

# Next-Generation Storage Efficiency with Dell PowerScale SmartDedupe

January 2023

H12395.17

## White Paper

### Abstract

This paper describes Dell PowerScale SmartDedupe software that is used for data deduplication in PowerScale scale-out NAS storage environments. PowerScale SmartDedupe is a native data reduction capability that enables enterprises to reduce storage costs and footprint and increase data efficiency, without sacrificing data protection or management simplicity.

Dell Technologies

## Copyright

The information in this publication is provided as is. Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © 2023 Dell Inc. or its subsidiaries. All Rights Reserved. Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Intel, the Intel logo, the Intel Inside logo and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Other trademarks may be trademarks of their respective owners. Published in the USA January 2023 H12395.17.

Dell Inc. believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

# Contents

- Executive summary.....4
- OneFS SmartDedupe .....5
- SmartDedupe management .....10
- SmartDedupe monitoring and reporting.....11
- SmartDedupe best practices and considerations.....22
- SmartDedupe and OneFS feature integration .....23
- SmartDedupe use cases .....26
- SmartDedupe and OneFS storage utilization .....27
- Conclusion.....27

## Executive summary

### Overview

Information technology managers across most areas of commerce are grappling with the challenges presented by explosive file data growth, which significantly raises the cost and complexity of storage environments. Business data is often filled with significant amounts of redundant information. For example, each time multiple employees store an email attachment, many of the same files are stored or replicated, resulting in multiple copies that take up valuable disk capacity. Data deduplication is a specialized data reduction technique that allows for the elimination of duplicate copies of data.

Deduplication is yet another milestone in Dell PowerScale data efficiency solutions and a key ingredient for organizations that want to maintain a competitive edge.

### Audience and scope

The target audience for this white paper is anyone configuring and managing SmartDedupe deduplication in a PowerScale clustered storage environment. It is assumed that the reader has an understanding and working knowledge of the OneFS components, architecture, commands, and features.

This paper presents information for deploying and managing SmartDedupe deduplication on a Dell PowerScale cluster. This paper does not intend to provide a comprehensive background to the OneFS architecture.

For more details about the OneFS architecture, see the [OneFS Technical Overview white paper](#).

For more information about OneFS commands and feature configuration, see the [OneFS Administration Guide](#).

### Revisions

Date	Description
November 2013	Initial release for OneFS 7.1
June 2014	Updated for OneFS 7.1.1
November 2014	Updated for OneFS 7.2
June 2015	Updated for OneFS 7.2.1
November 2015	Updated for OneFS 8.0
September 2016	Updated for OneFS 8.0.1
April 2017	Updated for OneFS 8.1
November 2017	Updated for OneFS 8.1.1
February 2019	Updated for OneFS 8.1.3
April 2019	Updated for OneFS 8.2
August 2019	Updated for OneFS 8.2.1
December 2019	Updated for OneFS 8.2.2
June 2020	Updated for OneFS 9.0

Date	Description
October 2020	Updated for OneFS 9.1
April 2021	Updated for OneFS 9.2
September 2021	Updated for OneFS 9.3
April 2022	Updated for OneFS 9.4
January 2023	Updated for OneFS 9.5

## 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:** Nick Trimbee

---

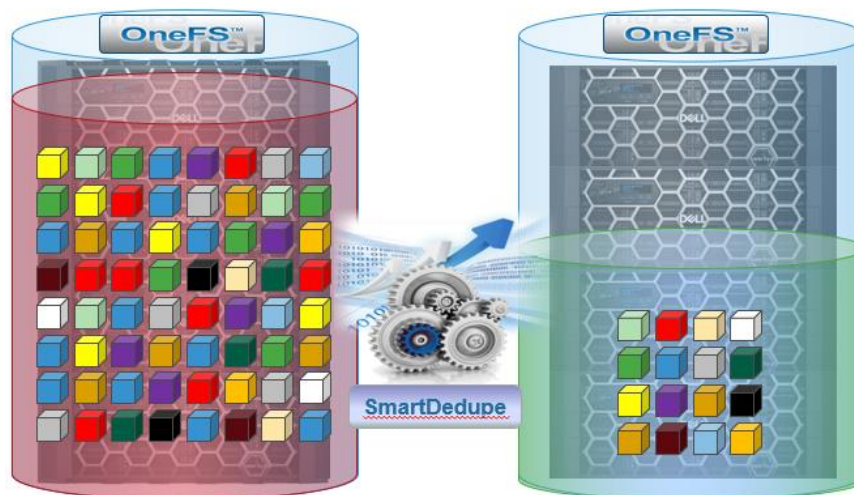
**Note:** For links to other documentation for this topic, see the [PowerScale Info Hub](#).

---

# OneFS SmartDedupe

## Overview

Dell PowerScale SmartDedupe maximizes the storage efficiency of a cluster by decreasing the amount of physical storage required to house an organization's data. Efficiency is achieved by scanning the on-disk data for identical blocks and then eliminating the duplicates. This approach is commonly referred to as post-process, or asynchronous, deduplication.



**Figure 1. Storage efficiency with SmartDedupe**

After discovering duplicate blocks, SmartDedupe moves a single copy of those blocks to a special set of files known as shadow stores. During this process, duplicate blocks are removed from the actual files and replaced with pointers to the shadow stores.

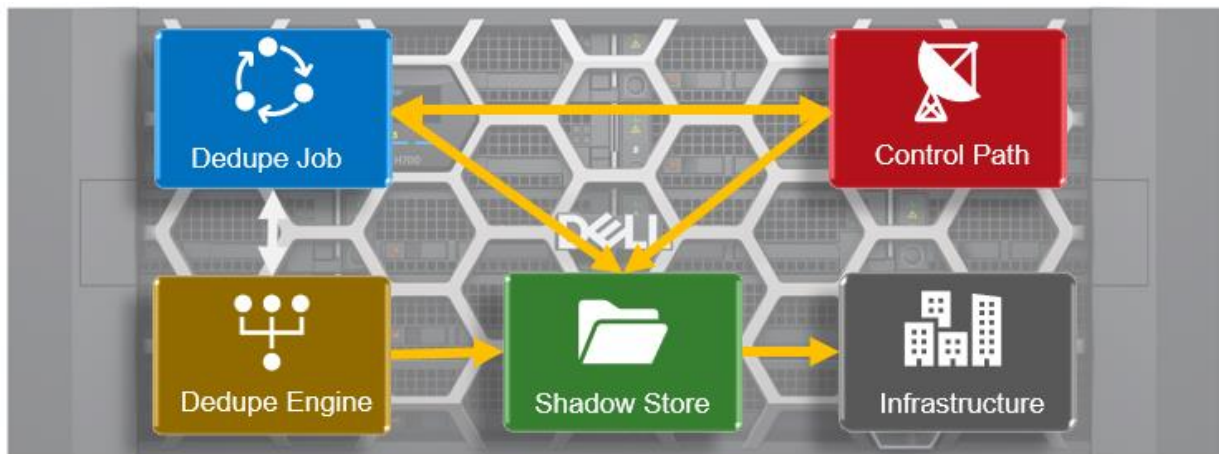
Post-process deduplication first stores new data on the storage device and then analyzes the data, looking for commonality. The initial file-write or modify performance is not affected because additional computation is not required in the write path.

## SmartDedupe architecture

Architecturally, SmartDedupe consists of five principal modules:

- Deduplication Control Path
- Deduplication Job
- Deduplication Engine
- Shadow Store
- Deduplication Infrastructure

The SmartDedupe control path includes the OneFS Web Management Interface (WebUI), command-line interface (CLI), and RESTful platform API. It is responsible for managing the configuration, scheduling, and control of the deduplication job. The job itself is a highly distributed background process that manages the orchestration of deduplication across all the nodes in the cluster. Job control encompasses file system scanning, detection, and sharing of matching data blocks, in concert with the Deduplication Engine. The Deduplication Infrastructure layer is the kernel module that performs the consolidation of shared data blocks into shadow stores, the file system containers that hold both physical data blocks and references, or pointers, to shared blocks. These elements are described later in more detail.



**Figure 2. SmartDedupe modular architecture**

## Deduplication engine—sampling, fingerprinting, and matching

One of the most fundamental components of SmartDedupe, and deduplication in general, is fingerprinting. In this part of the deduplication process, unique digital signatures, or fingerprints, are calculated using the SHA-1 hashing algorithm, one for each 8 KB data block in the sampled set.

When SmartDedupe runs for the first time, it scans the dataset and selectively samples data blocks from it, creating the fingerprint index. This index contains a sorted list of the digital fingerprints, or hashes, and their associated blocks. After the index is created, the fingerprints are checked for duplicates. When a match is found, during the sharing phase, a byte-by-byte comparison of the blocks is performed to verify that they are absolutely

identical and to ensure there are no hash collisions. Then, if they are determined to be identical, the block's pointer is updated to the existing data block and the new, duplicate data block is released.

Hash computation and comparison are only used during the sampling phase.

[Deduplication job and infrastructure](#) describes the deduplication job phases in detail. For the block sharing phase, full data comparison is employed. SmartDedupe also operates on the premise of variable length deduplication, where the block matching window is increased to encompass larger runs of contiguous matching blocks.

## Shadow stores

OneFS shadow stores are file system containers that allow data to be stored in a shareable manner. As such, files on OneFS can contain both physical data and pointers, or references, to shared blocks in shadow stores. Shadow stores were introduced in OneFS 7.0, initially supporting OneFS file clones, and there are many overlaps between cloning and deduplicating files. The other main consumer of shadow stores is OneFS Small File Storage Efficiency (SFSE) for archive. This feature maximizes the space utilization of a cluster by decreasing the amount of physical storage required to house a small file archive repository, such as a typical healthcare PACS dataset.

Shadow stores are similar to regular files but are hidden from the file system namespace, so they cannot be accessed through a pathname. A shadow store typically grows to a maximum size of 2 GB (or about 256 K blocks), with each block able to be referenced by 32,000 files. If the reference count limit is reached, a new block is allocated, which may or may not be in the same shadow store. Additionally, shadow stores do not reference other shadow stores. And snapshots of shadow stores are not permitted because the data stored in shadow stores cannot be overwritten.



Figure 3. OneFS duplicate block sharing

## Deduplication job and infrastructure

Deduplication is performed in parallel across the cluster by the OneFS Job Engine through a dedicated deduplication job, which distributes worker threads across all nodes. This distributed work allocation model allows SmartDedupe to scale linearly as a cluster grows and additional nodes are added.

The Job Engine performs the control, impact management, monitoring, and reporting of the deduplication job in a manner that is similar to other storage management and maintenance jobs on the cluster.

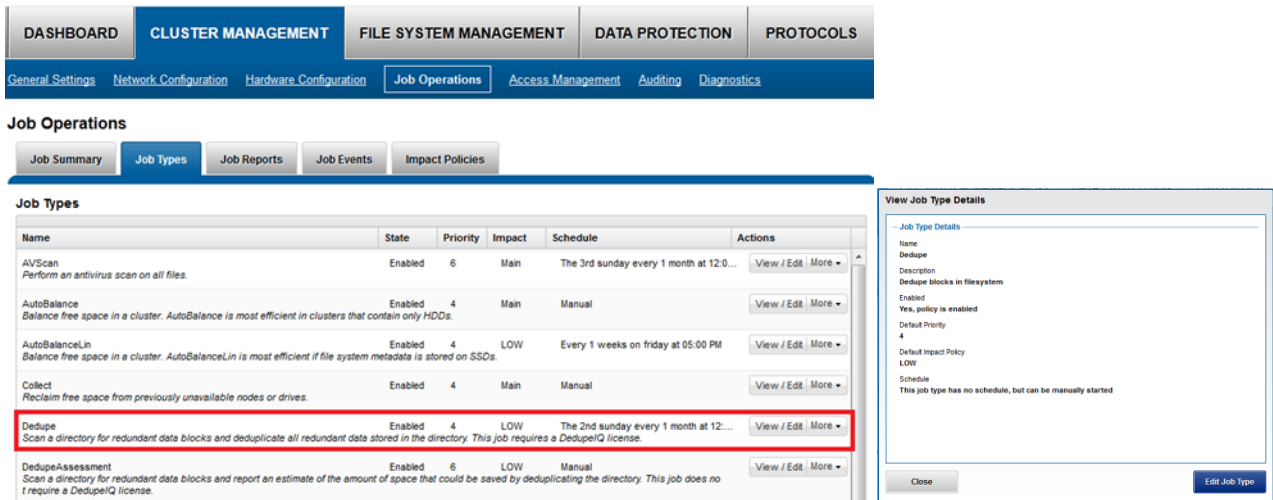


Figure 4. SmartDedupe job control through the OneFS WebUI

While deduplication can run concurrently with other cluster jobs, only a single instance of the deduplication job, albeit with multiple workers, can run at any one time. Although the overall performance impact on a cluster is relatively small, the deduplication job does consume CPU and memory resources.

The primary user-facing component of SmartDedupe is the deduplication job. This job performs a file system tree walk of the configured directory, or multiple directories, hierarchy.

---

**Note:** The deduplication job automatically ignores (does not deduplicate) the reserved cluster configuration information located under the `/ifs/.ifsvar/` directory, and also any file system snapshots.

---

Architecturally, the duplication job, and supporting deduplication infrastructure, consists of the following phases:

- Sampling
- Duplicate Detection
- Block Sharing
- Index Update

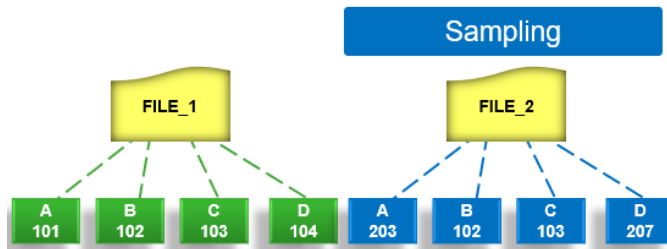
These phases are described in more detail below.

Because the SmartDedupe job is typically long running, each of the phases runs for a set time period, performing as much work as possible before yielding to the next phase. When all four phases have been run, the job returns to the first phase and continues from where it left off. Incremental deduplication job progress tracking is available through the OneFS Job Engine reporting infrastructure.

### Sampling phase

In the sampling phase, SmartDedupe performs a tree-walk of the configured dataset to collect deduplication candidates for each file.



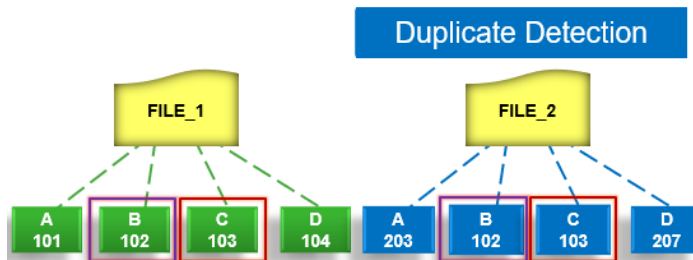


**Figure 5. SmartDedupe job sampling phase**

The rationale is that a large percentage of shared blocks can be detected with only a smaller sample of data blocks represented in the index table. By default, the sampling phase selects one block from every sixteen blocks of a file as a deduplication candidate. For each candidate, a key/value pair consisting of the block's fingerprint (SHA-1 hash) and file system location (logical inode number and byte offset) is inserted into the index. Once a file has been sampled, the file is marked and is not rescanned until it has been modified, drastically improving the performance of subsequent deduplication jobs.

### Duplicate detection phase

During the duplicate, or commonality, detection phase, the deduplication job scans the index table for fingerprints (or hashes) that match those of the candidate blocks.

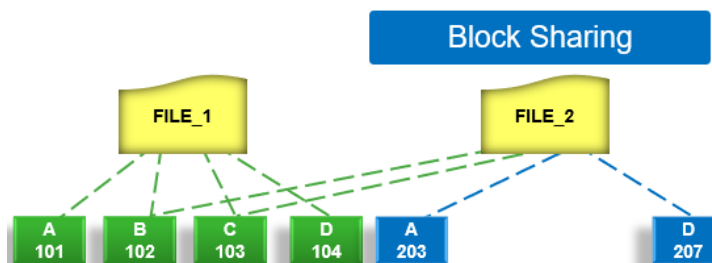


**Figure 6. SmartDedupe job duplicate detection phase**

If the index entries of two files match, a request entry is generated. To improve deduplication efficiency, a request entry also contains pre- and post-limit information. This information contains the number of blocks in front of and behind the matching block that the block sharing phase should search for a larger matching data chunk, and typically aligns to a OneFS protection group's boundaries.

### Block sharing phase

During the block sharing phase, the deduplication job calls into the shadow store library and deduplication infrastructure to perform the sharing of the blocks.



**Figure 7. SmartDedupe job block sharing phase**

Multiple request entries are consolidated into a single sharing request, which is processed by the block sharing phase and ultimately results in the deduplication of the common blocks. The file system searches for contiguous matching regions before and after the matching blocks in the sharing request; if any such regions are found, they are also shared. Blocks are shared by writing the matching data to a common shadow store and creating references from the original files to this shadow store.

### Index update phase

This phase populates the index table with the sampled and matching block information gathered during the previous three phases. After the deduplication job scans a file, OneFS might not find any matching blocks in other files on the cluster. Once a number of other files have been scanned, if a file continues to not share any blocks with other files on the cluster, OneFS removes the index entries for that file. This helps prevent OneFS from wasting cluster resources searching for unlikely matches. SmartDedupe scans each file in the specified dataset once, after which the file is marked, preventing subsequent deduplication jobs from rescanning the file until it has been modified.

## SmartDedupe management

### Introduction

There are two principal elements to managing deduplication in OneFS—the configuration of the SmartDedupe process itself and the scheduling and running of the deduplication job. These elements are described in the following sections.

### Configuring SmartDedupe

SmartDedupe works on datasets that are configured at the directory level, targeting all files and directories under each specified root directory. Multiple directory paths can be specified as part of the overall deduplication job configuration and scheduling.

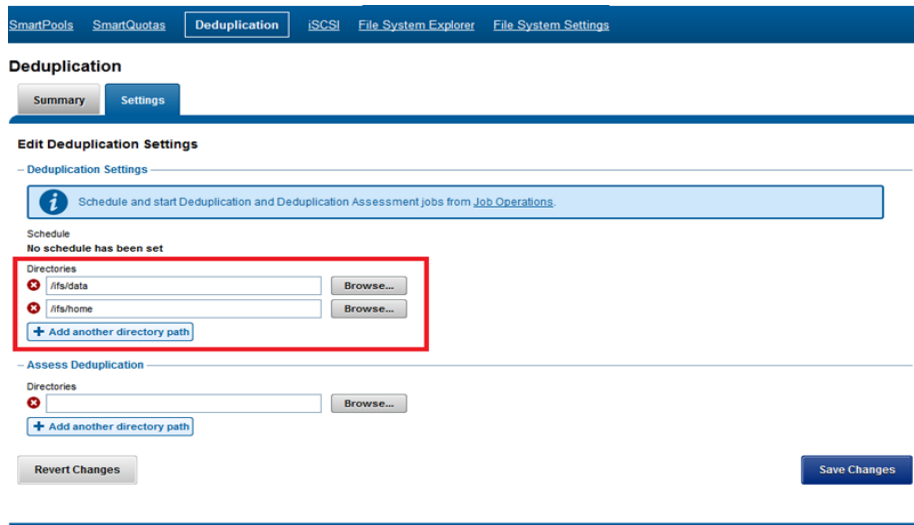


Figure 8. SmartDedupe configuration through the OneFS WebUI

**Note:** The permissions required to configure and modify deduplication settings are separate from those needed to run a deduplication job. For example, a user’s role must have job engine privileges to allow the user to run a deduplication job. However, to configure and modify deduplication configuration settings, the user must have the deduplication role privileges.

## Scheduling and running SmartDedupe

SmartDedupe can be run either on-demand (started manually) or on a predefined schedule, which is configured through the cluster management **Job Operations** section of the WebUI.

Figure 9. SmartDedupe job configuration and scheduling through the OneFS WebUI

Dell Technologies recommends scheduling and running deduplication during off-hours, when the rate of data change on the cluster is low. If clients are continually writing to files, the amount of space saved by deduplication will be minimal because the deduplicated blocks are constantly being removed from the shadow store.

For most clusters, after the initial deduplication job has completed, the recommendation is to run an incremental deduplication job once every two weeks.

## SmartDedupe monitoring and reporting

### Deduplication efficiency reporting

The amount of disk space currently saved by SmartDedupe can be determined by viewing the cluster capacity usage chart and deduplication reports summary table in the WebUI. The cluster capacity chart and deduplication reports can be found by going to **File System Management > Deduplication > Summary**.

## Deduplication

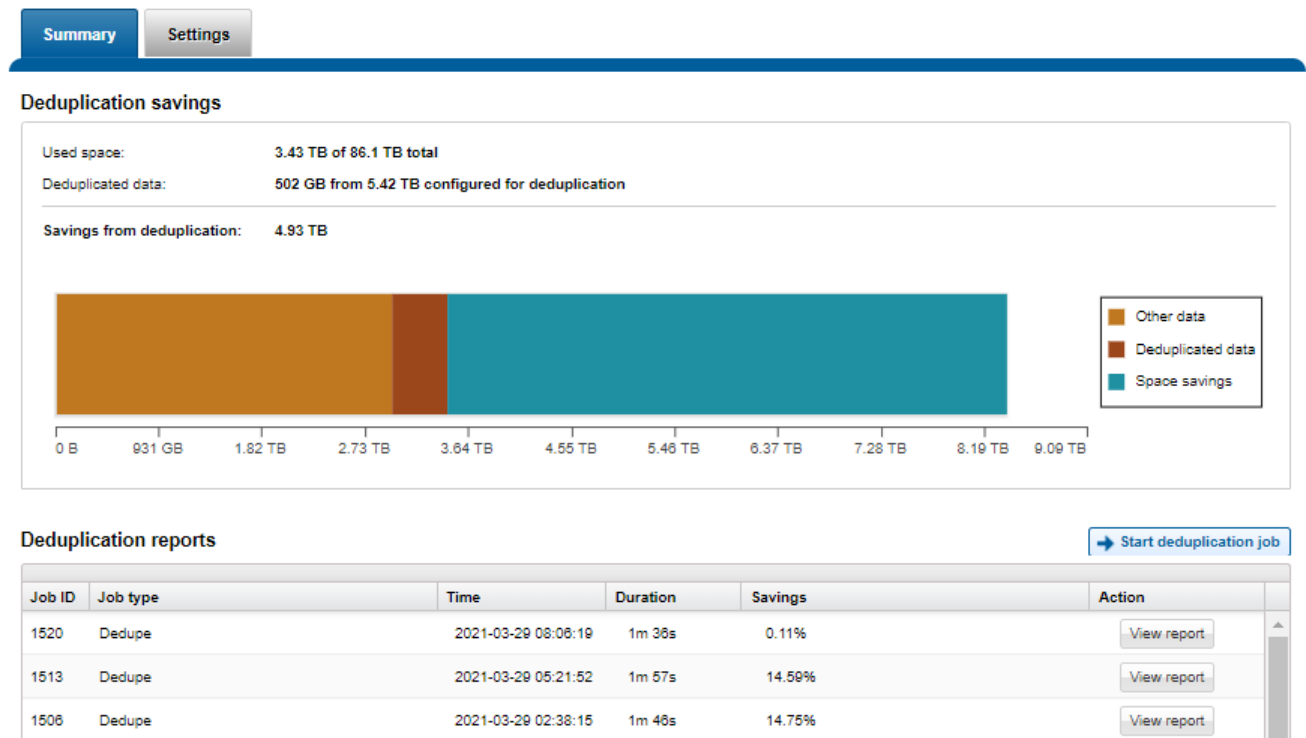


Figure 10. SmartDedupe cluster capacity savings WebUI chart

In addition, the deduplication job report overview field also displays the SmartDedupe savings as a percentage.

SmartDedupe space efficiency metrics are also provided through the `isi dedupe stats` CLI command:

```
# isi dedupe stats
    Cluster Physical Size: 86.14T
    Cluster Used Size: 3.43T
    Logical Size Deduplicated: 4.01T
    Logical Saving: 3.65T
    Estimated Size Deduplicated: 5.42T
    Estimated Physical Saving: 4.93T
```

Figure 11. SmartDedupe efficiency statistics from the CLI

The most comprehensive of the data reduction reporting CLI utilities is the `isi statistics data-reduction` command. For example:

```
# isi statistics data-reduction
                                Recent Writes Cluster Data Reduction
                                (5 mins)
-----
Logical data                      6.18M                      6.02T
Zero-removal saved                 0                      -
Deduplication saved               56.00k                 3.65T
Compression saved                 4.16M                  1.96G
Preprotected physical             1.96M                  2.37T
Protection overhead               5.86M                  910.76G
Protected physical                 7.82M                  3.40T
Zero removal ratio                1.00 : 1                -
Deduplication ratio               1.01 : 1                2.54 : 1
Compression ratio                 3.12 : 1                1.02 : 1
Data reduction ratio              3.15 : 1                2.54 : 1
Efficiency ratio                   0.79 : 1                1.77 : 1
-----
```

**Figure 12. Data reduction and storage efficiency statistics from the CLI**

The `Recent Writes` data to the left of the output provides precise statistics for the 5-minute period before the command was run. By contrast, the `Cluster Data Reduction` metrics on the right of the output are slightly less real-time but reflect the overall data and efficiencies across the cluster.

---

**Note:** In OneFS 9.1 and earlier, the right-hand column metrics are designated by the `Est` prefix, denoting an estimated value. However, in OneFS 9.2 and later, the `Logical data` and `Preprotected physical` metrics are tracked and reported accurately, rather than estimated.

---

The ratio data in each column is calculated from the values above it. For instance, to calculate the data reduction ratio, the `Logical data` (effective) is divided by the `Preprotected physical` (usable) value. From the output in the preceding figure, the calculation is:

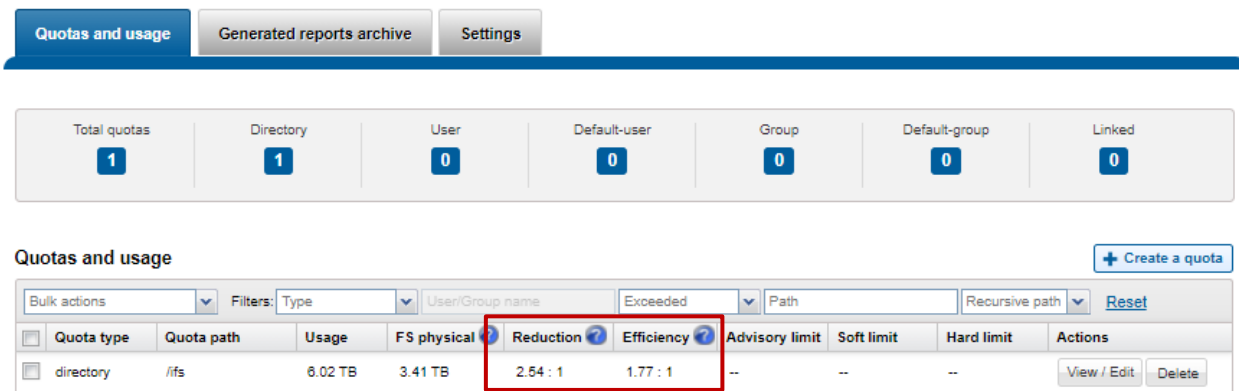
$6.02 / 2.37 = 1.76$  or a **data reduction ratio of 2.54:1**

Similarly, the `Efficiency ratio` is calculated by dividing the `Logical data` (effective) by the `Protected physical` (raw) value. From the output in the preceding figure, the calculation is:

$6.02 / 3.40 = 0.97$  or an **efficiency ratio of 1.77:1**

In OneFS 8.2.1 and later, SmartQuotas has been enhanced to report the capacity saving from deduplication, and data reduction in general, as a storage efficiency ratio. SmartQuotas reports efficiency as a ratio across a dataset as specified in the quota path field. The efficiency ratio is for the full quota directory and its contents, including any overhead, and reflects the net efficiency of compression and deduplication. On a cluster with licensed and configured SmartQuotas, this efficiency ratio can be easily viewed from the WebUI by going to **File System > SmartQuotas > Quotas and Usage**.

### SmartQuotas



**Figure 13. OneFS WebUI SmartQuotas quotas and usage status detailing efficiency and data reduction ratio**

Similarly, the same data can be accessed from the OneFS command line through the `isi quota quotas list` CLI command. For example:

```
# isi quota quotas list
Type      AppliesTo Path Snap Hard Soft Adv Used Reduction Efficiency
-----
directory DEFAULT /ifs No - - - 6.02T 2.54 : 1 1.77 : 1
-----
Total: 1
```

More detail, including both the physical (raw) and logical (effective) data capacities, is also available through the `isi quota quotas view <path> <type>` CLI command. For example:

```
# isi quota quotas view /ifs directory
Path: /ifs
Type: directory
Snapshots: No
Enforced: No
Container: No
Linked: No
Usage

Physical(With Overhead): 6.93T
FSPhysical(Deduplicated): 3.41T
FSLogical(W/O Overhead): 6.02T
AppLogical(ApparentSize): 6.01T
ShadowLogical: -
PhysicalData: 2.01T
Protection: 781.34G
Reduction(Logical/Data): 2.54 : 1
Efficiency(Logical/Physical): 1.77 : 1
```

To configure SmartQuotas for data efficiency reporting, create a directory quota at the top-level file system directory of interest, for example `/ifs`. Creating and configuring a directory quota is a simple procedure and can be performed from the WebUI, as follows:

Go to **File System > SmartQuotas > Quotas and Usage**, and select **Create a Quota**. Set **Quota type** to **Directory quota**, and add the preferred top-level path to report on. For **Quota accounting**, select **File system logical size**, and set **Quota limits** to **Track storage without specifying a storage limit**. Finally, click **Create quota** to confirm the configuration and activate the new directory quota.

The screenshot shows the 'Create a quota' dialog box in the OneFS WebUI. The dialog is titled 'Create a quota' and has a 'Help' icon. It is divided into several sections: 'Settings', 'Quota accounting', and 'Quota limits'. In the 'Settings' section, 'Quota type' is set to 'Directory quota' and 'Path' is '/ifs'. In the 'Quota accounting' section, 'Application logical size' is selected. In the 'Quota limits' section, 'Track storage without specifying a storage limit' is selected. The 'Create quota' button is highlighted in blue.

**Figure 14. OneFS WebUI SmartQuotas directory quota configuration**

The efficiency ratio is a single, current-in time efficiency metric that is calculated per quota directory and includes the sum of SmartDedupe plus inline data reduction. This is in contrast to a history of stats over time, as reported in the `isi statistics data-reduction` CLI command output, previously described. As such, the efficiency ratio for the entire quota directory reflects what is actually there.

The OneFS WebUI cluster dashboard also displays a storage efficiency tile, which shows physical and logical space utilization histograms and reports the capacity saving from inline data reduction as a storage efficiency ratio. This dashboard view is displayed by default when opening the OneFS WebUI in a browser and can be easily accessed by going to **File System > Dashboard > Cluster Overview**.

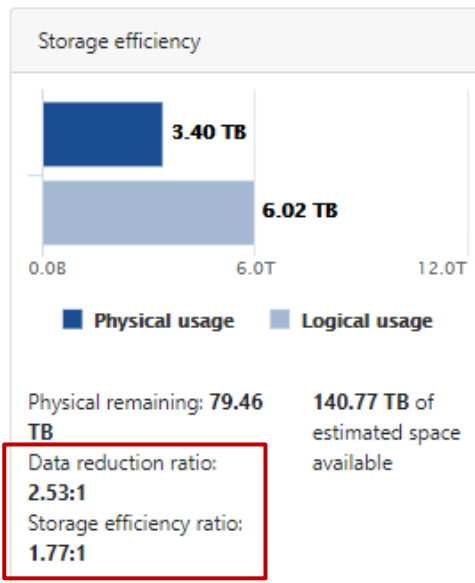


Figure 15. OneFS WebUI cluster status dashboard – storage efficiency summary tile

Similarly, the `isi status` CLI command output includes a `Data Reduction` field:

```
# isi status
Cluster Name: f8101
Cluster Health: [ OK ]
Data Reduction: 2.54 : 1
Storage Efficiency: 1.77 : 1
Cluster Storage: HDD
Size: 0 (0 Raw)
VHS Size: 3.2T
Used: 0 (n/a)
Avail: 0 (n/a)
```

```
SSD Storage
82.9T (86.1T Raw)
3.4T (4%)
79.5T (96%)
```

ID	IP Address	Health  DASR	Throughput (bps) In Out Total	HDD Storage Used / Size	SSD Storage Used / Size
1	10.245.110.69	OK	0  0  0	(No Storage HDDs)	878G/20.7T( 4%)
2	10.245.110.70	OK	0 73.9k 73.9k	(No Storage HDDs)	879G/20.7T( 4%)
3	10.245.110.71	OK	0  149k  149k	(No Storage HDDs)	879G/20.7T( 4%)
4	10.245.110.72	OK	0  494k  494k	(No Storage HDDs)	879G/20.7T( 4%)
Cluster Totals:			0  717k  717k	0/ 0( n/a)	3.4T/82.9T( 4%)

Health Fields: D = Down, A = Attention, S = Smartfailed, R = Read-Only

### SmartDedupe job progress

The Job Engine parallel execution framework provides comprehensive run time and completion reporting for the deduplication job.

While SmartDedupe is underway, job status is available at a glance in the progress column in the active jobs table. This information includes the number of files, directories, and blocks that have been scanned, skipped, and sampled, and any errors that may have been encountered.



Additional progress information is provided in an Active Job Details status update, which includes an estimated completion percentage based on the number of logical inodes (LINs) that have been counted and processed.

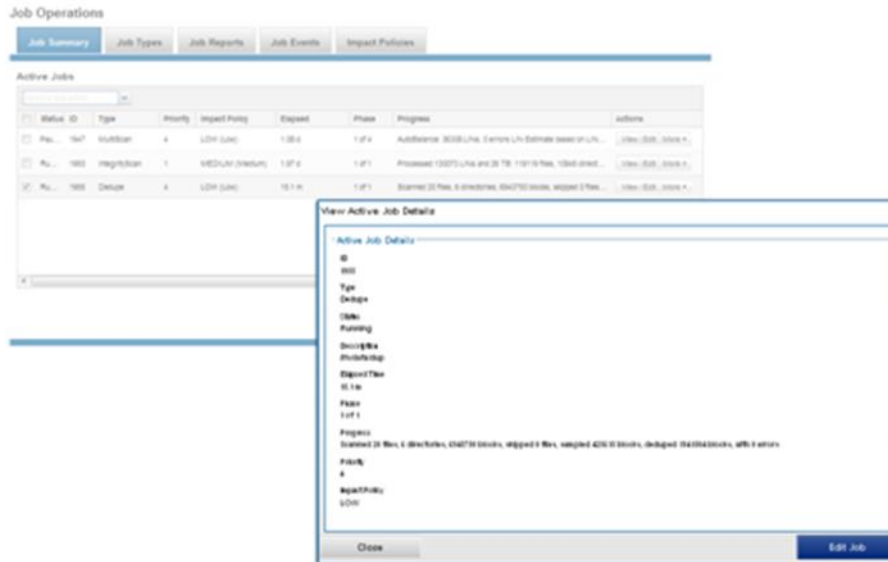


Figure 16. Example of active job status update

### SmartDedupe job reports

Once the SmartDedupe job has run to completion, or has been terminated, a full deduplication job report is available. This report can be accessed from the WebUI by going to **Cluster Management > Job Operations > Job Reports** and selecting **View Details**, under the **Action** column, for the job.

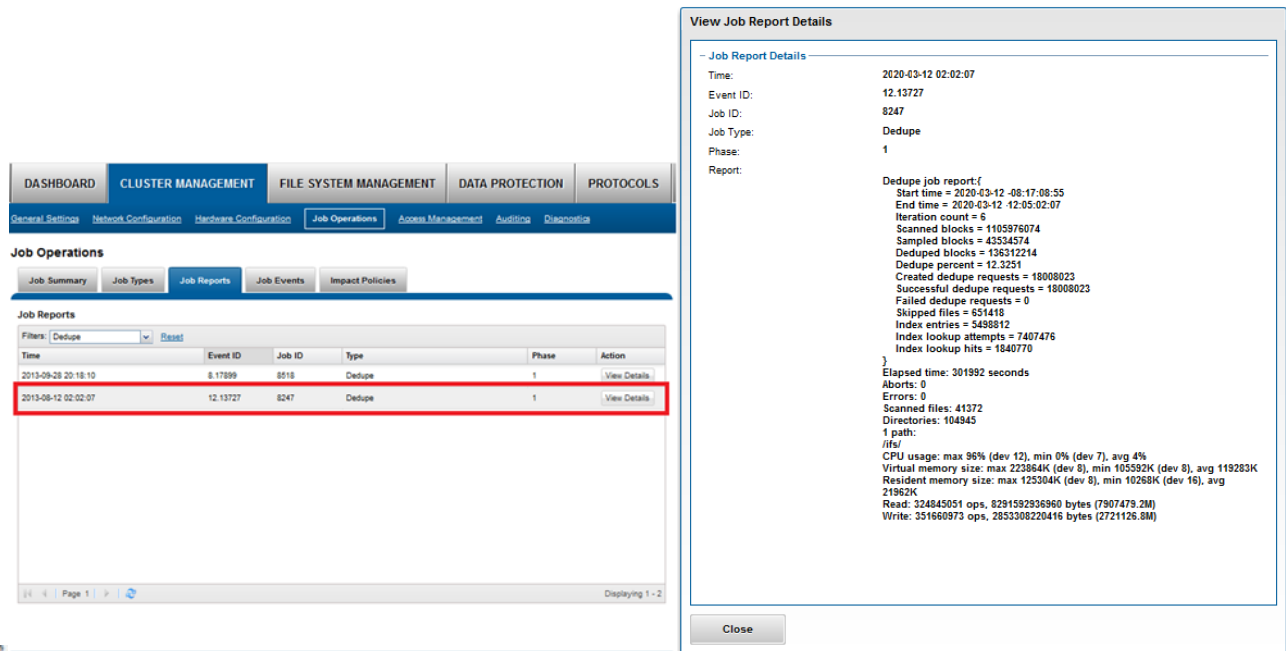


Figure 17. Example of WebUI deduplication job report

The job report contains the following relevant deduplication metrics.

**Table 1. Deduplication job report statistics**

Report field	Description of metric
Start time	When the deduplication job started.
End time	When the deduplication job finished.
Scanned blocks	Total number of blocks scanned under configured path or paths.
Sampled blocks	Number of blocks that OneFS created index entries for.
Created dedupe requests	Total number of deduplication requests created. A deduplication request gets created for each matching pair of data blocks. For example, three data blocks all match, two requests are created: One request to pair file1 and file2 together, the other request to pair file2 and file3 together.
Successful dedupe requests	Number of deduplication requests that completed successfully.
Failed dedupe requests	Number of deduplication requests that failed. If a deduplication request fails, it does not mean that the SmartDedupe job also failed. A deduplication request can fail for any number of reasons. For example, the file might have been modified since it was sampled.
Skipped files	Number of files that were not scanned by the deduplication job. The primary reason is that the file has already been scanned and has not been modified since. Another reason for a file to be skipped is if it is smaller than 32 KB. Such files are considered too small and do not provide enough space-saving benefit to offset the fragmentation they will cause.
Index entries	Number of entries that currently exist in the index.
Index lookup attempts	Cumulative total number of lookups that have been done by prior and current deduplication jobs. A lookup is when the deduplication job attempts to match a block that has been indexed with a block that has not been indexed.
Index lookup hits	Total number of lookup hits that have been done by earlier deduplication jobs plus the number of lookup hits done by this deduplication job. A hit is a match of a sampled block with a block in index.

Dedupe job reports are also available from the CLI through the `isi job reports view <job_id>` command.

---

**Note:** From a processing and reporting stance, the Job Engine considers the deduplication job to consist of a single process or phase. The Job Engine events list reports that Dedupe Phase1 has ended and succeeded. This indicates that an entire SmartDedupe job, including all four internal deduplication phases (sampling, duplicate detection, block sharing, and index update), has successfully completed.

---

For example:

```
# isi job events list --job-type dedupe
Time                Message
-----
2020-02-01T13:39:32 Dedupe[1955] Running
2020-02-01T13:39:32 Dedupe[1955] Phase 1: begin dedupe
2020-02-01T14:20:32 Dedupe[1955] Phase 1: end dedupe
2020-02-01T14:20:32 Dedupe[1955] Phase 1: end dedupe
2020-02-01T14:20:32 Dedupe[1955] Succeeded
```

**Figure 18. Example of CLI dedupe job events list**

For deduplication reporting across multiple OneFS clusters, SmartDedupe is also integrated with InsightIQ cluster reporting and analysis product. A report detailing the space savings delivered by deduplication is available through the InsightIQ File Systems Analytics module.

### Space savings estimation with the SmartDedupe assessment job

To complement the deduplication job, a dry-run deduplication assessment job is also provided to help estimate the amount of space savings that can be achieved by running deduplication on a particular directory or set of directories. The assessment job reports a total potential space savings. The assessment does not differentiate the case of a fresh run from the case where a previous deduplication job has already done some sharing on the files in that directory. The assessment job does not provide the incremental differences between instances of this job. Dell Technologies recommends that you run the assessment job once on a specific directory before starting a deduplication job on that directory.

The assessment job runs similarly to the deduplication job but uses a separate configuration. It also does not require a product license and can be run before SmartDedupe is purchased to determine whether deduplication is appropriate for a particular dataset or environment.

**Deduplication**

Summary Settings

**Edit Deduplication Settings**

- Deduplication Settings

*Schedule and start Deduplication and Deduplication Assessment jobs from [Job Operations](#).*

Schedule  
No schedule has been set

Directories  
✘ /ifs/data

- Assess Deduplication

Directories  
✘ /ifs/home

**Figure 19. Deduplication assessment job configuration**

The assessment job uses a separate index table. For efficiency, the assessment job also samples fewer candidate blocks than the main deduplication job and does not actually perform deduplication. Using the sampling and consolidation statistics, the job provides a report that estimates the total deduplication space savings in bytes.

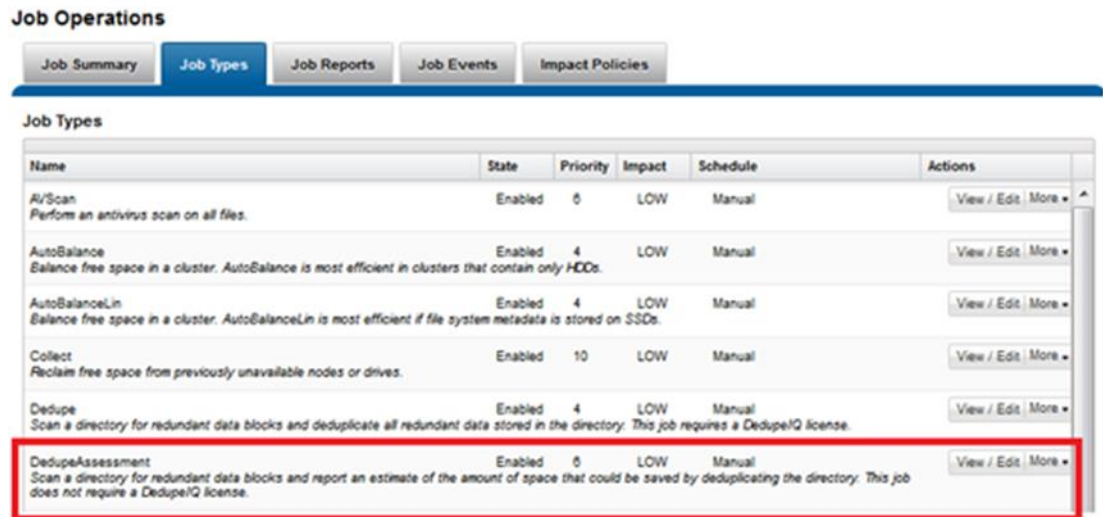


Figure 20. Deduplication assessment job control through the OneFS WebUI

## Performance with SmartDedupe

Deduplication is a compromise. To gain increased levels of storage efficiency, additional cluster resources (CPU, memory, and disk I/O) are used to find and share common data blocks.

Another important performance impact consideration with deduplication is the potential for data fragmentation. After deduplication, files that previously enjoyed contiguous on-disk layout often have chunks spread across less optimal file system regions. This can lead to slightly increased latencies when accessing these files directly from disk, rather than from cache. To help reduce this risk, SmartDedupe does not share blocks across node pools or data tiers and does not attempt to deduplicate files smaller than 32 KB. On the other end of the spectrum, the largest contiguous region that is matched is 4 MB.

Because deduplication is a data efficiency product rather than performance enhancing tool, usually the consideration is around cluster impact management. This consideration is from both the client data access performance front, because, by design, multiple files share common data blocks, and also from the deduplication job processing perspective, because additional cluster resources are consumed to detect and share commonality.

The first deduplication job run often takes a substantial amount of time to run because it must scan all files under the specified directories to generate the initial index and then create the appropriate shadow stores. However, deduplication job performance typically improves significantly on the second and subsequent job runs (incrementals), once the initial index and the bulk of the shadow stores have already been created.

If incremental deduplication jobs do take a long time to complete, this is most likely indicative of a dataset with a high rate of change. If a deduplication job is paused or interrupted, it automatically resumes the scanning process from where it left off.

As mentioned previously, deduplication is a long running process that involves multiple job phases that are run iteratively. SmartDedupe typically processes around 1 TB of data per day, per node.

## SmartDedupe licensing

SmartDedupe is included as a core component of OneFS but requires a valid product license key to be activated. This license key can be purchased through your Dell account team. An unlicensed cluster shows a SmartDedupe warning until a valid product license is purchased and applied to the cluster.

License keys can be easily added through the **Activate License** section of the OneFS WebUI, accessed by going to **Cluster Management > Licensing**.

---

**Note:** The SmartDedupe dry-run estimation job can be run without any licensing requirements, allowing you to assess the potential space savings that a dataset might yield before deciding whether to purchase the full product.

---

## Deduplication efficiency

Deduplication can significantly increase the storage efficiency of data. However, the actual space savings will vary depending on the specific attributes of the data itself. As noted previously, the deduplication assessment job can be run to help predict the likely space savings that deduplication would provide on a given dataset.

Virtual machines files often contain duplicate data, much of which is rarely modified. Deduplicating similar operating system type virtual machine images (for example VMware VMDK files, that have been block-aligned) can significantly decrease the amount of storage space consumed. However, as noted previously, the potential for performance degradation as a result of block sharing and fragmentation should be carefully considered first.

SmartDedupe does not deduplicate across files that have different protection settings. For example, if two files share blocks, but file1 is parity-protected at +2:1, and file2 has its protection set at +3, SmartDedupe will not attempt to deduplicate them. This ensures that all files and their constituent blocks are protected as configured. Additionally, SmartDedupe does not deduplicate files that are stored on different SmartPools storage tiers or node-pools. For example, if file1 and file2 are stored on tier 1 and tier 2 respectively, and tier1 and tier2 are both protected at 2:1, OneFS will not deduplicate them. This helps guard against performance asynchronicity, where some of a file's blocks could live on a different tier, or class of storage, from the others.

The following table shows some examples of typical space reclamation levels that have been achieved with SmartDedupe.

---

**Note:** These deduplication space savings values are provided solely as rough guidance. Because no two datasets are alike (unless they are replicated), actual results can vary considerably from these examples.

---

**Table 2. Typical workload space savings with SmartDedupe**

Workflow/data type	Typical space savings
Virtual Machine Data	35%
Home Directories / File Shares	25%
Email Archive	20%
Engineering Source Code	15%
Media Files	10%

## SmartDedupe best practices and considerations

### SmartDedupe best practices

For optimal cluster performance, Dell Technologies recommends observing the following SmartDedupe best practices:

- Deduplication is most effective when applied to datasets with a low rate of change—for example, archived data.
- Enable SmartDedupe to run at subdirectory levels below `/ifs`.
- Avoid adding more than 10 subdirectory paths to the SmartDedupe configuration policy,
- SmartDedupe is ideal for home directories, departmental file shares, and warm and cold archive datasets.
- Run SmartDedupe against a smaller sample dataset first to evaluate performance impact compared to space efficiency.
- Schedule deduplication to run during the cluster's low-usage hours—that is, overnight, weekends, and so on.
- After the initial deduplication job has completed, schedule incremental deduplication jobs to run every two weeks or so, depending on the size and rate of change of the dataset.
- Always run SmartDedupe with the default low-impact Job Engine policy.
- Run the deduplication assessment job on a single root directory at a time. If multiple directory paths are assessed in the same job, you will not be able to determine which directory should be deduplicated.
- When replicating deduplicated data, to avoid running out of space on target, verify that the logical data size (that is, the amount of storage space saved plus the actual storage space consumed) does not exceed the total available space on the target cluster.
- Run a deduplication job on an appropriate dataset before enabling a snapshots schedule.
- Where possible, perform any snapshot restores (reverts) before running a deduplication job, and run a deduplication job directly after restoring a prior snapshot version.

## SmartDedupe considerations

As discussed earlier, deduplication is not free. There is always trade-off between cluster resource consumption (CPU, memory, disk), the potential for data fragmentation and the benefit of increased space efficiency.

- Because deduplication trades cluster performance for storage capacity savings, SmartDedupe is not ideally suited for heavily trafficked data, or high-performance workloads.
- Depending on an application's I/O profile and the effect of deduplication on the data layout, read and write performance and overall space savings can vary considerably.
- SmartDedupe does not permit block sharing across different hardware types or node pools to reduce the risk of performance asymmetry.
- SmartDedupe does not share blocks across files with different protection policies applied.
- OneFS metadata, including the deduplication index, is not deduplicated.
- Deduplication is a long-running process that involves multiple job phases that are run iteratively.
- SmartDedupe does not attempt to deduplicate files smaller than 32 KB.
- Dedupe job performance typically improves significantly on the second and subsequent job runs, once the initial index and the bulk of the shadow stores have already been created.
- SmartDedupe does not deduplicate the data stored in a snapshot. However, snapshots of deduplicated data can be created.
- If deduplication is enabled on a cluster that already has a significant amount of data stored in snapshots, deduplication will take time to affect the snapshot data. Newly created snapshots will contain deduplicated data, but older snapshots will not.
- SmartDedupe deduplicates common blocks within the same file, resulting in even better data efficiency.
- In general, additional capacity savings may not warrant the overhead of running SmartDedupe on node pools with inline deduplication enabled.
- Deduplication of data contained within a writable snapshot is not supported in OneFS 9.3 or later.

## SmartDedupe and OneFS feature integration

### SyncIQ replication and SmartDedupe

When deduplicated files are replicated to another cluster through SyncIQ or backed up to a tape device, the deduplicated files are inflated (or rehydrated) back to their original size because they no longer share blocks on the target cluster. However, once replicated data has landed, SmartDedupe can be run on the target cluster to provide the same space efficiency benefits as on the source.

Shadow stores are not transferred to target clusters or backup devices. Thus, deduplicated files do not consume less space than non-deduplicated files when they are

replicated or backed up. To avoid running out of space on target clusters or tape devices, verify that the total amount of storage space saved, and storage space consumed, does not exceed the available space on the target cluster or tape device. To reduce the amount of storage space consumed on a target cluster, you can configure deduplication for the target directories of your replication policies. Although such configuration will deduplicate data on the target directory, it will not allow SyncIQ to transfer shadow stores. Deduplication is still performed post-replication, by means of a deduplication job running on the target cluster.

## Backup and SmartDedupe

Because files are backed up as if the files were not deduplicated, backup and replication operations are not faster for deduplicated data. You can deduplicate data while the data is being replicated or backed up.

---

**Note:** OneFS NDMP backup data will not be deduplicated unless the backup vendor's DMA software provides deduplication. However, compression is often provided natively by the backup tape or VTL device.

---

## Snapshots and SmartDedupe

SmartDedupe does not deduplicate the data stored in a snapshot. However, you can create snapshots of deduplicated data. If a snapshot is taken of a deduplicated directory, and then the contents of that directory are modified, the shadow stores are transferred to the snapshot over time. Thus, running deduplication before enabling snapshots saves more space on a cluster.

If deduplication is enabled on a cluster that already has a significant amount of data stored in snapshots, deduplication will take time to affect the snapshot data. Newly created snapshots will contain deduplicated data, but older snapshots will not.

It is also good practice to revert a snapshot before running a deduplication job. Restoring a snapshot will cause many of the files on the cluster to be overwritten. Any deduplicated files are reverted to normal files if they are overwritten by a snapshot revert. However, once the snapshot revert is completed, deduplication can be run on the directory again and the resulting space savings will persist on the cluster.

Deduplication of writable snapshot data is not supported. SmartDedupe ignores the files under writable snapshots.

## SmartLock and SmartDedupe

SmartDedupe is also fully compatible with SmartLock, the OneFS data retention and compliance solution. SmartDedupe delivers storage efficiency for immutable archives and write once, read many (or WORM) protected datasets.

## SmartQuotas and SmartDedupe

OneFS SmartQuotas accounts for deduplicated files as if they consumed both shared and unshared data. From the quota side, deduplicated files appear no differently than regular files to standard quota policies. However, if the quota is configured to include data-protection overhead, the quota will not account for the additional space used by the shadow store.



## SmartPools and SmartDedupe

SmartDedupe does not deduplicate files that span SmartPools node pools or tiers, or that have different protection levels, access patterns, or caching configurations set. This is to avoid potential performance or protection asymmetry, which could occur if portions of a file live on different classes of storage.

However, a deduplicated file that is moved by SmartPools to a different pool or tier retains the shadow references to the shadow store on the original pool. Retaining the shadow references breaks the rule for deduplicating across different disk pool policies but is less impactful than rehydrating files that are moved. Further deduplication activity on that file no longer references any blocks in the original shadow store. The file must be deduplicated against other files in the same disk pool policy. If the file had not yet been deduplicated, the deduplication index might have knowledge about the file and assess that it is on the original pool. This will be discovered and corrected when a match is made against blocks in the file.

Because the moved file has already been deduplicated, the deduplication index has knowledge of the shadow store only. Because the shadow store has not moved, it will not cause problems for further matching. However, if the shadow store is moved as well (but not both files), a similar situation occurs and the SmartDedupe job will discover this and purge knowledge of the shadow store from the deduplication index.

## In-line compression and SmartDedupe

SmartDedupe post-process deduplication and inline compression (currently available on the PowerScale F900, F810, F600, F200, H700/7000, H5600, and A300/3000 platforms) are compatible with each other. Inline compression can compress OneFS shadow stores. However, before SmartDedupe can process compressed data, the SmartDedupe job must first decompress the data to perform deduplication. In general, additional capacity savings might not warrant the overhead of running SmartDedupe on node pools with inline deduplication enabled.

## In-line deduplication and SmartDedupe

While OneFS has offered a native file system deduplication solution for several years, until OneFS 8.2.1 this deduplication was always accomplished by scanning the data after it had been written to disk, or post-process. With inline data reduction, deduplication is performed in real time as data is written to the cluster. Storage efficiency is achieved by scanning the data for identical blocks as it is received and then eliminating the duplicates using shadow stores.

Because inline deduplication and SmartDedupe use different hashing algorithms, the indexes for each are not shared directly. However, each deduplication solution can use the work performed by the other. For instance, if SmartDedupe writes data to a shadow store, when those blocks are read, the read hashing component of inline deduplication will see those blocks and index them.

When a match is found, inline deduplication performs a byte-by-byte comparison of each block to be shared to avoid the potential for a hash collision. Data is prefetched before the byte-by-byte check and then compared against the L1 cache buffer directly, avoiding unnecessary data copies and adding minimal overhead. Once the matching blocks have been compared and verified as identical, they are shared by writing the matching data to a common shadow store and creating references from the original files to this shadow store.

Inline deduplication samples every whole block written and handles each block independently, so it can aggressively locate block duplicity. If a contiguous run of matching blocks is detected, inline deduplication will merge the results into regions and process them efficiently.

In-line deduplication also detects deduplication opportunities from the read path, and blocks are hashed as they are read into L1 cache and inserted into the index. If an existing entry exists for that hash, inline deduplication knows there is a block sharing opportunity between the block it just read and the one previously indexed. It combines that information and queues a request to an asynchronous deduplication worker thread. As such, it is possible to deduplicate a dataset purely by reading it all. To help mitigate the performance impact, all the hashing is performed out-of-band in the prefetch path, rather than in the latency-sensitive read path.

### Small File Storage Efficiency (SFSE) and SmartDedupe

SFSE is mutually exclusive to all the other shadow store consumers (file clones, inline deduplication, SmartDedupe). Files can either be packed with SFSE or cloned/deduplicated, but not both.

Inlined files (small files with their data stored in the inode) are not deduplicated, and non-inlined datafiles that are once deduplicated will not inline afterwards.

### InsightIQ and SmartDedupe

InsightIQ, the Dell PowerScale multi-cluster reporting and trending analytics suite, is integrated with SmartDedupe. Included in the data provided by the File Systems Analytics module is a report detailing the space savings efficiency delivered by deduplication.

## SmartDedupe use cases

As previously noted, an enterprise's data typically contains substantial quantities of redundant information. Home directories, file shares, and data archives are examples of workloads that consistently yield solid deduplication results. Each time multiple employees save a spreadsheet, document, or email attachment, the same file is stored in full multiple times, taking up valuable disk capacity. SmartDedupe is typically used in the following ways:

### Example A: File shares and home directory deduplication

By architecting and configuring home directory and file share repositories under unifying top-level directories (for example, `/ifs/home` and `/ifs/data`, respectively), an organization can easily and efficiently configure and run deduplication against these datasets.

Performance-wise, home directories and file shares are typically mid-tier workloads, usually involving concurrent access with a reasonable balance of read and write and data and metadata operations. As such, they make great candidates for SmartDedupe.

SmartDedupe should ideally be run during periods of low cluster load and client activity (nights and weekends, for example). Once the initial job has completed, the deduplication job can be scheduled to run every two weeks or so, depending on the data's rate of change.

**Example B: Storage-efficient archiving**

SmartDedupe is an ideal solution for large, infrequently accessed content repositories. Examples of these include digital asset management workloads, seismic data archives for energy exploration, document management repositories for legal discovery, compliance archives for financial or medical records, and so on.

These are all excellent use cases for deduplication because the performance requirements are typically low and biased towards metadata operations, and there are typically numerous duplications of data. As such, trading system resources for data efficiency produces significant, tangible benefits to the bottom line. SmartDedupe is also ideal for SmartLock-protected immutable archives and other WORM datasets, typically delivering attractive levels of storage efficiency.

For optimal results, where possible, ensure that archive data is configured with the same level of protection. For data archives that are frequently scanned or indexed, metadata read acceleration is the recommended metadata SSD strategy.

**Example C: Disaster recovery target cluster deduplication**

For performance-oriented environments that would prefer not to run deduplication against their primary dataset, the typical approach is to deduplicate the read-only data replica on their target, or disaster recovery (DR), cluster.

Once the initial deduplication job has successfully completed, subsequent incremental deduplication jobs can be scheduled to run soon after completion of each SyncIQ replication job, or as best fits the rate of data change and frequency of cluster replication.

## SmartDedupe and OneFS storage utilization

SmartDedupe is one of several components of OneFS that enables a cluster to deliver a very high level of raw disk utilization. Another major storage efficiency attribute is the way that OneFS natively manages data protection in the file system. Unlike most file systems that rely on hardware RAID, OneFS protects data at the file level and, using software-based erasure coding, allows most customers to enjoy raw disk space utilization levels in the 80 percent range or higher. This is in contrast to the industry mean of around 50-60 percent raw disk capacity utilization. SmartDedupe serves to further extend this storage efficiency headroom, bringing an even more compelling and demonstrable TCO advantage to primary file-based storage.

## Conclusion

Until now, traditional deduplication implementations have typically been expensive, limited in scale, confined to secondary storage, and administratively complex.

SmartDedupe integration with the PowerScale scale-out NAS architecture delivers on the promise of simple data efficiency at scale by providing significant storage cost savings, without sacrificing ease of use or data protection.

With its simple, powerful interface, and intelligent default settings, SmartDedupe is easy to estimate, configure and manage, and provides enterprise data efficiency within a single,

## Conclusion

highly extensible storage pool. Scalability to petabytes and the ability to add new capacity and new technologies, while retaining older capacity in the same system, means strong investment protection. Integration with OneFS core functions eliminates data risks and gives the user control over what system resources are allocated to data movement.

### **TAKE THE NEXT STEP**

Contact your Dell sales representative or authorized reseller to learn more about how PowerScale NAS storage solutions can benefit your organization.

Visit [Dell PowerScale](#) to compare features and get more information.