Technical white paper

# HPE GreenLake for File Storage architecture

# Contents

# Executive summary

The world of hybrid cloud with the mainstream emergence of artificial intelligence across enterprises poses several challenges from an explosion of unstructured data and from the pervasive and ever-increasing demand to add more business value while at the same time future-proofing their IT infrastructure with scale-out architectures that can truly scale up to AI scale size.

Hewlett Packard Enterprise tackles these challenges head-on with **HPE GreenLake for File Storage solution**, which is built on the **HPE Alletra Storage MP** disaggregated platform in standard and high-density variants coupled with the HPE GreenLake for File Storage operating system (OS). This solution is built with a unique disaggregated and shared-everything (DASE) architecture to deliver enterprise performance at AI scale while providing an intuitive cloud experience with one of the most efficient capacities and performance per rack unit.

To further validate AI performance at AI scale, HPE GreenLake for File Storage has achieved NVIDIA DGX BasePOD certification and NVIDIA OVX storage validation in addition to NVIDIA GPUDirect Storage (GDS) across both standard and high-density variants.

## Target audience

This guide is intended for solution architects, field consultants, IT specialists, and anyone else who might need this information as they design and implement an HPE GreenLake for File Storage solution. The paper provides an overview of the solution with a technical deep dive into the hardware and software architecture. Readers are expected to understand the basics of file storage, x86 architecture, basic networking, authentication methods, and protocols such as NFS and SMB.

# Hardware architecture

HPE Alletra Storage MP has three underlying components.

1. Compute enclosure, also referred to as **CBox**
2. Storage enclosure, or JBOF, also referred to as **DBox**
3. Back-end NVMe over Ethernet fabric

HPE Alletra Storage MP comes in **standard-density** and **high-density** variants. The underlying storage architecture is disaggregated in nature, with the flexibility to scale up compute and storage independently as needed regardless of the underlying standard or high-density modules used. Both HPE Alletra Storage MP high-density and standard-density modules have been certified with NVIDIA DGX systems and validated with NVIDIA OVX systems.

**Standard-density** modules come with the CBox in a 2U footprint in standard rack depth with 2 CNodes with-in the CBox. For storage, the standard density modules come in a 2U DBox footprint also in standard rack depth and with 2 DNodes with-in the DBox. The back-end NVMe over Ethernet fabric comes in a 32-port 100GbE HPE Aruba Networking redundant switch pair, and it can be grown to a larger footprint with the option of expanding the back-end fabric in spine and leaves.

**High-density** modules come with the CBox in a 2U footprint in standard depth with 4 CNodes with-in the CBox doubling the form factor, south and northbound connections. For storage, the high-density modules come in a 1U DBox footprint in an extended depth rack with 4 DPUs, also referred to as smart NICs with-in the DBox, also doubling the form factor and connections to the back-end fabric. The back-end NVMe over Ethernet fabric comes in 64-port 100GbE NVIDIA® Spectrum SN4600 redundant switch pair, and it can be grown to a larger footprint with the option of expanding the back-end fabric in spine and leaves.

**Table 1.** Underlying building blocks comparison

| Platform component | Properties | HPE Alletra Storage MP in standard density | HPE Alletra Storage MP in high density |
|---|---|---|---|
| **Compute enclosure** | Enclosure | 2U enclosure with 2 nodes. Standard depth chassis | 2U enclosure with 4 nodes. Standard depth chassis |
| | Node | 1–32C CPU (AMD) 256 GB RAM, 2 dual 100G port PCIe4 cards | 2–16C CPU (Intel®) 256 GB RAM, 2 dual 100G port PCIe4 cards |
| **Storage enclosure** | Enclosure | 2U enclosure with 2 nodes. Standard depth chassis | 1U enclosure with 2 nodes. Standard depth chassis |
| | Node | 1 dual 100GbE port PCIe4 cards (4 ports per DBox) | 1 dual 100GbE port PCIe4 cards (8 ports per DBox) |
| | Capacity options | 154 TB (Raw) DBox with 8 TB drives, 308 TB (Raw) DBox with 15 TB drives & 550 TB (Raw) DBox with 30 TB drives | 338 TB (Raw) DBox with 15 TB drives & 1352 TB (Raw) DBox with 60 TB drives |
| **Back-end NVMe-oF n/a** | n/a | 1U HPE Aruba Networking CX 8325 100GbE 32-port in a switch pair | 2U NVIDIA Spectrum SN4600 100GbE 64-port in a switch pair |

## Compute node architecture

HPE GreenLake for File Storage built on HPE Alletra Storage MP compute nodes are a vital part of the system. A single CBox compute enclosure consists of two compute nodes in the standard-density variant and 4 CNodes in the high-density variant. From every CNode, two 100GbE ports are connected to the back-end NVMe over Ethernet fabric, and the other two 100GbE ports are connected to the customer Ethernet or InfiniBand network. For client connectivity, in addition to the 100GbE connectivity, 40GbE, 25GbE, and 10GbE connectivity is supported through appropriate transceivers with OM4 LC cables or appropriate active optical cables. See the QuickSpecs for more details.

## DBox enclosure (for both SCM and NVMe)

In HPE GreenLake for File Storage built on HPE Alletra Storage MP, the JBOF DBox enclosure is composed of NVMe SSD drives and storage class memory (SCM) drives.

HPE Alletra Storage MP in **standard-density** DBox comes with 24 SFF drive slots. In every DBox, about 2% of the total capacity is reserved for SCM, which is used like NVRAM. This distributed SCM layer also happens to be the metadata store. In a DBox containing 7.68 TB or 15.36 TB drives, 4 drive slots are reserved for SCM drives, and the remaining 20 slots are for NVMe drives. In a DBox containing 30 TB drives, 6 drive slots are reserved for SCMs while the remaining 18 slots are for NVMe drives. Each DNode consists of one 100GbE two-port host bus adapter (R7C84A). Two 100GbE ports are connected to the back-end HPE Aruba Networking switches. This means each DBox has four 100GbE ports connected to the back-end switches. Visit the QuickSpecs for detailed system building blocks.

HPE Alletra Storage MP in **high-density** DBox comes with 22 ruler-based NVMe SSD drives and 8 SCM drives. In every DBox, about 2% of the total capacity is reserved for SCM, which is used like NVRAM. This distributed SCM layer also happens to be the metadata store. Unlike in standard density, the NVMe SSDs and SCM drives are reserved regardless of the drive capacity. The DBox with 15 TB NVMe drives has 1.6 TB SCMs, and in a DBox with 60 TB NVMe, SSD drives have 3.2 TB SCM drives. Each DPU in the high-density DBox consists of one 100GbE two-port host bus adapter. Two 100GbE ports are connected to the back-end NVIDIA Spectrum SN4600 switches. This means each DBox has eight 100GbE ports connected to the back-end switches. See QuickSpecs for detailed system building blocks.

### Front-end connectivity

HPE GreenLake for File Storage supports Ethernet and InfiniBand for front-end connectivity on both standard and high-density variants of HPE Alletra Storage MP platform. HPE GreenLake for File Storage can be configured with all Ethernet options on all the CBoxes or all InfiniBand on all the CBoxes, or it can be configured in mixed mode with Ethernet and InfiniBand, and the selection is made at the CBox level.

HPE GreenLake for File Storage built on **standard-density** modules comes with 2 options for front-end connectivity. Solution with all Ethernet configurations should order R7C84A adapters while solution with all InfiniBand and mix of InfiniBand and Ethernet should order S2H33A adapters. HPE GreenLake for File Storage built on **high-density** modules on the other hand comes with dual-purpose Ethernet/InfiniBand adapters.

In essence, the selection available through OCA to select the front connectivity option for standard-density and the high-density CBoxes by default are shipped with dual-purpose InfiniBand/Ethernet capable adapters, which offer the choice between InfiniBand and Ethernet connectivity at the time of installation.

## Software architecture

Several features set the HPE GreenLake for File Storage solution apart from other scale-out NAS offerings. DASE is a unique software architecture that enables you to start small and scale independently of capacity and performance to a scale that was not possible with shared-nothing systems. This solution offers data reduction technologies that are unmatched in the industry with several unique techniques.

### DASE architecture

DASE architecture design is a giant leap in the world of distributed systems. This shared-everything architecture offers revolutionary benefits. The following section describes some of the key differentiating characteristics.

#### Designed to scale linearly

Typically, shared-nothing architecture relies heavily on the ability to scale up each controller node to achieve high performance, which means that the ownership of data is divided among controller nodes. This architecture might have made sense on a smaller-scale system to help guarantee high performance, but in the age of **data explosion**, scale-out systems start to reveal some of the shortcomings of shared-nothing architecture. Specifically, shared-nothing architecture requires east–west traffic, which is considered overhead, and which increases rapidly as the number of controller nodes increases. DASE architecture shines by helping eliminate all these shortcomings. Because there is no east–west traffic, latency does not suffer from the overhead as you scale up. You can add CBoxes independently to gain more performance, and you can add DBoxes independently without the need to add compute because, in this solution, storage ownership is no longer tied to controller nodes or node pairs.

#### No east–west traffic

If the controller nodes have too many shared states, shared-nothing architecture might suffer from the contention of resources among controller nodes. This means there is a lot of crosstalk between servers. As a side effect, server rebuilds take significantly longer and might affect the overall system performance and reliability. However, the same contention does not exist with the DASE architecture because all CNodes have the same paths to the entirety of the data structure. All CNodes in the same cluster can operate and serve the I/Os in the same manner. As a result, I/Os can be distributed evenly among all CNodes without causing contention. Therefore, in this architecture, compute node failure does not require a rebuild.

### Ephemeral compute

In the HPE GreenLake for File Storage solution, CNodes are stateless because the DASE architecture enables the compute layer to be fully independent of metadata, which is always stored in a persistent distributed SCM layer (also called NVRAM) in the DBoxes. All CNodes in the file cluster can see every drive in the cluster as if they were directly attached. All the CNodes also have access to the distributed SCM layer where the metadata resides. There is no crosstalk between the CNodes, no server rebuilds upon failure, and so on. The completion of an I/O helps guarantee that the data has been committed to persistent storage (SCM). This design alleviates the pressure of the cache and memory flush during a component failure to prevent any data loss or inconsistency in the data structure. Various mechanisms such as atomic updates, modifying tree structures from the bottom up, locking, and time-outs are in place to help ensure the consistency of the data structure while access to the data structure is being shared among CNodes.

Table 2 summarizes the high-level points of comparison.

**Table 2.** A summary of comparisons between shared-everything architecture and shared-nothing architecture

| Challenges with shared-nothing architecture | Characteristics of shared-everything architecture | Benefits of shared-everything architecture |
|---|---|---|
| Latency suffers from an overhead increase on scaled-out systems. | There is no east–west traffic among CNodes. | Scaling CNodes does not add performance overhead; latency does not suffer, and throughput scales linearly as the system scales out. |
| There is contention for resources among CNodes. | CNodes have the same access to the entire data structure. | There is minimal resource contention from CNodes. |
| State persistency (always maintaining cache coherency) is difficult to achieve. | CNodes have no persistent state, see the same persisted global state, and never have to maintain any cache. | There is no data loss or data structure inconsistency at CNode failure, and compute node failures are far easier to handle. |

## Service-centric software design

The HPE GreenLake for File Storage software stack can be simplified into three layers:

– Firmware

– OS

– Docker engine for containers

With firmware to provide communication to the HPE Alletra Storage MP platform and an OS to host minimum kernel modifications and packages on top of a standard Linux® OS, the brain of the operations resides in the containers running on a container orchestrator. By taking advantage of the benefits of containers, the software stack can provide relatively independent service to each functionality. The result is greater efficiency, consistency, and agility.

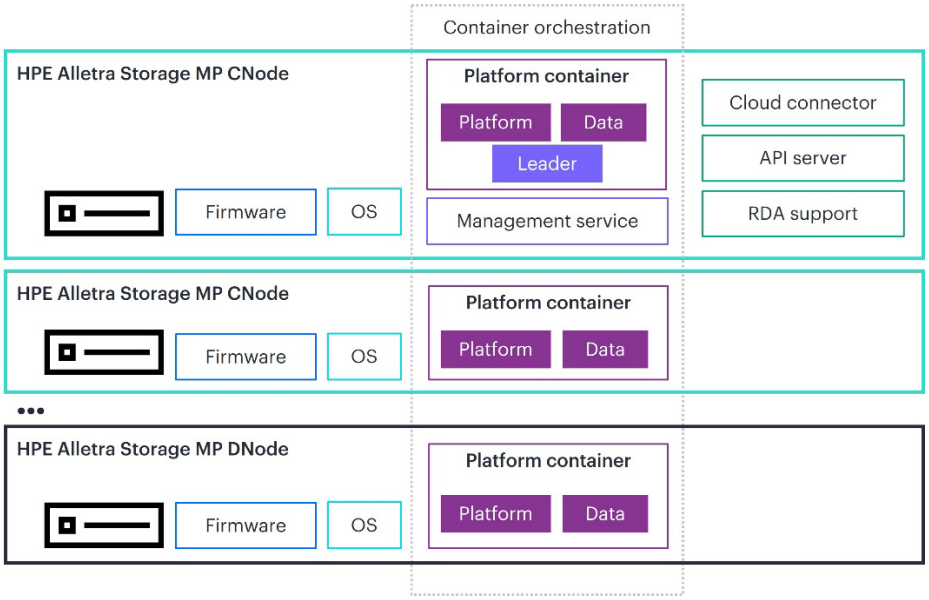


**Figure 1.** Software architecture block diagram

There are two containers (Figure 1) to note:

– The platform container

– The management service container

The platform container runs on every CNode as well as on every DNode; however, these services behave differently on CNodes and DNodes. The platform container on the CNodes runs most of the logic for the I/O operations, whereas on the DNodes, it functions mainly to expose the storage to other services.

A cluster has only one instance of the leader process at any given time, and the leader process runs on a CNode to make critical decisions for the cluster, such as determining which CNode the management service container runs on. The single-instance mechanism of the leader avoids complexities with voting and consistency. Other processes inside the platform container perform essential functions to service the I/Os for file services, such as erasure coding. The management service container controls management activities, such as taking snapshots, local management interfaces, upgrades, and so forth.

Finally, there are a few additional processes to note:

– The cloud connector, which establishes and maintains all communications on a tunnel connection to HPE GreenLake cloud

– An API server

– An RDA service for remote support

## File system, protocols, and cross-protocol

The file system is designed to generalize and combine the attributes of different file systems into a single data tree structure called an **element store**. All file operations through different protocols, such as NFS and SMB, are mapped to operations on the element store. This design not only easily provides cross-protocol functionality (because the underlying data structure is shared and is in the same namespace), but also offers the same level of data security, data reduction, and visibility to the metadata across all protocols.

Because of the single data structure for the entire storage space (also called a global namespace), there is no isolation of data; different protocols can expose different or overlapping portions of the element store by attaching the protocol-specific settings (view policies) to a directory path. The policy-attached portion becomes a mountable file share, also called a **view** (see Figure 2).

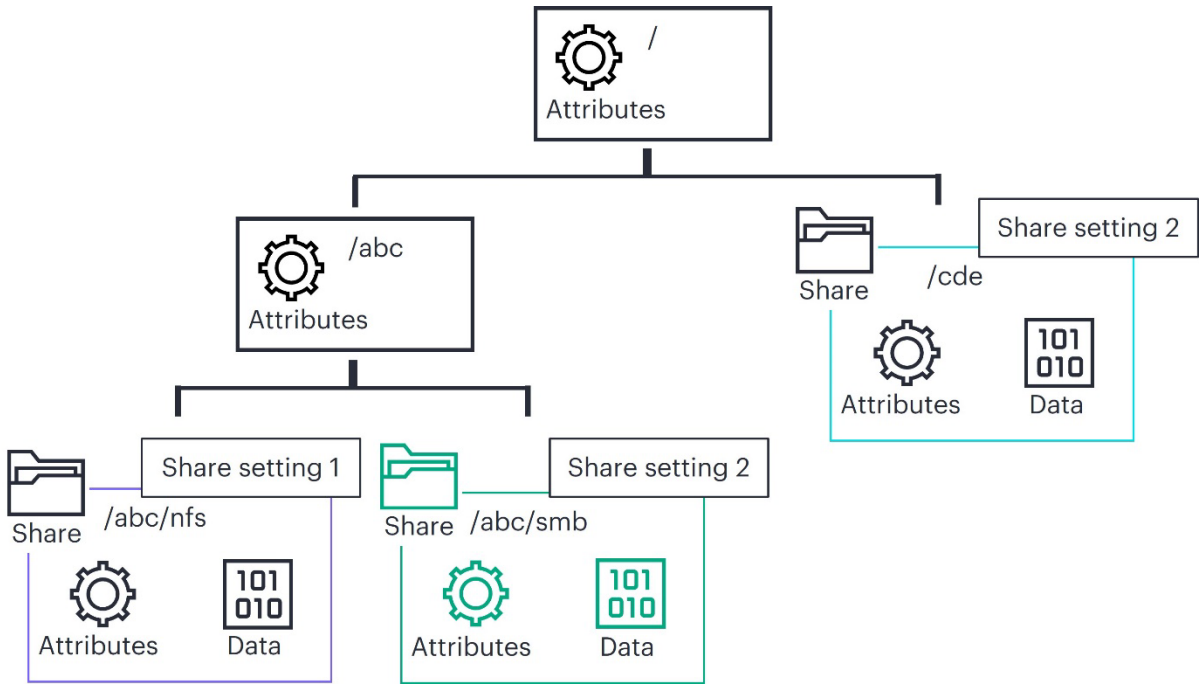


**Figure 2.** An example of the tree structure of an element store (file system)

Some key rules of file share can be defined in share settings, which then can be associated with multiple file shares. Key rules include which protocols are allowed to set permissions for the share (security flavor), authentication source (group membership source), **virtual IP (VIP) pools** that are available for the shares to choose as access points, and other protocol-specific settings.

File or directory count limit is bound by the number of DBoxes and the SSD drive types. The following table has the number of files supported per each variant of DBox we support.

**Table 3.** Table listing max file count per configuration

| Configuration | 7 TB SSD-based DBox | 15 TB SSD-based DBox | 30 TB SSD-based DBox | 60 TB SSD-based DBox |
|---|---|---|---|---|
| Standard density (Files per DBox in millions) | 760 | 1520 | 2280 | n/a |
| Standard density with "many files" enabled (Files per DBox in millions) | 1442 | 2850 | 4180 | n/a |
| High density (Files per DBox in millions) | n/a | 1900 | n/a | 5130 |
| High density with "many files" enabled (Files per DBox in millions) | n/a | 3800 | n/a | 5130 |

The file and directory count limit scales linearly with the number of DBoxes in the system. For workloads requiring significantly more file count support, there is an option to enable a feature called many-file support, which will double the file count of the said configuration. It is important to note that the many-file support is enabled during the time of installation.

### Global namespace

As a newly introduced feature available in v3.1, global namespace is a path-based access of data across multiple clusters for both read and write access. It warrants a strong consistency for reads and writes. An origin cluster in the context of global namespace feature refers to the cluster where the datapath is created, and satellite clusters are where the global namespace configured datapath is also available for read and write.

This is a typical I/O workflow of data access to a global namespace configured path. A file share is first created on the origin cluster, and the data can be accessed by clients through a regular datapath.

1. When the global namespace feature is configured on the same datapath, the same data is available on a satellite cluster for clients read/write access as well.
2. When a client requests access to a file in the global namespace enabled file share on a satellite cluster, the satellite cluster will obtain a copy of the file and a read lease from the origin cluster. For the first time of accessing the file, it typically takes longer (depending on the network latency between the origin cluster and the satellite cluster), but once the first copy of the file is obtained on the satellite cluster, any subsequent reads of the file share content will be much faster because the content will be directly accessed from the cached version of the file on the satellite cluster locally as long as the read lease is valid.
3. Either after a configurable fixed amount of time is reached or when a write I/O is issued on the file(s), the read lease expires, and the cached content of that file(s) becomes invalidated.

When considering using global namespace, understanding its limitations is crucial. Replication and global namespace cannot be used together on the same datapath. Global namespace is only available for NFSv3 access on satellite clusters; however, the global namespace enabled file shares can still be available for other cross-protocol access on the origin cluster. Write performance can be hindered by the latency between the origin cluster and the satellite clusters. This feature is not considered a data protection enhancement feature yet because the data is only enabled to be available on the origin cluster. Specifically, if the origin cluster is unreachable, the satellite clusters will not serve data for that path; if a satellite cluster is unreachable, the origin and other satellite clusters will function as normal.

## Performance

The HPE GreenLake for File Storage solution is designed to deliver high performance that is predictable, reliable, and scalable to an unprecedented degree. It also delivers linear performance as you scale—independently of compute for performance or storage for capacity, and without flat overheads. Because it overcomes two problems of traditional systems built on shared-nothing architecture, this capability is unique for two key reasons:

– The architectural limitations of traditional systems result in a flat overhead because, after a certain point, the system grows too complex to manage effectively and becomes unstable.

– These architectural limitations force scaling to be done in combination with compute and storage.

The following sections describe the key elements of the HPE GreenLake for File Storage architecture with a focus on performance and scalability.

### Datapath

The datapath for a write I/O is unique compared to other network-attached storage (NAS) solutions because it is designed to overcome the common NAS challenges of achieving data consistency with volatile memories. Figure 3 shows the write datapath.

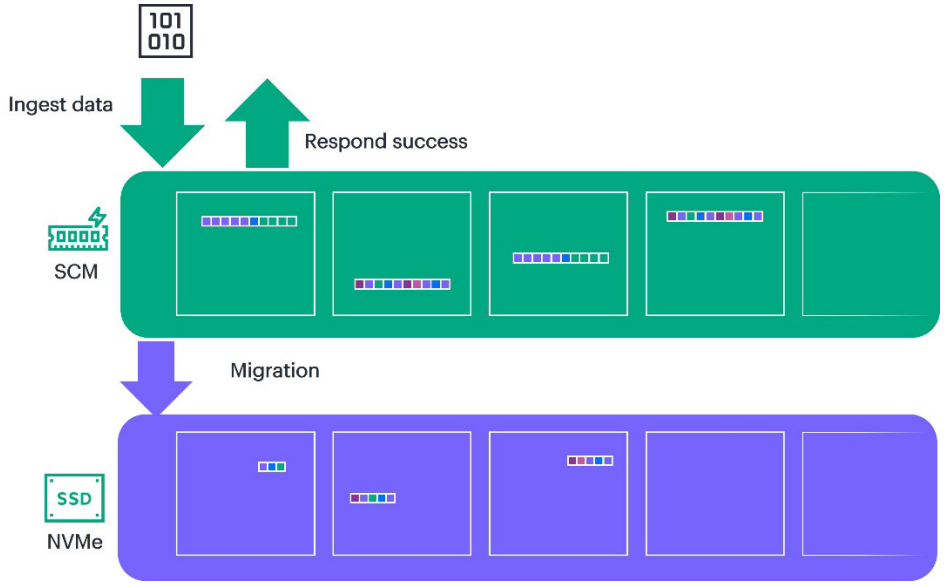


**Figure 3.** Write datapath

Because write I/O originates from a client through any standard NAS protocol and reaches any of the CNodes through one of the VIP pools of mount point IPs, the CNode cannot acknowledge the write request as successfully completed until the following three tasks are performed:

1. Data is written to multiple SCM drives (Write buffer RAID or mirrored) so that no data is lost in the event of an SCM drive failure.
2. Data is sharded randomly across multiple SCM drives to increase throughput and decrease contention.
3. Metadata is updated in the internal data structure (tree) for consistency.

This process helps ensure the consistency of data. After the write I/O is completed, the written data is located on persistent storage (SCM) with redundancy, which means the chance of data loss is extremely low.

In addition to these immediate tasks of completing a write I/O, the process of migrating data from SCM to SSD is also required. This migration is an out-of-band process. It is performed by all CNodes in the cluster, and it occurs at its own pace. Migration occurs mostly at idle times or slower times from clients. The process becomes more aggressive as SCM fills. When necessary, however, CNodes throttle the responses to the client's write requests with the goal of slowing down the write throughput to help the migration keep up with the data ingestion rate. This problem can be solved by expanding DBoxes, which increases the distributed SCM layer. Data is always preserved in SCM until safely migrated to SSD. The total cost of a write I/O (including the cost of migration combined with the original cost of a write I/O to SCM) includes the following steps:

1. Data is written to SCM (Write buffer RAID or mirrored).

   Consider that all the following steps occur only during migration from SCM to SSD after the write requests have already been acknowledged as successful to clients.
2. Data is read into DNode RAM.
3. CNodes CPU cycles for the migration process, which entails data reduction computation and so forth.
4. Data is written once to SSD.

The read datapath is more straightforward than the write datapath. After a CNode receives a read request from a client through any standard NAS protocol, it traverses the SCM metadata to find the data location. The CNode then retrieves the data from SCM or SSD and returns it to the client. Because CNodes have the same access to SCM and SSD, the performance of the read I/O is not affected, whether the requested data is on SCM or on SSD.

## Controller Pooling, quality of service, and multitenancy

HPE GreenLake for File Storage offers a powerful feature called **Controller Pooling** that enables you to designate all or a set of compute nodes to an application to help ensure that enough compute is reserved for the application. This is accomplished by creating VIP pools to distribute across all or a subset of CNodes. For example, imagine an application that requires all the resources it can get to meet its SLA and a VIP pool with 80–90% of its CNodes assigned to the VIP pool to which the application is assigned. The remaining 10%–20% of CNodes can be assigned for low-priority tenants such as backup or replication. However, if the app ever needed all available compute resources, there would be flexibility to expand its VIP pool to spread across all the nodes without disruption or tedious tuning tasks. In the event of a CNode failure, all virtual IPs would be moved over to the surviving CNode.

In this solution, the CNodes provide the front-end protocol service to the clients or applications, and all the intelligence, such as striping the data across the DBoxes or reading data from the DBoxes, runs on these CNodes. Nevertheless, CNodes are ephemeral in nature, meaning that they contain no state within this architecture, as explained in the DASE architecture section.

**Quality of service (QoS)** policies enable you to define fine-grained control of bandwidth and IOPS by individual share or per user with minimum and maximum limits. In addition to the minimum and maximum QoS limits, QoS policy can enforce a burst limit and a credit limit. When a burst limit is set, burst credits are accumulated if the workload consumes less resources than set by the maximum limit. The credits can later be spent to gain performance that exceeds the maximum limit, up to the configured burst limit. The amount of credit can be accumulated and capped by a credit limit. For instance, if a QoS policy defines a maximum limit of 100 MB/s, a burst limit of 1000 MB/s, and a credit limit of 10000, after 100 seconds of idle time, the credit limit will be reached. Given that credit balance, the application can run at 1000 MB/s for 10 seconds following, which will be throttled down to 100 MB/s.

QoS per view policy can choose a QoS provisioning mode and set static and/or capacity-based QoS limits. However, with QoS, per user view can set static QoS limits for one or more users.

In this architecture with a global namespace, data is exposed to the clients with one or more protocols that are defined and governed by share policies. These policies and shares can be associated with a construct called a **tenant** that enables rules to be set to segregate shares and share policies, making it possible to implement **multitenancy**. A tenant can have its own client IP address range to access its associated shares. This enables each tenant to have its unique identity providers, such as Active Directory, LDAP, NIS (not exceeding eight Active Directory providers, eight LDAP providers, and one NIS provider per system). Each tenant can have its own encryption keys for data-at-rest encryption.

Controller Pooling, QoS, and multitenancy are independent of each other. However, they can be used either separately or together. In essence, **Controller Pooling** enables coarse-grained control over the utilization of the CNode resources and bandwidth, and **multitenancy** enables fine-grained control over viewing and accessing data, whereas **QoS** enables fine-grained control over resource utilization.

### Enhanced NFS performance

The HPE GreenLake for File Storage solution offers enhanced NFS performance (Figure 4). In addition to the default NFS mount with single-socket connection between the client and the storage port, the multipathing feature is available for multisocket access through the Linux nConnect feature. With nConnect, on certain Linux distributions such as RHEL 8.3, Ubuntu 20.04, or newer versions, the nConnect mount option can be used to configure up to 16 TCP connections between the client and the single storage port (VIP pools on the file cluster).

In addition, the solution supports multipathing for NFS clients. This feature enables a client to open multiple connections from multiple ports to multiple addresses. It is supported for both versions, NFSv3 and version NFSv4.1, and it requires a package to be installed on the NFS client.

In addition, the solution supports NFS over RDMA, bypassing CPU resources to deliver enhanced performance. And to take it even further, GPUDirect Storage™ will take the performance to a whole new level as it will even bypass client-side memory to achieve up to a remarkable 170 GB/sec performance per host. These multiple connections and paths enable the solution to achieve enhanced NFS performance scaling.

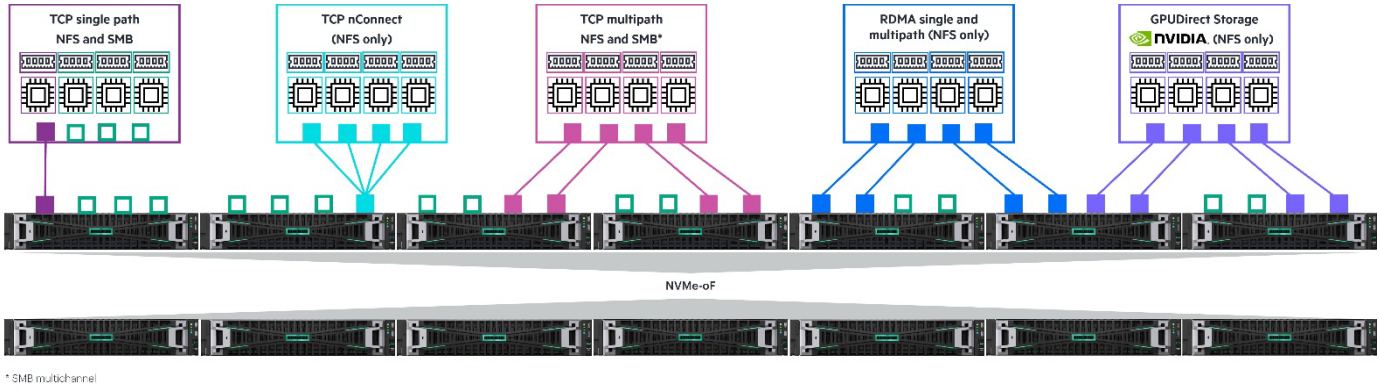


**Figure 4.** Enhanced performance use cases

## Storage capacity

The HPE GreenLake for File Storage solution offers capacity starting at approximately 220 TB of usable capacity with the ability to grow into a significantly larger footprint. The starting configuration is configured with a 100GbE switch pair (leaf) scaling up to a larger system with spine and leaves for the back-end NVMe fabric. The total capacity always excludes SCM drives. It is important to understand the terminology and definitions associated with capacity—particularly factoring in data reduction:

- **Raw capacity** refers to the total aggregate capacity across the NVMe drives without factoring in space reservation, erasure coding, or data reduction.
- **Logical capacity** reflects the data applications stored on the array.
- **Usable capacity** is the capacity that is available after calculating erasure coding and space reservation. It does not factor in any data reduction calculations.
- **Effective capacity** is the capacity after factoring in all data reduction calculations on the array.
- **Auxiliary capacity** is the capacity that is waiting to be released back to the usable capacity upon deletion of data from the client. Underlying snapshots of the deleted data keep the auxiliary space from being cleared.

### Data reduction technologies

Data reduction with HPE GreenLake for File Storage is a significant advancement in many respects. Over the past few decades, most, if not all, storage solutions have had deduplication and lossless compression on sets of data within the storage subsystem. HPE GreenLake for File Storage offers a unique approach to data reduction and applies these techniques across the single global namespace to achieve superior data reduction ratios.

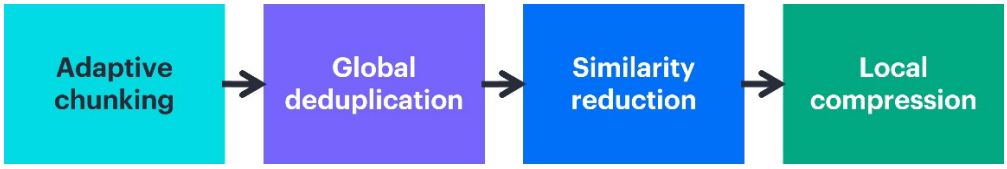


**Figure 5.** Data reduction workflow

Data reduction is an in-line process in HPE GreenLake for File Storage (Figure 5). During the data migration process (commonly referred to as destaging) from SCM to SSD, the data pipeline goes through a process called **adaptive chunking**. Then the global deduplication engine works to compare the data blocks with the entire namespace. At this level, the process of similarity reduction finds data blocks that are similar but not identical to reduce the data footprint even further. In the end, local compression kicks in. Even though the data reduction technologies used are all block-based reduction methods, there is not a fixed block size as in traditional block storage.

### Adaptive block size chunking

As an optional feature (on by default), adaptive block size chunking takes advantage of a sliding block size to help maximize the opportunity to perform the similarity reduction and deduplication. With an adaptive block size between 16 KB and 64 KB, the algorithm adjusts the block size based on its likelihood of best data reduction. As a result, a file that is edited or modified at a later state has very minimal impact on the block set of the file and still achieves the best possible chance of data reduction as opposed to fixed block chunking, in which the entirety of the file blocks are mismatched from their original state after an edit operation.

### Global deduplication

Each data block generates a cryptographic hash, and the hash values of all the data on disk are stored in SCM. Before the data is written to the NVMe SSD layer in the migration process, and after the data passes through adaptive block chunking, the data blocks are compared against all existing hash values on the entire namespace. If a match is found, the data is stored as a pointer to the duplicated block found on the existing data. This deduplication process can be far more effective than local deduplication because the shared-everything architecture gives all CNodes access to the entire data set without the tradeoff of CNode east–west traffic to take away CPU cycles.

### Similarity reduction

Instead of finding an exact match for a block to deduplicate, similarity reduction compares a calculated similarity hash to find blocks that have high similarity but small differences. If two blocks are sufficiently similar in content by having a collided similarity hash, they are stored as pointers to a deduplicated baseline block and byte-level deltas (Figure 6).

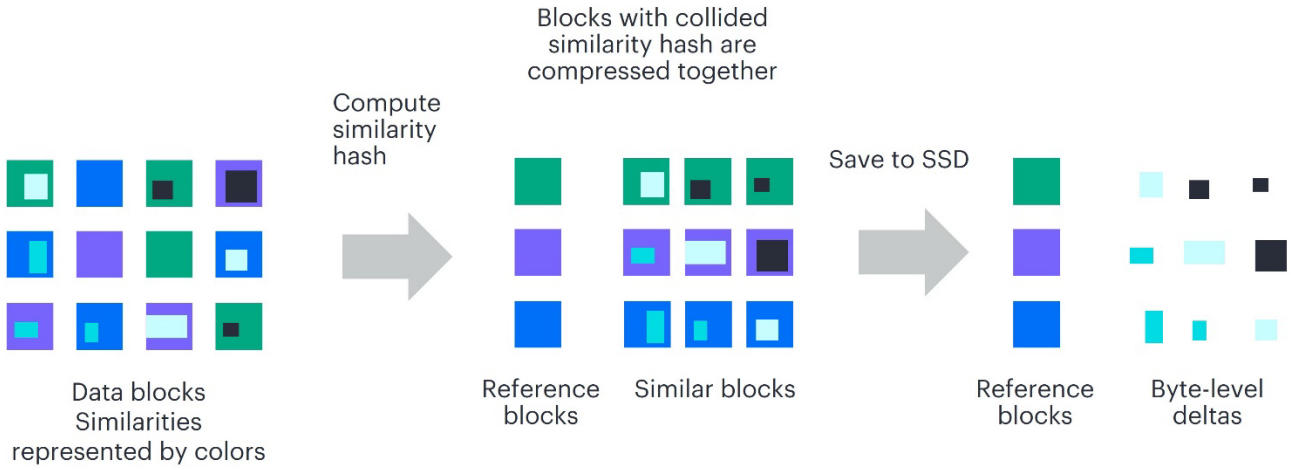


**Figure 6.** Similarity reduction

### Local compression

A lossless compression algorithm, Zstandard (ZSTD), is used to compress data blocks to reduce block size. ZSTD is a fast lossless compression algorithm that targets real-time compression scenarios and better compression ratios.

### Defragmentation

Thanks to the larger-sized striping to benefit performance, a single stripe might contain multiple files. Therefore, when data is updated or deleted, parts of the stripes become stale. A defragmentation process is in place to reclaim the stale space on those stripes when there is enough stale space on a stripe to make it worth the cost of the defragmentation process. The defragmentation process occurs only when needed. In other words, the fuller the capacity gets, the more aggressively the defragmentation process runs. This process has a greater impact on performance as the capacity gets near full and the process takes away more CPU cycles from other operations.

### Quotas

In a solution such as HPE GreenLake for File Storage that offers global namespace, it is trivial to have a mechanism in place to assign capacity limits granularly. This is where quotas come in. Quotas can be set at the directory level by capacity and/or by the number of files and directories that can be created under a share or directory. In addition, quotas can be set by user and by group, limiting the amount of capacity individual users and groups can use within a directory as well as the number of files and directories.

## Data resiliency and reliability

Data resiliency is built into every level of HPE GreenLake for File Storage, specifically in the write datapath, hardware fault tolerance, and disaster recovery features. This section of the white paper focuses on the technical details of how these capabilities help protect data.

### Write datapath

Two main capabilities that are built into the write path to the SCM layer help to maximize data resiliency: **Write buffer RAID**[1] or **mirroring** and **random sharding** depending on the system configuration. All SCM devices are split into 8 GB sections. When an inbound I/O arrives at a CNode, it requests a section address and is given the physical addresses of two SCM devices to which data should be written. All sections are either written in a RAID 6 format or mirrored depending on the system configuration. These section pairs are randomly and evenly distributed across the available physical SCM devices. The CNodes write directly to those SCM devices and acknowledge the write request to the client as successful after they have received confirmation that data is permanently stored in both locations. Data chunks within those sections might be of variable sizes, and data can be written to any logical offset of the 8 GB sections.

Committed data does not live in any form of nonvolatile memory, which alleviates pressure from the system to flush any type of dirty cache in the event of failure. After the data is written to the SCM, the probability of data loss becomes extremely small.

### Locally decodable erasure coding protection

Erasure coding on SSDs can be simplified to be understood as wide striping and parity redundancy. An erasure coding stripe width is the number of drives a stripe is spread across, and the value is defined as **D + P**. Parity (P) is always 4, which means that a maximum of 4 SSD drives failed simultaneously without data loss, but data (D) is approximated in the following way: Take the number of healthy SSD drives and subtract 4. D equals the lesser of 146 and the number calculated from the previous step. The overhead percentage can be approximated by calculating P/D. The wide striping decreases overhead percentage and thus is more advantageous on a larger configuration. Larger configurations can take advantage of an overhead as low as 2.7%.

The design of having no fixed grouping size enables easy integration of varying drive sizes and expansion, as well as easy failure handling for drive failures. As the solution expands (for example, in a scenario of adding more DBoxes to the existing configuration), the new configuration determines the new larger stripe size, and the previously written existing stripes with the smaller stripe size are gradually copied and rewritten into the new stripe size that goes across the new configuration.

[1] Write Buffer RAID feature currently only applies to newly installed high-density variant of HPE GreenLake for File Storage.

Although the erasure coding algorithm used in this solution provides rigid data resiliency with the wide striping, the locally decodable aspect of the algorithm offers the additional advantage of being able to reconstruct a corrupted fraction of the stripe without the tradeoff of needing to read across the entire wide stripe. It is the case that **1/X** of the data stripes can be locally decodable, where X is the number of protection stripes in the set. For a typical **nD+4P** encoding, only **1/4** of the data must be read during reconstruction.

## Hardware fault tolerance

In most cases, multipoint failure is tolerated at every level of the hardware components, with only a few exceptions. The following list details each of the hardware components:

- **SCM:** Generally, one SCM device failure can be tolerated without data loss or disruption to servicing I/Os. During an SCM device failure, the leader process instructs all the CNodes to rebuild all affected 8 GB sections on the failed SCM device in parallel. Any read request in the meantime is fulfilled by reading from the replica. Although these 8 GB sections are evenly distributed and the rebuild traffic is highly parallel, there is still a performance penalty during the time of the SCM rebuild. In typical production usage, the SCM rebuild takes 30 minutes or less, and in the worst-case scenario in which the SCM devices are at 100% full capacity, the SCM rebuild takes up to an hour. If the second SCM device failure does not occur before the SCM rebuild completes, there is no data loss; however, a failure of more than one SCM device might lead to data loss.
- **SSD:** During a single SSD drive failure or for up to three simultaneous SSD drive failures, locally decodable erasure coding allows reading only 1/4 of the data set of the stripe to recover the fraction of the stripe from the corrupted SSD drives. During the time of reconstruction, no front-end performance penalty is incurred. However, if four SSD drive failures occur simultaneously, the system must read 100% of the blocks from the stripe to be able to reconstruct corrupted data from the failed drives. During that time, there is a performance penalty for the reason of expediting the data reconstruction to help minimize the chance of data loss.
- **CNode:** DASE architecture allows an unlimited number of CNode failures if there is one surviving CNode. All the CNodes have the same access to the entire storage space and metadata pool, and they are stateless/ephemeral in nature. Surviving CNodes can resume the workloads without I/O disruption by leveraging the VIP pooling mechanism.
- **DNode:** DNodes operate in HA pairs and expose the SSD and SCM to the CNodes. Each SSD and SCM supports duo ports, and duo port drives can be accessed by the surviving node on the same DBox enclosure during a DNode failure. If each DBox enclosure keeps at least one DNode functional during a DNode failure event, there is no disruption to the client. The performance might be impacted by loss of NICs and PCI bus throughput on the failed DBox.
- **CBox:** The CBox has the same level of fault tolerance as CNode failure. The system remains functional as long as one or more CNodes survive from any of the other CBoxes in the cluster.
- **DBox:** A DBox failure is highly unlikely because the DBox is built with redundant components. However, in the unlikely event that a failure results in an outage, no data loss occurs because of a DBox failure. To recover from a DBox failure, all drives must be removed from the failed enclosure and reseated in a healthy replacement DBox. In a future release, the system can be configured with a higher number of DBoxes with DBox redundancies.
- **Data network:** Each CNode and DBox has redundancy with ports and connectivity to the internal switches as well as to the ToR switches within the customer network. If enough ports or NICs failures occur to make a node or enclosure unreachable, such a failure can be treated the same as the node or enclosure failure.
- **Management network:** In addition to interrupted manageability (from management service, SSH, and so on), failure of the management network might also cause these functions to fail. Failure would result in an inability to send outbound call-home bundles, loss of registry access, which entails change of user population and cluster access interruption, and loss of access to DNS, which might cause break mounts.

## Online upgrades

Software upgrades are online upgrades, and they are implemented in a rolling fashion. For most of the configurations, one out of N nodes would be offline at some time during the upgrade of that node. For larger configurations, multiple nodes would be upgraded simultaneously. Generally, the order of the upgrade is to upgrade the management service container first and then to upgrade software on all of the DBoxes, one at a time. After that, software on all of the CNodes is upgraded one at a time except for the management service hosting node, which is upgraded last.

Not every upgrade requires a node reboot. No system-wide downtime is required for upgrades, and only minimal performance impact can be observed during an upgrade. Firmware upgrades do not require system downtime. The duration of an upgrade process varies based on the size of the configuration. For a configuration of 12 or the smallest configuration of a single CNode and a single DBox, a typical software upgrade without an OS upgrade takes about 30 minutes.

## Snapshots, retention, and immutability

Snapshots can be used as an effective method of local ransomware protection. When a snapshot is taken for any given directory tree, a consistent point-in-time record of the data on that path is created. Generally, at the time the snapshot is taken, there is no capacity usage or performance impact. The capacity cost of the snapshots is incurred when data is deleted, and the usage is reflected in auxiliary capacity.

When snapshots are deleted, there is little or no impact on performance, with only two exceptions:

– If a snapshot is deleted that contains a large amount of deleted data

– If the various deleted snapshots contain many small modifications to the same file

In both cases, the defragmentation process for cleaning up a large amount of stale capacity usage or an increased amount of small random I/O during the defragmentation might temporarily impact performance.

Snapshots can be taken at the directory level, either manually or as scheduled through protection policies. For snapshot auto deletion, snapshots can be configured to be retained for a set time with protection policies. Snapshots can also be nested. Snapshots can be set to become immutable to prevent deletion, and immutable snapshots can be deleted, modified, or have their retention period reduced only when HPE Support is engaged and the issuer's identity is verified.

## Replication

Replication can be configured from one HPE GreenLake for File Storage system to another. Protection through native peer replication can be configured per directory and can be integrated with snapshots. At initial release, asynchronous replication is supported with RPO to be as low as 15 seconds and up to a few minutes.

This is the workflow to protect a file share through peer replication:

1. Set up two systems to establish a peer-to-peer replication relationship, and during the process, optionally allocate a VIP pool specifically for replication connections.
2. Create a protection policy to define attributes of a replication setup, such as RPO, replication starting time, and retention periods of the local and replicated snapshots.
3. Assign the protection policy to any directory path to protect the data through replication.

Replication can be highly parallelized, depending on the VIP pool assignment. Data being replicated over the wire is compressed, however, not deduplicated with similarity reduction or global deduplication. In other words, if two copies of the same file exist on the source system, the same file is sent twice through replication. Then it is further deduplicated at the target system.

After the replication process starts, the replicated data on the designated path on the target system is not automatically attached with a share setting; therefore, the replicated data is not yet available as a mountable file share. For it to be exposed to any client as a mountable file share, a manual process must assign a share setting to the designated path on the target system. The destination file share on the target system has read-only access to all clients, although the source file share maintains read-write access to clients on the source system, as shown in Figure 7. Bidirectional replication is supported as long as the direct paths of both replication directions do not overlap.

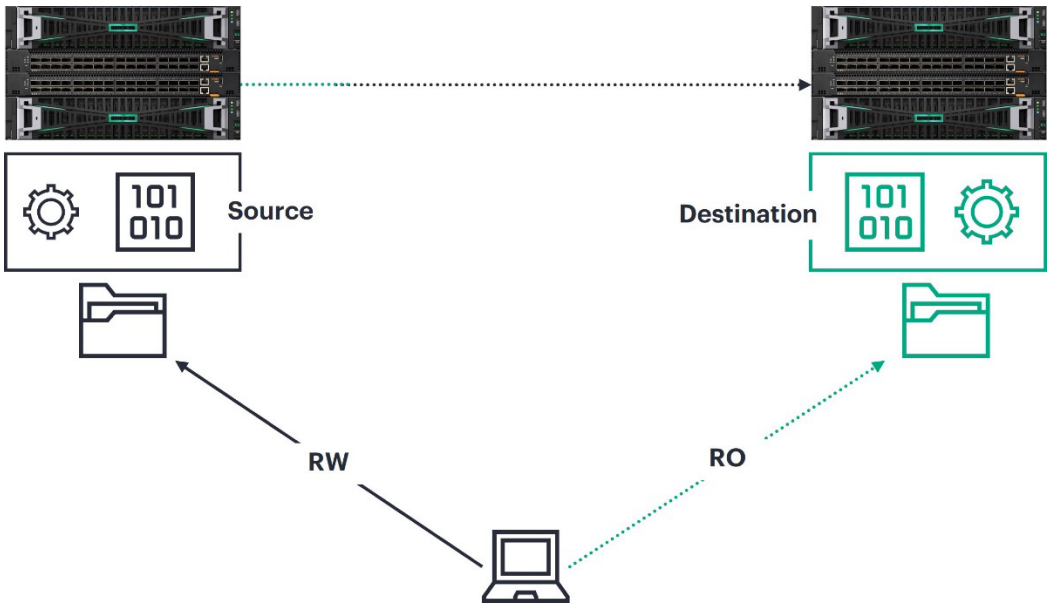


**Figure 7.** Starting status of a peer replication setup

Two types of failovers are available: **graceful** and **ungraceful** options. Graceful failover can be initiated only from the target system for each protected path, in which case, you can fail over a file share without failing over an entire system. After a graceful failover is triggered, the system follows this process:

1. The source file share becomes read-only.
2. A snapshot is taken at the source directory.
3. The previously read-only destination file share on the target system becomes read-writable with the latest data.
4. The replication direction is reversed, and future snapshots are taken from the former destination file share.

Figure 8 shows what happens after a graceful failover process is completed, with the resulting replication direction and the client access. Graceful failback is the reverse of the graceful failover process, and the resulting statuses are the same as they were before the failover occurred.

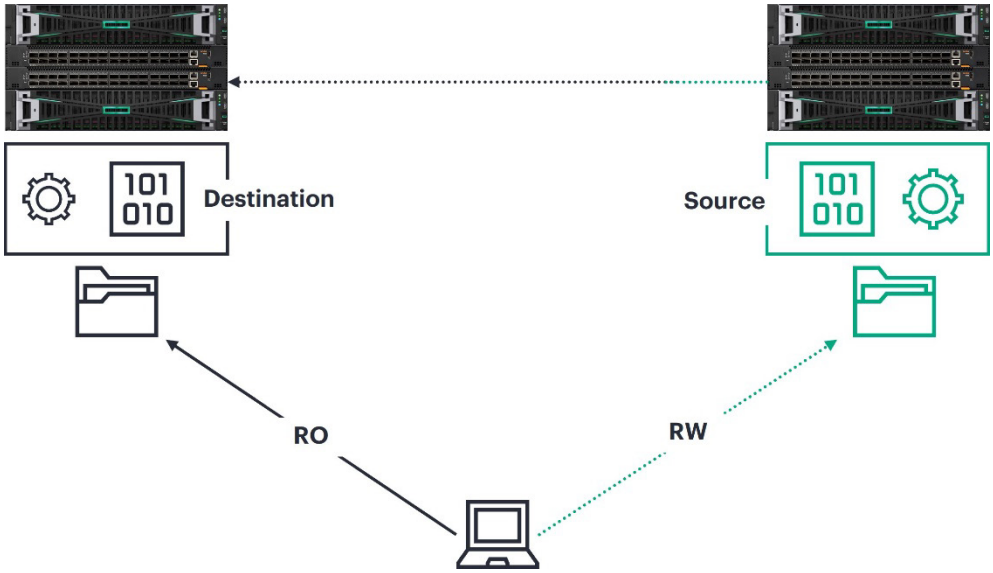


**Figure 8.** Peer replication post-failover state

Ungraceful failover immediately makes the destination file share on the target system writable and terminates replication. Clients can obtain write access to the replicated data only if the share setting is configured and assigned to the replicated directory. Snapshots are captured and obtained locally on source and target systems independently. Failback after an ungraceful failover requires extra consideration and preparation because, during the failover, both copies of the file share on source and target maintain write access to clients. In this case, the content of the replicated copy on the target system might diverge from the content on the source system. After the failback, the file share on the target system becomes read-only and loses the delta of changes that occurred during the ungraceful failover. However, the diverged content is preserved as snapshots on the target system.

### Global Snapshot Clones

Global Snapshot Clones gives clients instantaneous write accessibility to a snapshot of the data on a remote site, usually after 10 seconds or less from creation of the Global Snapshot Clone (Figure 9). It can be created from a snapshot on either a replication peer cluster or on the local cluster. Global Snapshot Clones can be configured to replicate using 2 modes, the first one being the background sync on, in which case, the snapshot data is to be copied from the source to the destination right after the clone is created. During the copying stage, read requests are directed to the source if the requested data is not yet available on the remote site; the second mode is the background sync off, which means the snapshot data is not copied to the destination except when there is a request to read data.

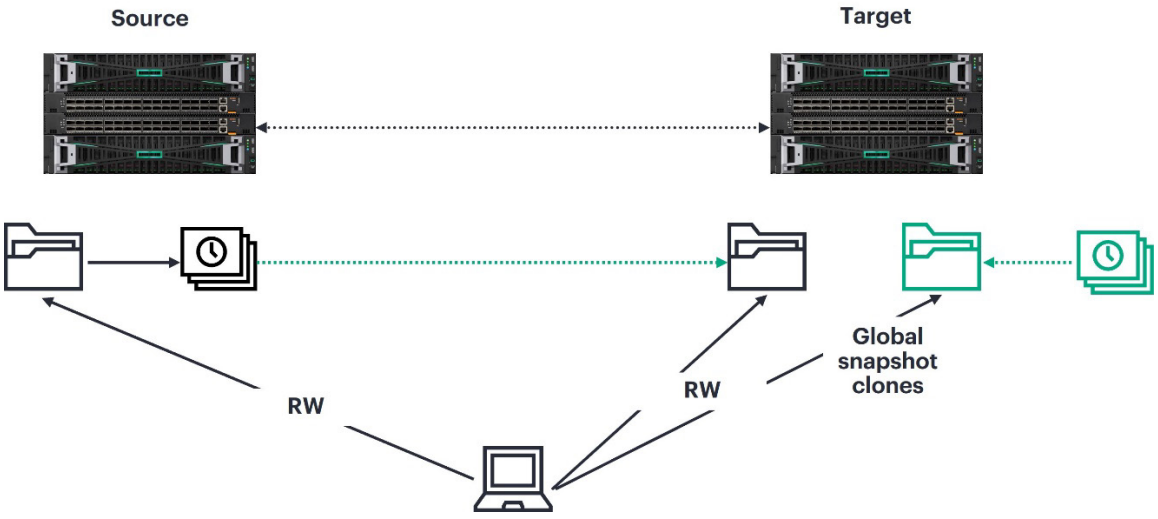


**Figure 9.** Global Snapshot Clones diagram

#### Group replication

Group replication allows for 1:N replication on the same datapath with different policies and schedules for multiple remote sites. The data is intelligently resynched during a failover event using common resync points, which allows for significantly faster access to the latest version of the data on the remote site after a failover. Resync points can be explicitly defined by users or can be implicitly defined by overlapping schedules.

This is a typical datapath of a group replication configured:

1. A group replication is configured to replicate from the source cluster A to 2 remote sites B and C. Data is replicated to B on an every-hour schedule while the same data is also replicated to C on an every 30-minute schedule.
2. During a failover of cluster A, cluster B becomes the new source cluster and thus the data can be accessed through reads and writes. During the failover from cluster A, the data on cluster B only needs to perform a partial resync with cluster C for the deltas between the latest available copies on both clusters. The replications from cluster B will continue to replicate from B to C and to A if A is reachable.
3. If the failover is to set the remote site to be in a stand-alone mode, then after the failover, cluster B will no longer replicate to C or A.

## Security

HPE GreenLake for File Storage contains a rich set of security features to enhance protection from various perspectives, including authentication, encryption, auditing, and multitenancy for segregated data access, as well as the rigorous security measures equipped with HPE GreenLake manageability.

### Data authorization and authentication

All data access is controlled, maintained, and stored through an internal user database. Access bits are exposed as POSIX mode bits, POSIX ACLs, SMB ACLs, and NFSv4 ACLs. All external providers and local users and groups, such as UID, GID, SID, and so on, are mapped to internal IDs within the internal user database. The internal user database caches for each user and is periodically refreshed from configured external authentication providers, such as AD, NIS, and LDAP, through the management network. The internal user database lives in the SCM layer and is mirrored and sharded across all available SCM devices. Remember that the scalability of caching a large number of user entries from external providers is bounded by SCM capacity.

### Data-at-rest encryption

All user data and metadata on an HPE GreenLake for File Storage system can optionally be encrypted at rest, and this option can be configured at the time of installation. The encryption keys are maintained internally. The data-at-rest encryption feature is agnostic to all file protocol clients.

### Auditing

Auditing can be configured for protocol specifics, such as NFS or SMB, as well as types of file operations to audit, such as create, delete, and so on. A management services activities auditing feature is also available to monitor system-wide management activities. Capacity for auditing can be capped per CNode. Retention can also be configured to be a fixed amount of time and provides the option of keeping the audit files forever. The performance and capacity impact might vary depending on the level of auditing configured.

### Multitenancy

The multitenancy feature of HPE GreenLake for File Storage also provides additional options for data segregation and data domains. For more information, refer to the Controller Pooling, QoS, and multitenancy section of this white paper.

### Cloud-based management security

Cloud-based management from HPE offers many advantages for both data infrastructure and the data itself. Data Services Cloud Console is the HPE cloud-based application for current and future data and storage management.

Security is at the center of the design of Data Services Cloud Console. HPE developed it and its associated products to meet customers' security needs. For more information about all security aspects of the HPE GreenLake as well as the Data Services Cloud Console, visit the Data Services Cloud Console Security Guide white paper.

### Other security features

In recent updates with v3.1, few security features have been introduced.

First is NFSv4.2 security label, which is a toggle that can be enabled or disabled on a per-tenant basis. The NFSv4.2 security labels are designed to work with SELinux to provide mandatory access controls that are stored as extended attributes. This feature is only applicable when SELinux security labels are required.

The second security feature introduced in v3.1 is the attribute-based access control (ABAC) as part of the zero trust HPE Alletra Storage MP features. The ABAC feature realizes access control based on attributes associated with files or folders and separates the access control from the storage admin but relies on the authentication provider instead. ABAC tags can be selected per file share, but they won't be evaluated against authentication provider at the time of creation and cannot be changed later once created. Users with ABAC tags are refreshed much more frequently, every 60 seconds or manually. ABAC tags are evaluated before and in conjunction to the file system permissions and ACLs, and the most restrictive permission combination prevails. Similarly, with multiple ABAC tags, the most restrictive combination wins.

The third feature is the write-once, read-many (WORM) feature that is available for NFSv3 and SMB file shares. There are two ways to define the retention period on files for WORM, due to the fact that for file workloads, it is often difficult to determine when a user has finished editing a file. The first way to configure a file or folder to transition to read-only mode is by assigning a future time as the time, which means as long as that retain until date isn't reached, the file will remain writeable. The second way is the auto-commit to WORM mode, which sets a file share level retention time, so that once the auto-calculated time since creation is up, the file(s) and folder(s) will become read-only.

## Storage management

The HPE GreenLake for File Storage user interfaces offer a simplified experience for storage admins to manage file workloads. Data Services Cloud Console provides simple file data management with an intuitive cloud experience along with the capabilities you need to manage your HPE infrastructure and workloads (such as block, file, backup, and recovery) all in one place. Local administration UI is also available for managing individual arrays with a GUI.

Some terms used in the onboard UI as well as other Data Services Cloud Console terms might differ from commonly used NAS terminologies. These terms are summarized in Table 4 in the Appendix.

### Data Services Cloud Console

Data Services Cloud Console is an application offered on the HPE GreenLake for all your data services needs. It mitigates management complexity by removing infrastructure management silos. Data Services Cloud Console provides a cloud-managed single console accessible from anywhere and from any device. Within Data Services Cloud Console, there are two services for managing your file storage:

– **Data Ops Manager** enables you to manage your organization's variety of storage platforms, including but not limited to HPE Alletra Storage 6000, HPE Alletra Storage 9000, and HPE Alletra Storage MP. You can use Data Ops Manager to view system inventory and details. From the system list, you can filter, search, and sort systems with both storage arrays for file and block. From the dashboard, you can find an overall array status, capacity, and performance. Besides general array information, the most helpful detail is found on the system-centric tabs. For example, the issues panel highlights anything critical that requires the storage administrator's attention. It is easy to see the capacity and performance of the system by browsing the historical metrics. Data Ops Manager is also where the storage administrator can modify system-level configurations such as creating or modifying VIP pools, users and groups, replication, network settings, and so on.

– **The file storage application dashboard** provides a unified and aggregated view across all your HPE GreenLake for file storage systems to include all the performance, capacity, and important metrics you need to know about as well as storage consumption details. Each tile highlights the top-ranked file shares or clients for any potential hot spots to focus on. From the file share list, you can view shares based on characteristics such as protocols, protection policies, and storage arrays, or you can choose to view them in a performance-centric view. In this view, you can view and filter file shares based on usage, file count, quota, and performance metrics. As you drill into the details of each file share, you can see the capacity, performance, user and group accounts, and protection details.

In the file storage application, you can create file shares as well as file share settings. For the general settings, you can define parameters such as the file server, which hosts the shares to which the settings will be applied:

– **Security flavor**, which determines the protocol rules that the file operations will follow

– **VIP pools** for load balancing and QoS

– **Authentication sources**, which can be labeled as local or external providers

– **Host-access minimal protection levels** for NFS 4.1, with choice of client system

– **Kerberos credentials**, which can be used for additional security

For more security control, the host-based access settings let you specify groups or individual clients for protocol-specific access rules, such as read-write and root squash based on clients.

## Onboard UI

Onboard UI is a web-based GUI for administering and monitoring HPE GreenLake for File Storage system. It is accessible over HTTP access to the users on management network connection to the CNodes. Each instance the onboard UI manages is hosted on the cluster, and it manages that cluster. Onboard UI provides a rich set of options for configurations, storage consumptions, and real-time and historical analytics.

## HPE iLO

HPE iLO is a web-based GUI that enables users to securely configure, monitor, and upgrade hardware components. Each CNode and DNode is equipped with HPE iLO installed. HPE iLO is key to making the CNodes and DNodes operational and enabling them to boot. It helps to simplify setup and engage health monitoring, as well as power and thermal control. These industry-leading features enhance server administrator productivity. For more information about HPE iLO, visit HPE iLO Server Management product page.

## API

The HPE GreenLake for File Storage API also provides a powerful and flexible way to manage your HPE file storage. This API provides functionalities to programmatically manage all your HPE GreenLake File Storage autonomously. With this API, file storage management can easily be integrated with your automation process. HPE GreenLake for File Storage API provides the ability to manage file shares and settings, as well as hardware, through a set of HTTPS requests. The API consists of a server that is part of the HPE GreenLake for File Storage built on HPE Alletra Storage MP OS and runs on both the storage system itself and a definition of the operations, inputs, and outputs of the API.

## Container workload support

Container has become the increasingly prevalent choice for enterprise production applications as more businesses have benefited from the agility and efficiency that containers offer. In addition to providing the standard NFS share for pods to mount, HPE GreenLake for File Storage also supports the container storage interface (CSI) by VAST Data. This CSI driver has the capabilities of dynamic provisioning, volume expansion, multiple storage classes, persistent volume snapshot, and ephemeral volumes. Because the architecture can expose all available storage to a single mount point, the capacity of a dynamically provisioned volume relies on the quota that can be set and expanded automatically on the storage array.

## Summary

As enterprises continue to grow and generate unstructured data at an unprecedented rate, it is critical to implement future-proof solutions that can not only keep up with the growing demands of this AI and ML era but also deliver insights into the data to achieve business results. By using the architectural foundations provided in this paper, you can effectively architect and implement the HPE GreenLake for File Storage solution to solve business challenges in the unstructured data market.

## Appendix

**Table 4.** Terminology used in HPE GreenLake for File Storage

| Commonly used terms | HPE GreenLake for File Storage | Definitions |
|---|---|---|
| Enclosure | CBox | Compute node enclosures, each with two controller nodes |
| Compute node | CNode | Compute node, or compute node input/out modules (IOMs) |
| JBOF enclosure | DBox | JBOF chassis, where the SCMs and SSDs are housed, each of which might house multiple JBOF IOMs |
| JBOF node | DNode | JBOF IOM |
| Drives | SSD, NVRAM | Disk drives, including NVMe SSD drives and NVMe SCM or NVRAM drives |
| System settings | Cluster settings | System-wide configurations |
| File share | File share, view | A directory path that has file share settings/view policy assigned to it and becomes a mountable file share to a client |
| Share settings | Share settings, view policies | A set of attributes that defines the fundamental attributes that determines file protocol specific behaviors |
| Local users/groups | Users/groups | Users and groups that are defined on the HPE GreenLake for File Storage cluster, excluding users and groups from any external authentication providers |

**Note**
HPE does not control and is not responsible for any third-party websites (or the products, services, or content available through them).

**Resources**

HPE GreenLake for File Storage

Data Services Cloud Console Security Guide white paper

HPE iLO Server Management product page

## Learn more at

HPE GreenLake for File Storage

Visit HPE.com

Chat now



a00132707ENW, Rev. 4

HEWLETT PACKARD ENTERPRISE

hpe.com