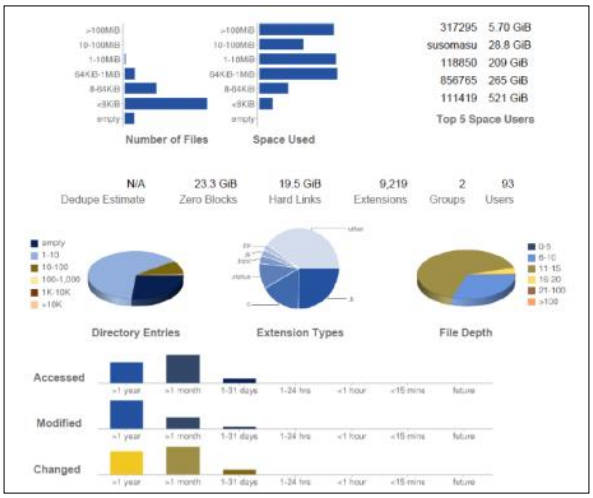
XCP has sometimes reduced the length of data migration by 20 to 30 times for high-file-count environments. In addition, XCP has reduced the file list time for 165 million files from 9 days on a competitor's system to 30 minutes on NetApp technology—a performance improvement of 400 times. As of XCP 1.5, the tool is officially supported by NetApp support.

XCP 1.6 also adds File Systems Analytics functionality. This is similar to the File Systems Analytics functionality added to ONTAP 9.8, but is able to scan systems that are not running ONTAP as well.

Note: For best results, use the latest XCP release available.

XCP also gives some handy reporting graphs, as shown in Figure 70.

Figure 70) XCP reporting graphs.



For more information, see the official XCP website at http://xcp.netapp.com.

Using XCP to scan files before migration

When deploying a FlexGroup volume, evaluate the file system and structure to help you determine initial sizing considerations and the best way to lay out member volumes. In high-file-count environments, this can be time consuming and tedious. XCP allows you to scan files and export to the CSV or XML format to easily review your file system. For an example, see TR-4571. For XCP Best Practices, see TR-4863.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites

Technical reports

- TR-4063: pNFS Best Practices
  www.netapp.com/pdf.html?item=/media/19761-tr-4063.pdf
- TR-4067: NFS Best Practices and Configuration Guide
  www.netapp.com/pdf.html?item=/media/10720-tr-4067.pdf
- TR-4100: Nondisruptive Operations with SMB File Shares
  www.netapp.com/pdf.html?item=/media/16338-tr-4100pdf.pdf
- TR-4571: FlexGroup Volumes Best Practice Guide
  www.netapp.com/pdf.html?item=/media/12385-tr4571pdf.pdf
- TR-4523: DNS Load Balancing in ONTAP
  www.netapp.com/pdf.html?item=/media/19370-tr-4523.pdf
- TR-4569: Security Hardening Guide for ONTAP
  docs.netapp.com/us-en/ontap/ontap-security-hardening/security-hardening-overview.html

- TR-4598: FabricPool Best Practices
  www.netapp.com/pdf.html?item=/media/17239-tr-4598.pdf
- TR-4616: NFS Kerberos in ONTAP
  www.netapp.com/pdf.html?item=/media/19384-tr-4616.pdf
- TR-4668: Name Services Best Practice Guide
  www.netapp.com/pdf.html?item=/media/16328-tr-4668pdf.pdf
- TR-4678: Data Protection and Backup for NetApp ONTAP FlexGroup Volumes
  www.netapp.com/pdf.html?item=/media/17064-tr4678pdf.pdf
- TR-4743: NetApp ONTAP FlexCache Volumes
  www.netapp.com/pdf.html?item=/media/7336-tr4743pdf.pdf
- TR-4814: S3 in ONTAP Best Practices
  www.netapp.com/pdf.html?item=/media/17219-tr4814pdf.pdf
- TR-4835: How to Configure LDAP in ONTAP
  www.netapp.com/pdf.html?item=/media/19423-tr-4835.pdf
- TR-4863: XCP Best Practices
  docs.netapp.com/us-en/netapp-solutions/xcp/xcp-bp-introduction.html
- TR-4867: Best Practice Guide for File System Analytics
  www.netapp.com/pdf.html?item=/media/20707-tr-4867.pdf

Miscellaneous content

- Tech ONTAP Podcast Episode 46: FlexGroups
- Tech ONTAP Podcast Episode 188: FlexGroup Update
- Tech ONTAP Podcast Episode 219: FlexVol to FlexGroup Conversion
- Tech ONTAP Podcast Episode 270: File System Analytics
- What's New For FlexGroup Volumes in ONTAP 9.3?
- FlexGroup Volumes: An Evolution of NAS
- 7 Myths about NetApp ONTAP FlexGroup Volumes
- Volume Affinities: How ONTAP and CPU Utilization Has Evolved
- FlexGroup lightboard video
- NetApp's FlexGroup Volumes – A Game Changer for EDA Workflows
- Concurrency and Collaboration – Keeping a Dispersed Design Team in Sync with NetApp
- EDA is Better in the Cloud. Here's Why the Cloud is the Future of EDA
- EDA Cloud Challenges and How Cloud Volumes ONTAP Can Solve Them
- Cloud Architects: Supercharge Your HPC Workloads in Azure
- Simple Solutions for Complex Azure EDA Workloads
- EDA Cloud Challenges and How Cloud Volumes ONTAP Can Solve Them
- Chip Design and the Azure Cloud: An Azure NetApp Files Story

Version history

Version	Date	Document version history
Version 1.0	August 2017	Justin Parisi: Initial commit.
Version 2.0	August 2020	Updates and fixes
Version 2.1	March 2021	New template, revisions and ONTAP 9.8 specific information added.

Version 2.2	September 2021	• Revisions and ONTAP 9.9.1 information • Storage efficiency section • TCP connection/mount storm section • Nconnect details • NFSv3 and NFSv4.x performance comparison
Version 2.21	February 2025	• Updates and fixes
Version 2.22	April 2025	• Added export policy considerations

Contact us

Let us know how we can improve this technical report.

Contact us at doccomments@netapp.com and include TECHNICAL REPORT 4617 in the subject line.

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

NetApp