# Dell PowerStore: Data Efficiencies

July 2022

H18151.4

White Paper

Abstract

This white paper provides an overview of the data efficiency features found within the Dell PowerStore system, including information about the benefits, underlying structures, and management methods.

Dell Technologies

## Copyright

# Contents

# Executive summary

## Overview

Data reduction technologies play a critical role in environments to help reduce the amount of physical storage that is needed to save a dataset. Reducing the amount of data storage that is required not only reduces the number of drives that are required, it also reduces the physical footprint of the solution. This ability helps reduce the total cost of ownership of the storage system. With Dell PowerStore, data efficiency techniques such as pattern matching, deduplication, and compression help to reduce the amount of data that is physically stored within the system. These data efficiency methods are always on, and their data reduction achievements can be viewed within various resources of the system.

This white paper discusses the data efficiency techniques and includes technical information about the data path through the system and the underlying efficiency technologies. Data efficiency savings and reporting is also discussed.

## Audience

This white paper is intended for IT administrators, storage architects, partners, and Dell Technologies employees. It is also intended for any other individuals that are involved in the evaluation, acquisition, management, operation, or design of a Dell networked storage environment using PowerStore.

## Revisions

| Date | Description |
|---|---|
| April 2020 | Initial release: PowerStoreOS 1.0 |
| September 2020 | Updates to the Theory of operation section; other minor updates |
| April 2021 | PowerStore 500 and PowerStoreOS 2.0 updates |
| November 2021 | Template update |
| July 2022 | Minor updates: PowerStoreOS 3.0 |

# Overview

## Introduction

Large amounts of data are created by applications daily and ensuring that data is saved in storage efficiently, helps to reduce the overall solution cost. Data efficiency not only reduces the amount of data that is stored, but it also reduces the physical capacity that is required to store the data. Reducing the footprint of the system can also lead to power and cooling savings. PowerStore includes several data efficiency methods to help reduce the total space that is consumed by storage resources created within the system. Utilizing a combination of thin provisioning, deduplication, and compression techniques, PowerStore reduces the amount of data in the system and stores it as efficiently as possible. These features are controlled by the system and are always enabled.

In PowerStore, all incoming I/O for resources that are supported on the system follow the same data path. These resources include, but are not limited to, volumes, file systems, thin clones, and virtual machines that are based on VMware vSphere Virtual Volumes (vVols). The PowerStore data path, which varies by system model, includes deduplication and compression techniques to help reduce the amount of data that is stored within the system. This document describes the PowerStore data efficiency mechanisms in more detail and outlines how space saving is achieved across these resources.

## PowerStore overview

PowerStore achieves new levels of operational simplicity and agility. It uses a container-based microservices architecture, advanced storage technologies, and integrated machine-learning to unlock the power of your data. PowerStore is a versatile platform with a performance-centric design that delivers multidimensional scale, always-on data reduction, and support for next-generation media.

PowerStore brings the simplicity of public cloud to on-premise infrastructure, streamlining operations with an integrated machine-learning engine and seamless automation. It offers predictive analytics to easily monitor, analyze, and troubleshoot the environment. PowerStore is highly adaptable, providing the flexibility to host specialized workloads directly on the appliance and modernize infrastructure without disruption. It also offers investment protection through flexible payment solutions and data-in-place upgrades.

## Terminology

The following table provides definitions for some of the terms that are used in this document.

**Table 1. Terminology**

| Term | Definition |
|---|---|
| Appliance | Term used for a solution containing a base enclosure and any attached expansion shelves. The size of an appliance could be only the base enclosure, or the base enclosure plus expansion shelves. |
| Cluster | Multiple appliances in a single grouping. Clusters can consist of one appliance or more. Up to four PowerStore appliances can be clustered by simply adding additional appliances as required. |
| File system | A storage resource that can be accessed through file sharing protocols such as SMB or NFS. |
| NAS server | A virtualized network-attached storage server that uses the SMB, NFS, or FTP/SFTP protocols to catalog, organize, and transfer files within file system shares and exports. A NAS server, the basis for multi-tenancy, must be created before you can create file-level storage resources. NAS servers are responsible for the configuration parameters on the set of file systems that it serves. |
| Network File System (NFS) | An access protocol that enables users to access files and folders that are on a network. NFS is typically used by Linux/UNIX hosts. |
| PowerStore Command Line Interface (PSTCLI) | An interface that allows a user to perform tasks on the storage system by typing commands instead of using the user interface. |
| PowerStore Manager | The web-based user interface (UI) for storage management. |

| Term | Definition |
|---|---|
| PowerStore REpresentational State Transfer (REST) API | A set of resources (objects), operations, and attributes that provide interactive, scripted, and programmatic management control of the PowerStore cluster. |
| PowerStore T model | Container-based storage system that is running on purpose-built hardware. This storage system supports unified (block and file) workloads, or block-optimized workloads. |
| PowerStore X model | Container-based storage system that is running inside a virtual machine that is deployed on a VMware hypervisor. In addition to the block-optimized workloads that this storage system offers, it also allows users to deploy applications directly on the array. |
| Server Message Block (SMB) | An access protocol that allows remote file data access from clients to hosts on a network. This protocol is typically used in Microsoft Windows environments. |
| Snapshot | A point-in-time view of data stored on a storage resource. A user can recover files from a snapshot or restore a storage resource from a snapshot. |
| Storage resource | The top-level object a user can provision that is associated with a specific quantity of storage. All host access and data protection activities are performed at this level. |
| Thin clone | A read/write copy of a volume, volume group, file system, NAS server, or snapshot that shares blocks with the parent resource. |
| Volume | A block-level storage device that can be shared using a protocol such as iSCSI or Fibre Channel. |
| Volume group | A storage instance which contains one or more volumes within a storage system. |

### We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

**Author:** Ryan Poulin

**Note**: For links to other documentation for this topic, see the PowerStore Info Hub.

## Theory of operation

### Introduction

The following section describes PowerStore data efficiency techniques and how the data path handles read and write operations. The PowerStore 1000 – 9200 models and the PowerStore 500 have different I/O handling and are described in detail in the sections below.

### PowerStore 1000 to 9200

The following sections describe how PowerStore system models 1000 to 9200 provide inline data efficiencies.

## Write path background information

Before the data efficiency techniques of PowerStore 1000, 1200, 3000, 3200, 5000, 5200, 7000, 9000, and 9200 are discussed, the system design and data path must be reviewed to explain how these PowerStore systems provide inline data efficiencies. Figure 1 shows a high-level diagram of the components within these systems that are part of the data path. In this figure, node A and node B are shown side-by-side for demonstration purposes. Within the system, DRAM memory is used as a caching layer as data enters and exits the system. All data passes through and interacts with DRAM memory. How the data interacts with DRAM memory depends if the I/O is a read or a write I/O. How a read and write I/O passes through DRAM memory is explained later in this section.

Another major component of the data path are the data drives. The drives, which are outlined in orange in the following figure, provide the physical capacity to the system to store data. If additional enclosures are attached to the system, they similarly add to the usable capacity of the system. Within PowerStore, any data drives in a PowerStore 1000 to 9200 system contribute to a single, large, usable capacity. Drives used for the NVMe NVRAM cache, or SCM drives dedicated for metadata tiering, are excluded from the usable capacity of the system. This space is shared for all resources within the system.

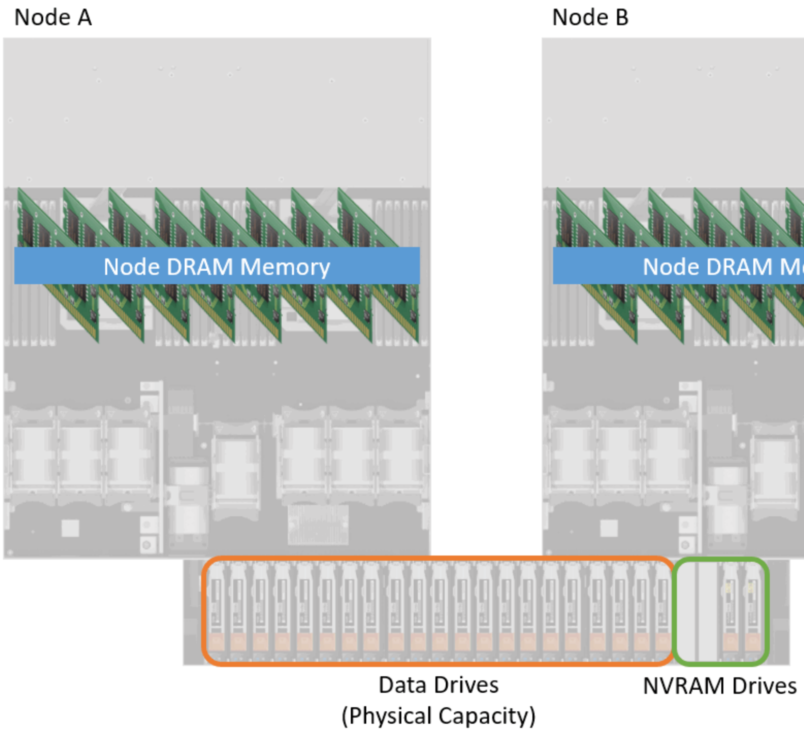


**Figure 1. Data path components in PowerStore 1000 to 9200 models**

PowerStore 1000 to 9200 model systems include NVRAM drives, which are part of the PowerStore system’s data path. These drives are circled in green in the lower right of the above figure. The NVRAM drives are hard drives that are located in the far-right slots of the base enclosure of the appliance. Depending on the system model, these two or four NVRAM drives are used as an additional write I/O cache location for protection purposes within the system. As writes enter the system, a copy of each write is placed in DRAM memory and the NVRAM drives before the host is acknowledged that the write has been saved within the system. After the host is acknowledged, data is passed through the deduplication and compression logic before being saved to the data drives of the system.

The following figure shows more information about the NVRAM drives. As mentioned previously, each appliance contains either two or four NVRAM drives, which are physically located in the far-right drive slots of the base enclosure of the appliance. If the system only contains two NVRAM drives, they are populated in drive slots 23 and 24. If four NVRAM drives are present, then slots 21 through 24 are occupied. NVRAM drives are deployed in mirrored pairs to guard against a single drive failure. Drives 21 and 22 are configured in a mirrored pair, as well as drives 23 and 24. During an NVRAM drive failure, write caching continues to be enabled, and the mirrored pair is reestablished when the faulted drive is replaced.

The appliance also provides battery backup to the NVRAM drives using internal batteries within each node. Slots 21 and 23 are connected to Node A’s internal battery device, while Node B provides power to slots 22 and 24. Battery backup is required as NVRAM drives contain both volatile and nonvolatile media. The volatile media provides fast access speeds, and is used as a backup location for write caching within the system while the appliance is under normal operation. If power to the appliance is interrupted or the system is powered off, the volatile write cache is destaged to the nonvolatile media within the NVRAM drive. Once the write cache information is safely stored, power is removed from the drives and the system completes its shutdown operation. The NVRAM design and operations replace the need to protect the contents of DRAM write cache.

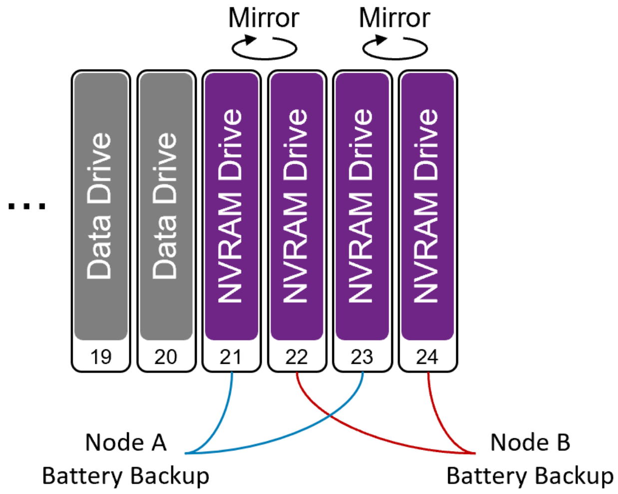


**Figure 2. NVRAM drives in PowerStore**

## Writes

All writes entering a PowerStore 1000, 1200, 3000, 3200, 5000, 5200, 7000, 9000, and 9200 appliance follow the same data path, regardless of the resource type. Figure 3 displays a high-level diagram of a write I/O entering the system. The figure also shows the steps that are taken to store the information within the system before acknowledging the host. Step 1 depicts an I/O entering Node B. The I/O is saved within the node DRAM memory and is analyzed to determine what type of I/O it is, what resource it is intended for, and the location within the resource being updated or requested.

If the I/O is determined to be a write, the information is stored within write cache in the Node DRAM Memory (step 2). A copy of the write is also written to the NVRAM drives, as depicted by Step 3 in the following figure. As mentioned previously, the appliance NVRAM drives are used as an additional write cache location within the system for redundancy purposes. Each NVRAM drive is also dual-ported, meaning that each node has access to the drives through physical internal connections and the information that is contained in them. If required, the peer node can access the data as needed.

Step 4 in the following figure occurs for each write I/O that enters the system. Information is passed between the nodes to update the peer that a new write has been received and that it has the newest copy of the data for the resource. This token includes information about the I/O such as what resource was updated, the address within the resource that was updated, and the location where the I/O was saved to within the write cache. If the information is requested through the peer node, the node can access the data within the NVRAM drives by its own internal channels.

The last step (5) in completing the I/O is to acknowledge the host. After the information is safely stored within the appliance and all other actions are complete, the host is acknowledged. The data is passed through the deduplication and compression algorithms at a later time.

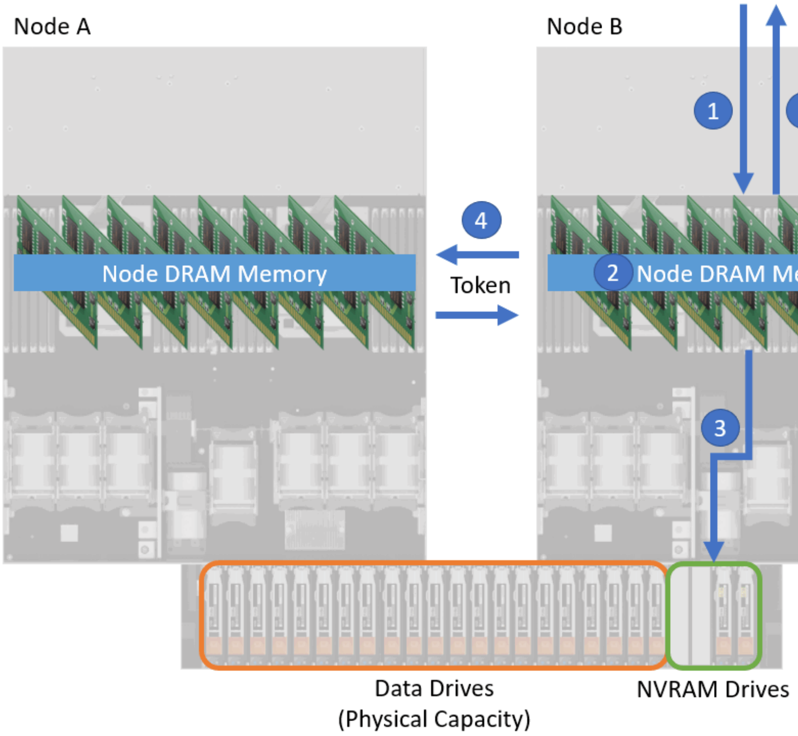


**Figure 3. A write I/O entering PowerStore 1000 to 9200 model system**

## PowerStore 500

### Writes

Unlike the PowerStore 1000, 1200, 3000, 3200, 5000, 5200, 7000, 9000, 9200 appliances, the PowerStore 500 does not support NVMe NVRAM drives. Instead, the system utilizes each Node DRAM memory as a write caching layer to temporarily store data before it is passed through the deduplication and compression process. The following figure is a high-level diagram of a write I/O entering a PowerStore 500 system and the steps that are taken before acknowledging the host. Step 1 depicts a write request entering Node B. The information is stored within the Node DRAM Memory, as shown in Step 2 below. The information is then analyzed to determine what type of I/O it is, what resource it is intended for, and the location within the resource being updated or requested.

If the I/O is determined to be a write, the information is then mirrored to the peer Node DRAM Memory, as depicted by Step 3. A full copy of the data and information about the request is passed through the internal connections between the nodes in the system. After the data is mirrored to the peer node, the host is acknowledged that the write has been persisted on the system. The data is passed through the deduplication and compression algorithms at a later time.

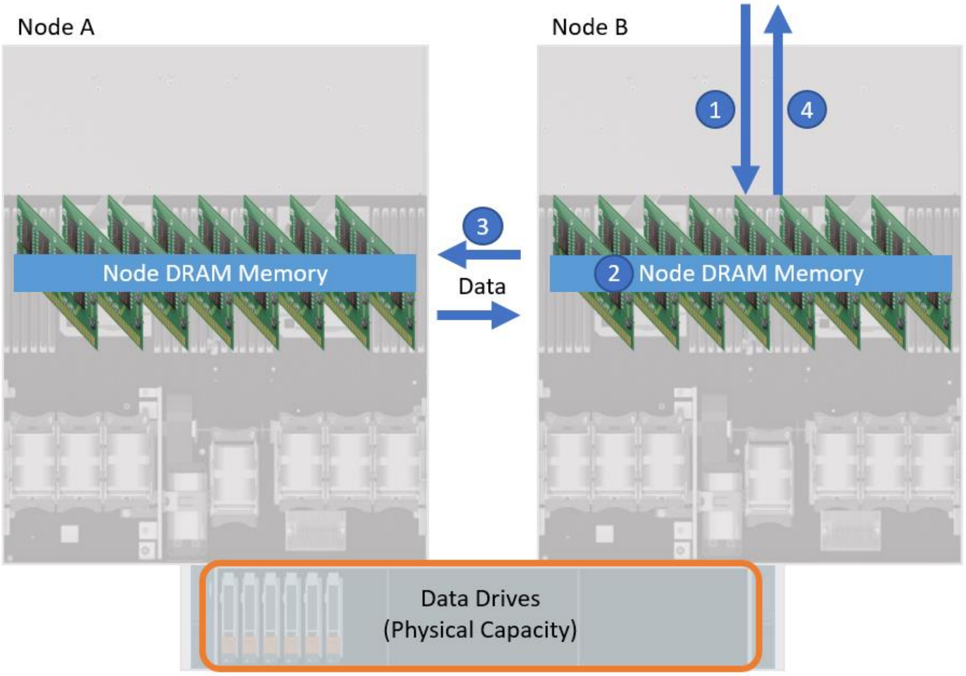


**Figure 4. A write I/O entering PowerStore 500 model system**

Within a PowerStore 500 appliance, Battery Backup Units (BBUs) are used to protect the contents of the appliance’s volatile DRAM Memory. If power is interrupted to the system, the BBUs ensure the cached contents within DRAM memory are safely persisted to a nonvolatile M.2 SATA device contained within the node. Once all data is protected, the system finishes its graceful power down procedure. Once power is resumed, cached data is restored to DRAM Memory during the bootup process, thus preserving the previous cache contents. Normal operations continue and data is passed through deduplication and compression algorithms at a later time.

## PowerStore deduplication and compression

Deduplication and compression within a PowerStore appliance occur as data is copied from the appliance's DRAM memory-based write cache to the data drives within the system. During this process, data is stored within the back-end storage in full stripe writes, which are created with the data remaining after it is passed through the deduplication and compression process. The following figure outlines this process at a high level for each PowerStore model.

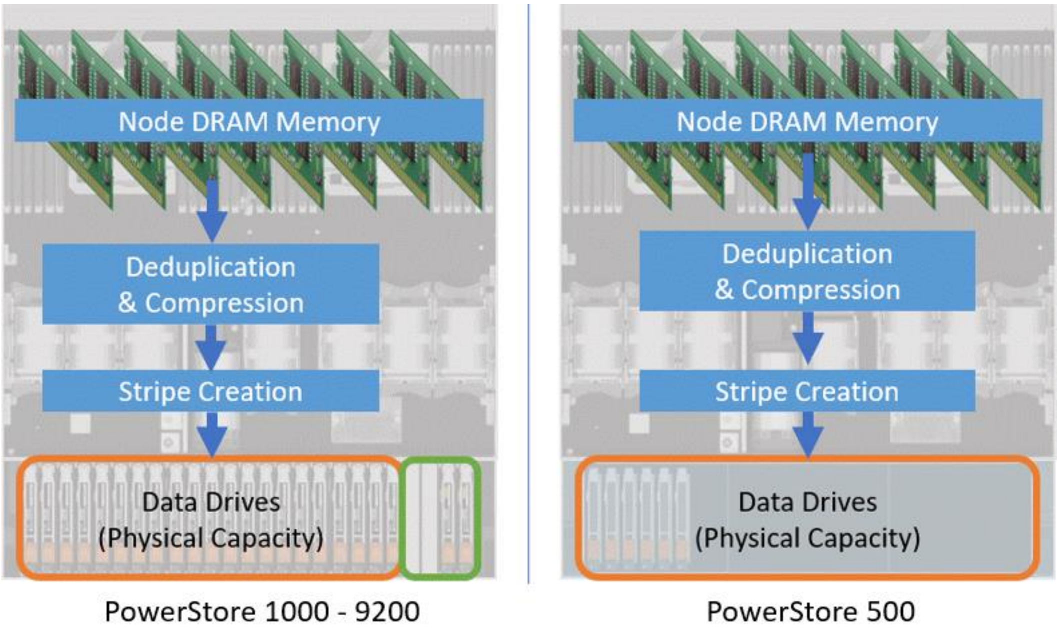


**Figure 5. Data being stored in the data drives**

All writes to the system pass through deduplication and compression logic to ensure the data is stored as efficiently as possible. These software features are part of PowerStore data path operations and cannot be disabled by the user. In releases prior to the PowerStoreOS 2.0 release, deduplication and compression are always engaged inline and cannot be bypassed. In the PowerStoreOS 2.0 release and higher, PowerStore will dynamically prioritize workload performance and defer deduplication operations to a period in time with a less extreme I/O demand. If this occurs, compression remains enabled and continues to achieve space savings. Once deduplication is resumed, blocks that were skipped are passed through the deduplication algorithm to determine if deduplication can occur. Used space is recovered if deduplication can be achieved.

When data is passed through the deduplication and compression logic, it first passes through common pattern detection, reviewing the data for all zeros or ones. If one of these patterns is matched, deduplication occurs and the metadata within the resource is updated to state what data pattern was found in that location. The metadata for a resource tells the system how it can re-create or locate data for an address within a storage resource. For example, if 4 KB of zeros were deduplicated, starting at a particular address within the resource, the metadata would have information to provide zeros as a response for that address if it is requested.

If a pattern was not found, the data is passed through the deduplication algorithm. This algorithm first creates a fingerprint for each 4 KB block of data within the resource using a hashing algorithm. Once created, the fingerprint is compared to other fingerprints which represent data within the PowerStore single deduplication domain. The deduplication domain allows data for any resource within the appliance to deduplicate. If a match is found, the physical storage is single-instanced. If a match is not found, the data is compressed and placed into a full stripe write to be written to the system.

In releases prior to PowerStoreOS 2.0, the fingerprint cache is fixed in size and resides in system memory. Due to the fixed size, older fingerprints may be replaced with newer fingerprints, potentially reducing the ability to achieve deduplication to previously written data. In PowerStoreOS 2.0 and higher, the fingerprint cache can expand into the data drives, consuming space as needed to store fingerprints. This allows the system to retain all fingerprints created on the appliance, potentially allowing for larger deduplication savings than in previous releases. The entire fingerprint cache is also mirrored within the data drives in the 2.0 release, protecting it against unforeseen issues.

The following figure shows an example of the result of the deduplication and compression logic within the PowerStore system. In this example, Resource 1 and Resource 2 have written four blocks of data each, and they are stored within the system. Deduplication within the PowerStore system allows blocks from different resources to deduplicate, reducing the amount of space consumed by duplicate data within the system. In this example, Block **A** is unique to Resource 1, and it is only referenced by Resource 1. Block **E** is present in both resources, and has deduplicated within the system, thus saving space. Further savings can be achieved if the data is compressible, as the resources can both reference a compressed block.

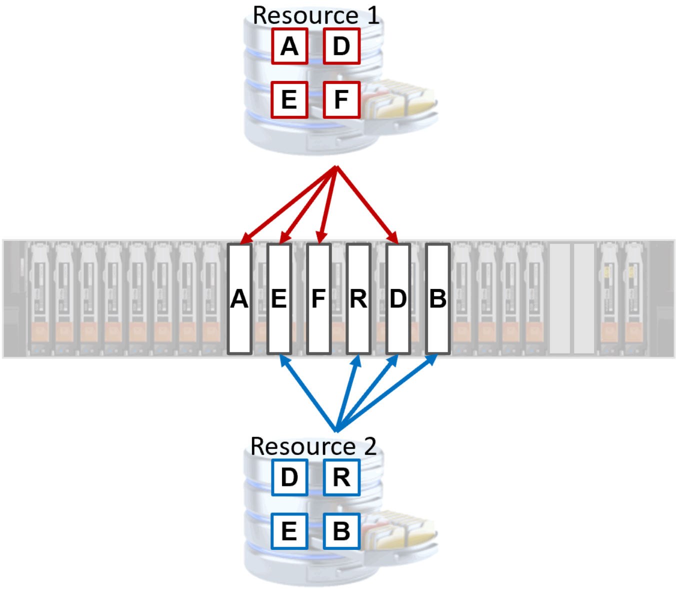


**Figure 6. Deduplication example in PowerStore**

## Reads

When a read operation is sent to a resource within the PowerStore system, the system must first determine where the latest copy of the data being requested is located. If the latest copy is in the Node DRAM memory, the information is returned to the host or application requesting the data, and the read operation completes. If not, the resource metadata is reviewed to determine where the current data is located. If the data was a simple pattern, such as all zeros or ones, the block is re-created in DRAM memory and sent to the host. If the data being requested resides in the data drives within the system, the data is read and re-created in DRAM memory and sent to the host. Read operations to data residing within the drives does not cause the data to be changed in any way, thus preserving any deduplication and compression savings.

### Overwrites

When an update to a previously written location is received, the write follows the same data path as a new write into the system. The data is saved in write cache and the host is acknowledged to complete the I/O operation. The new data that is received eventually passes through the deduplication and compression algorithms. If needed, the data is written to the back-end drives as part of the normal cache-cleaning operation to a new location within the storage, and the location previously used is invalidated.

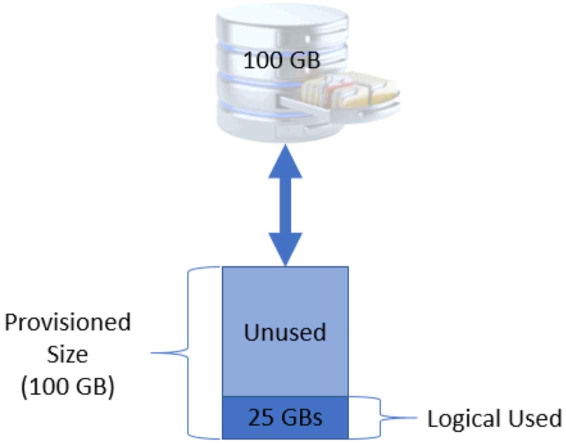
## Efficiency reporting

### Overview

By using multiple space-savings techniques, PowerStore attempts to use the least amount of disk space to store user data. For reporting purposes, space-savings information is available at the cluster, appliance, and resource level, and includes multiple space-saving metrics. Depending on the object being viewed, these metrics may include an overall efficiency value, and thin, snapshot, and data reduction savings values. At the cluster level, space savings information is aggregated from each of the appliances within the cluster. Likewise, at the appliance level, savings information is aggregated for any resources on the appliance.

The overall efficiency value is a computed ratio of the total provisioned space to the physical used space. The total provisioned space not only includes the provisioned size for each of the resources within the system or cluster, such as the size of the volumes, file systems, and all thin clones, but also any snapshots. The physical used space is the total used space on the drives to store the data after thin, deduplication, and compression savings are achieved. The used space is not the total amount of space that is consumed from a host or client perspective before space savings is applied. The overall efficiency value is available for the entire cluster and each appliance within the cluster.

All resources within PowerStore are thinly provisioned, which means the system does not reserve the total provisioned size of a resource from the storage, but rather allocates space as needed. The thin savings achieved for a cluster, appliance, or resource is the ratio of the total provisioned size to the logical space used. The logical space that is used in this instance is the total space that is consumed from a host or client perspective without the effects of deduplication or compression. As shown in the following figure, if a 100 GB volume has 25 GB of written data, the savings due to the resource being thin is 4:1.



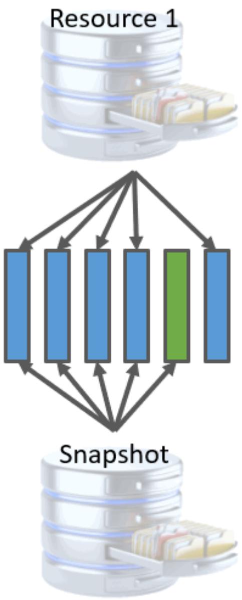
$$Thin\ Savings = \frac{Provisioned\ Size}{Logical\ Used}$$

$$Thin\ Savings = \frac{100\ GBs}{25\ GBs}$$

$$Thin\ Savings = 4:1$$

**Figure 7. Thin savings example**

Snapshot savings, which are displayed as snap savings within PowerStore Manager, is the ratio of logical used space for the snapshots to data that is uniquely owned by the snapshots. The logical used space is the amount of space consumed by data which is in common between the snapshot and the parent resource when the snapshot was created. For example, a volume has 5 GB of data consumed and a snapshot is then taken. The logical used space by the snapshot in this instance is 5 GB, which is shared with the parent resource. If 1 GB of data is overwritten in the volume, the amount of data that is uniquely owned by the snapshot is now 1 GB. In this example, also shown in Figure 8, the snap savings is 5 GB/1 GB or 5:1.



$$Snapshot\ Savings = \frac{Logical\ Used\ by\ Snapshot}{Uniquely\ owned\ by\ Snapshot}$$

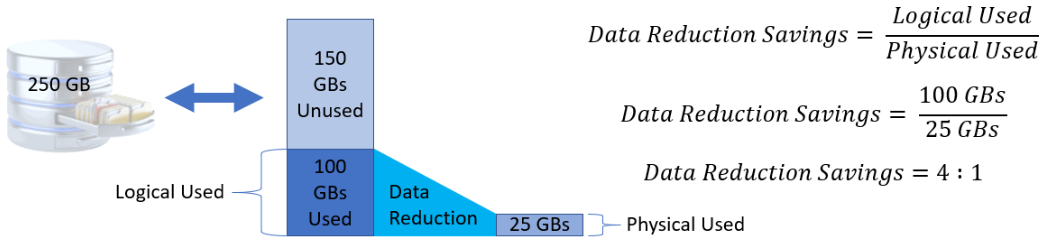
$$Snapshot\ Savings = \frac{5\ GBs}{1\ GB}$$

$$Snapshot\ Savings = 5:1$$

*Note: For demonstration purposes, assume each block is 1 GB of data

**Figure 8. Snapshot savings example**

The last space efficiency metric is the data reduction ratio. This ratio is a comparison between the amount of space a dataset would have consumed if no space savings were achieved (logical used), compared to the amount of physical space occupied after passing the data through deduplication and compression (physical used). If 100 GB of data only needs 25 GB of physical storage after deduplication and compression, then the data reduction savings is 4:1. This example is shown in the following figure. The data reduction ratio is only displayed for the cluster and appliance levels, since deduplication and compression savings span resources in PowerStore.



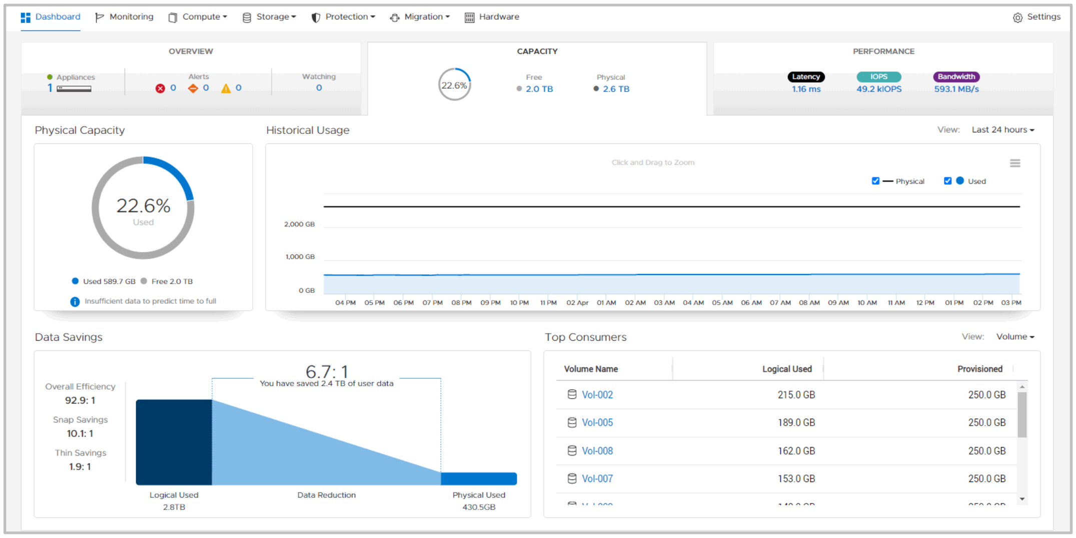
$$Data\ Reduction\ Savings = \frac{Logical\ Used}{Physical\ Used}$$

$$Data\ Reduction\ Savings = \frac{100\ GBs}{25\ GBs}$$

$$Data\ Reduction\ Savings = 4 : 1$$

Figure 9. Data reduction savings example

You can view the **Physical Capacity**, **Historical Usage**, and **Data Savings** metrics for the cluster, and the cluster top resource consumers from the **Dashboard > Capacity** tab. The cluster metrics are aggregated from the metrics of its appliances and resources, such as volume groups, volumes, file systems, and virtual machines. An example of the **Capacity** tab for the cluster is shown in the following figure. Under the **Data Savings** title, the **Overall Efficiency**, **Snap Savings**, and **Thin Savings** for the cluster can be viewed. A graphic exists showing the relationship between the **Logical Used** and **Physical Used** space, along with the **Data Reduction** savings that were achieved.



Insufficient data to predict time to full

You have saved 2.4 TB of user data

Overall Efficiency
92.9: 1
Snap Savings
10.1: 1
Thin Savings
1.9: 1

| Volume Name | Logical Used | Provisioned |
|---|---|---|
| Vol-002 | 215.0 GB | 250.0 GB |
| Vol-005 | 189.0 GB | 250.0 GB |
| Vol-008 | 162.0 GB | 250.0 GB |
| Vol-007 | 153.0 GB | 250.0 GB |

Figure 10. PowerStore Manager: Dashboard > Capacity tab

When hovering over the **Physical Capacity** chart, the user is provided with more information about the **Used** and **Free** capacity within the system. For each metric, **User** and **System** space is also reported. For **Used**, the **User** and **System** space metrics state how much capacity is currently in use for **User** data and by the **System**. The **System** value includes the amount of space used by system metadata. For **Free**, the **User** and **System** values report how much free space exists.

When a resource is deleted, capacity that was used by user data is freed back to user space and any metadata space is freed to system space. When additional capacity is required for new user data or metadata, the system utilizes the free capacity within the system. If there is no free system space to create additional metadata, user space will be reduced to satisfy the need. This information is available within the **Physical Capacity** chart for the cluster and each individual appliance. An example is shown in the following figure.

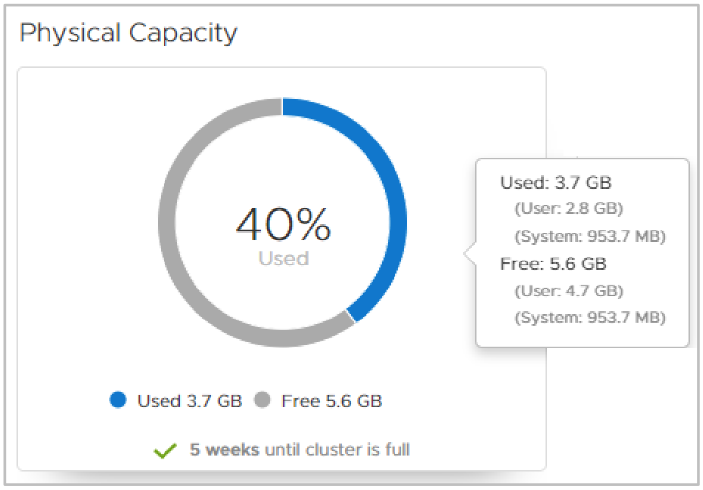


**Figure 11. PowerStore Manager: Dashboard > Capacity tab > Physical Capacity**

You can view the capacity metrics for an appliance from the **Hardware > Appliances > [appliance] > Capacity** tab. On this screen, as shown in the following figure, you can view the **Physical Capacity**, **Historical Usage**, and **Data Savings** information for the appliance. Within the **Data Savings** block, you can view the **Overall Efficiency**, **Snap Savings**, and **Thin Savings** for the appliance. As mentioned previously, this information is aggregated, along with information for other appliances within the same cluster, to create the cluster metrics. Lastly, the **Data Reduction** savings ratio is presented within the graphic displaying the comparison between **Logical Used** and **Physical Used** space for the appliance.

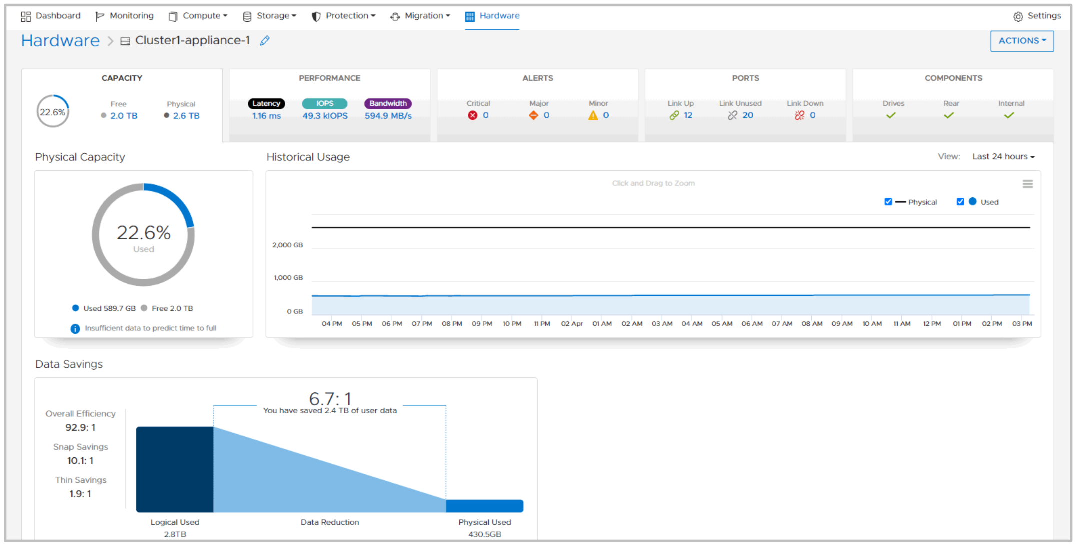
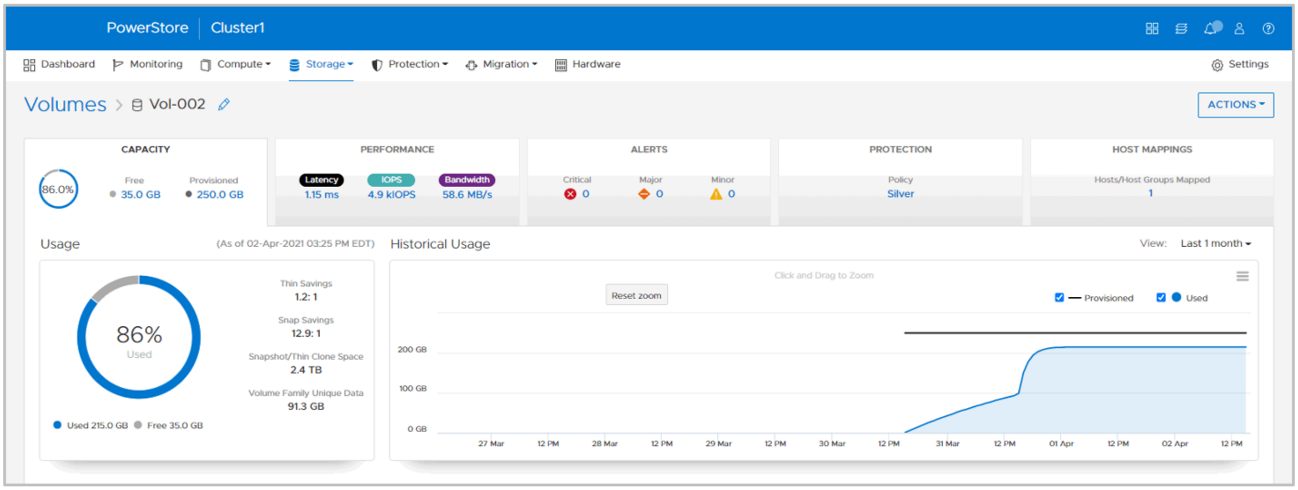


**Figure 12. PowerStore Manager: Hardware > Appliances > [appliance] > Capacity tab**

**Thin Savings** and **Snap Savings** can be viewed for volumes within a PowerStore system. This information can be found on the **Capacity** tab within the properties of the resource. Within the **Usage** box, the volume **Used** and **Free** space are displayed, along with the **Thin Savings** and **Snap Savings**. The amount of **Snapshot/Thin Clone Space** being consumed, and the amount of **Volume Family Unique Data** can be viewed on this page. The **Historical Usage** of the volume is also displayed on the **Capacity** tab, which shows the **Provisioned** size and **Used** space for the resource. An example of this page for a volume is found in the following figure. In PowerStoreOS 3.0, more usage information for file systems was added to the Capacity tab. The information includes Thin Savings, Snap Savings, Snapshot/Thin Clone space, and Filesystem Family Unique Data. The Historical Usage chart was also added for file systems.



Free 35.0 GB

Provisioned 250.0 GB

Latency 1.15 ms

IOPS 4.9 kIOPS

Bandwidth 58.6 MB/s

Critical 0

Major 0

Minor 0

Policy Silver

Hosts/Host Groups Mapped 1

(As of 02-Apr-2021 03:25 PM EDT)

View: Last 1 month

Thin Savings 1.2: 1

Snap Savings 12.9: 1

Snapshot/Thin Clone Space 2.4 TB

Volume Family Unique Data 91.3 GB

**Figure 13. PowerStore Manager: Volume Properties > Capacity tab**

# Interoperability

## Overview

Data efficiency features within the PowerStore system are always on and are directly part of the read and write code path. All blocks written to the drives within the system can be shared among all resources within the appliance. Therefore, all resources including their snapshots and thin clones support the thin, deduplication, and compression efficiency features of PowerStore. Other features that transfer data into the system, such as import, replication, and migration, support the data efficiency features of the system. These features are passed through the same code path as a host write into the system.

When data is transferred out of the PowerStore system, it must first be returned to its full size. For instance, when a host or client requests data, the data must first be re-created within DRAM memory before it can be passed to this host. This rule is true for replication since data is passing to a remote cluster and for internal migrations when a resource is being moved from one appliance to another within the cluster. For replication and migration, when the data reaches its destination, it is subject to the data efficiency algorithms of that appliance.

# Conclusion

## Summary

PowerStore offers powerful data efficiency techniques which help reduce the amount of storage that is required to store datasets for the various resources within the system. Through the always-on features of thin resources, deduplication, compression, and the intuitive management of data that is written within the storage of the system, PowerStore uses its space as efficiently as possible. By reducing the amount of storage needed to store a dataset, data efficiency techniques help to further reduce the total cost of ownership of a solution.

# Appendix: Technical support and resources

## Dell Technologies documentation

The Dell Technologies Info Hub > Storage site provides expertise that helps to ensure customer success with Dell storage platforms.

Dell.com/powerstoredocs provides detailed documentation about how to install, configure, and manage Dell PowerStore systems.