WHITE PAPER

NEXT GENERATION STORAGE EFFICIENCY WITH DELL EMC ISILON SMARTDEDUPE

Abstract

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

December 2019

Revisions

Version	Date	Comment
1.0	November 2013	Initial release for OneFS 7.1
2.0	June 2014	Updated for OneFS 7.1.1
3.0	November 2014	Updated for OneFS 7.2
4.0	June 2015	Updated for OneFS 7.2.1
5.0	November 2015	Updated for OneFS 8.0
6.0	September 2016	Updated for OneFS 8.0.1
7.0	April 2017	Updated for OneFS 8.1
8.0	November 2017	Updated for OneFS 8.1.1
9.0	February 2019	Updated for OneFS 8.1.3
10.0	April 2019	Updated for OneFS 8.2
11.0	August 2019	Updated for OneFS 8.2.1
12.0	December 2019	Updated for OneFS 8.2.2

Acknowledgements

This paper was produced by the following:

Author: Nick Trimbee

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.

TABLE OF CONTENTS

Introduction

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 an email attachment is stored by multiple employees, many of the same files are stored or replicated. This leads to multiple copies of the same data, which 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 Isilon's industry-leading data efficiency solutions, and a key ingredient for organizations that wish to maintain a competitive edge.

Intended Audience

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

📖 Please refer to the OneFS Technical Overview white paper for further details on the OneFS architecture.

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

📖 More information on OneFS commands and feature configuration is available in the OneFS Administration Guide.

Isilon SmartDedupe

Overview

Isilon's SmartDedupe product 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.

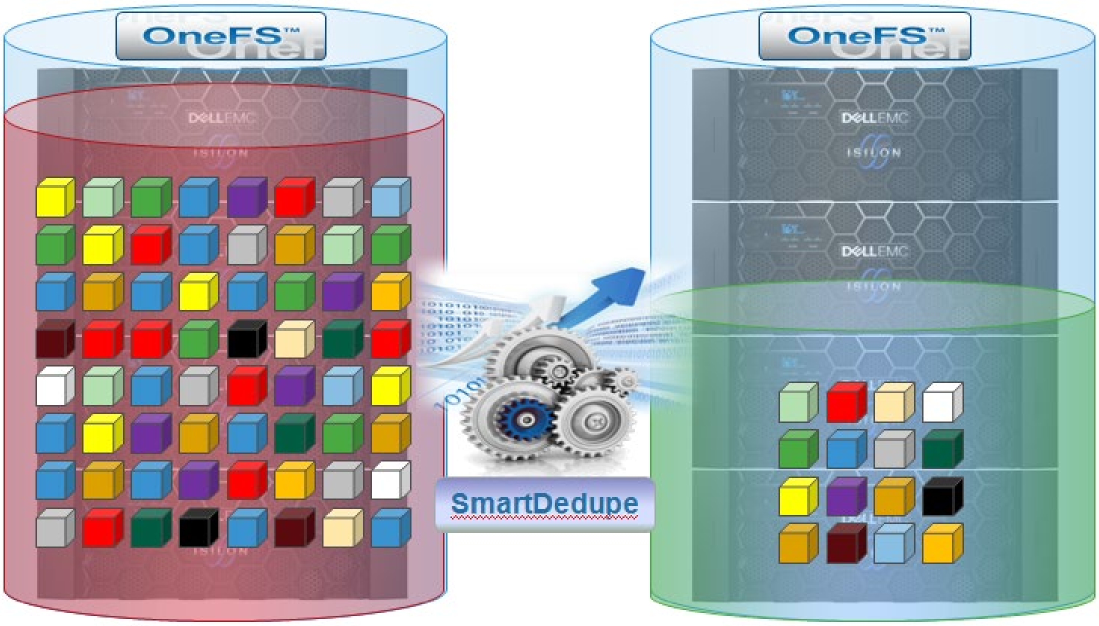


Figure 1: Isilon Storage Efficiency with SmartDedupe

After duplicate blocks are discovered, 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.

With post-process deduplication, new data is first stored on the storage device and then a subsequent process analyzes the data looking for commonality. This means that the initial file-write or modify performance is not impacted, since no additional computation is required in the write path.

SmartDedupe Architecture

The Dell EMC Isilon SmartDedupe architecture is comprised of five principle modules:

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

The SmartDedupe control path comprises the Isilon OneFS Web Management Interface (WebUI), command line interface (CLI) and RESTful platform API, and 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 in more detail below.

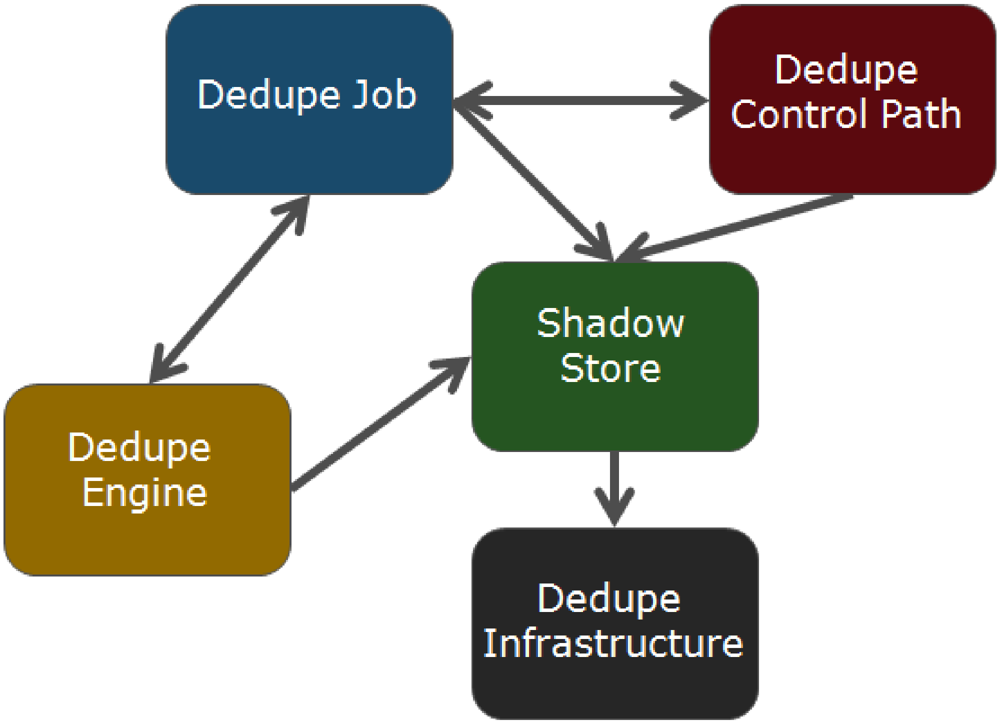


Figure 2: Isilon SmartDedupe Modular Architecture

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 8KB data block in the sampled set.

When SmartDedupe runs for the first time, it scans the data set 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 already existing data block and the new, duplicate data block is released.

Hash computation and comparison is only utilized during the sampling phase. The deduplication job phases are covered in detail below. 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 sharable 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 Isilon 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 cannot be accessed via a pathname. A shadow store typically grows to a maximum size of 2GB (or about 256K 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.

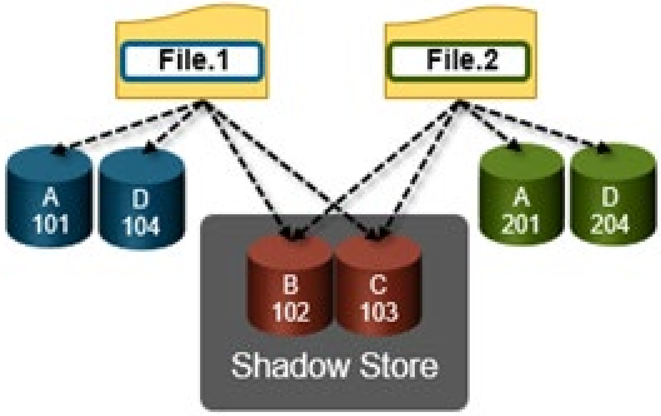
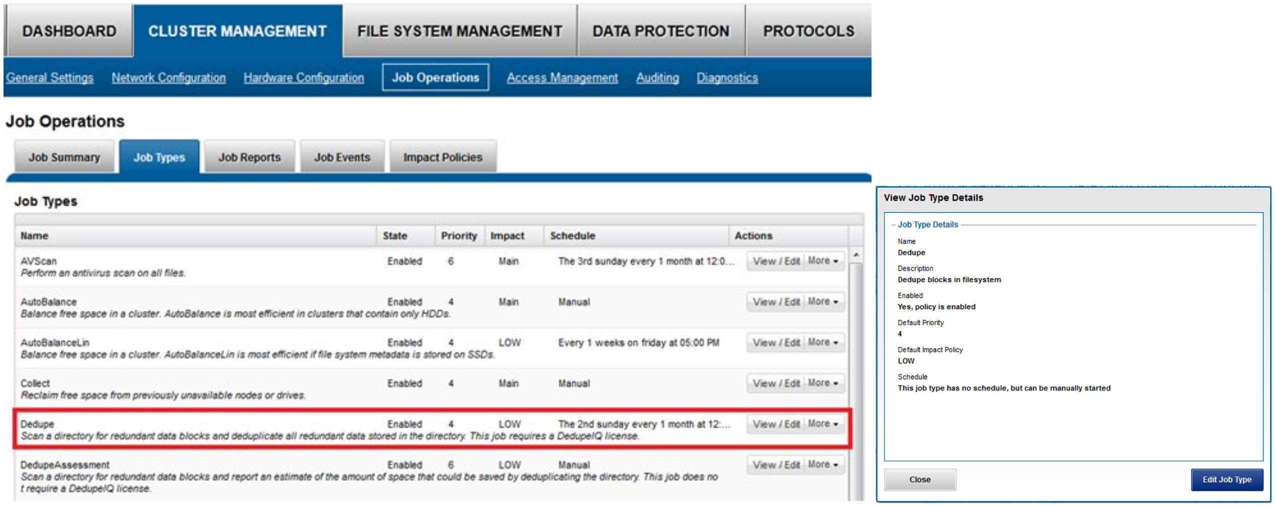


Figure 3: OneFS Duplicate Block Sharing.

Deduplication Job & Infrastructure

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

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



Job Operations

Job Types

Name	State	Priority	Impact	Schedule	Actions
AVScan *Perform an antivirus scan on all files.*	Enabled	6	Main	The 3rd sunday every 1 month at 12:0...	View / Edit More ▾
AutoBalance *Balance free space in a cluster. AutoBalance is most efficient in clusters that contain only HDDs.*	Enabled	4	Main	Manual	View / Edit More ▾
AutoBalanceLin *Balance free space in a cluster. AutoBalanceLin is most efficient if file system metadata is stored on SSDs.*	Enabled	4	LOW	Every 1 weeks on friday at 05:00 PM	View / Edit More ▾
Collect *Reclaim free space from previously unavailable nodes or drives.*	Enabled	4	Main	Manual	View / Edit More ▾
Dedupe *Scan a directory for redundant data blocks and deduplicate all redundant data stored in the directory. This job requires a DedupeIQ license.*	Enabled	4	LOW	The 2nd sunday every 1 month at 12:...	View / Edit More ▾
DedupeAssessment *Scan a directory for redundant data blocks and report an estimate of the amount of space that could be saved by deduplicating the directory. This job does no t require a DedupeIQ license.*	Enabled	6	LOW	Manual	View / Edit More ▾

View Job Type Details

Job Type Details

Name
Dedupe

Description
Dedupe blocks in filesystem

Enabled
Yes, policy is enabled

Default Priority
4

Default Impact Policy
LOW

Schedule
This job type has no schedule, but can be manually started

Figure 4: SmartDedupe Job Control via 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 Isilon 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 will automatically ignore (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 dedupe infrastructure, comprise the following four phases:

- Sampling
- Duplicate Detection
- Block Sharing
- Index Update

These four phases are described in more detail below.

Because the SmartDedupe job is typically long running, each of the phases are executed 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 dedupe job progress tracking is available via the OneFS Job Engine reporting infrastructure.

Sampling Phase

In the sampling phase, SmartDedupe performs a tree-walk of the configured data set in order to collect deduplication candidates for each file. 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 flagged and won’t be re-scanned until it has been modified. This drastically improves the performance of subsequent deduplication jobs.

Duplicate Detection Phase

During the duplicate, or commonality, detection phase, the dedupe job scans the index table for fingerprints (or hashes) that match those of the candidate blocks. If the index entries of two files match, a request entry is generated. In order 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 which 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 dedupe infrastructure to perform the sharing of the blocks. 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 too will be 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

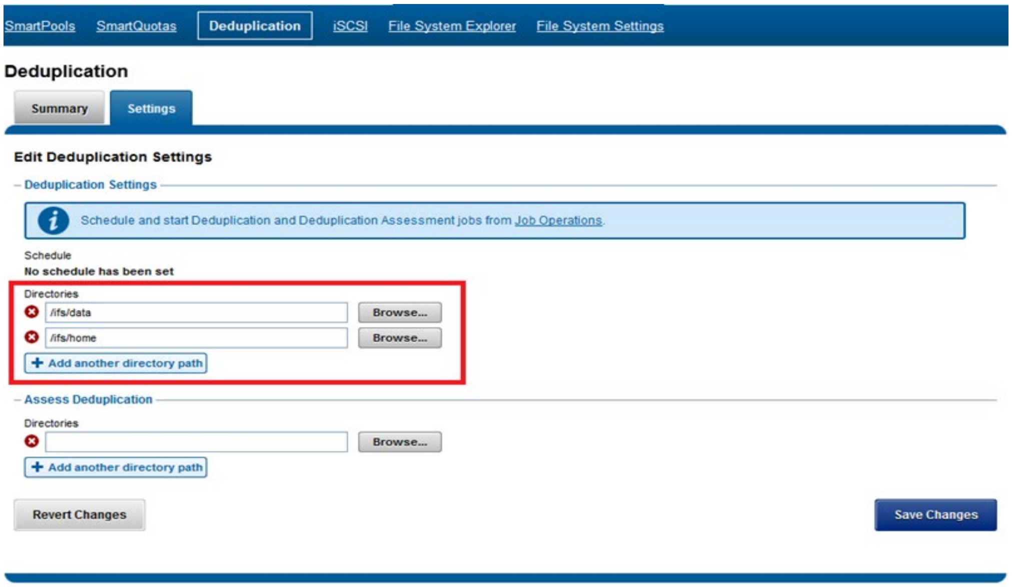
This phase populates the index table with the sampled and matching block information gathered during the previous three phases. After a file has been scanned by the dedupe job, OneFS may not find any matching blocks in other files on the cluster. Once a number of other files have been scanned, if a file continues to not share any blocks with other files on the cluster, OneFS will remove 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 data set once, after which the file is marked, preventing subsequent dedupe jobs from rescanning the file until it has been modified.

SmartDedupe Management

There are two principal elements to managing deduplication in OneFS. The first is the configuration of the SmartDedupe process itself. The second involves the scheduling and execution of the Dedupe job. These are both described below.

Configuring SmartDedupe

SmartDedupe works on data sets which 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.



Schedule and start Deduplication and Deduplication Assessment jobs from Job Operations.

Schedule
No schedule has been set

Directories
/ifs/data
/ifs/home

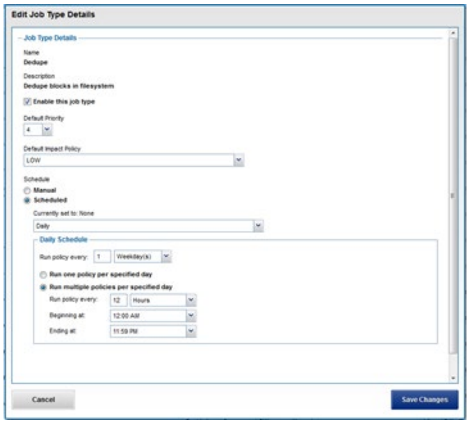
Assess Deduplication
Directories

Figure 5: SmartDedupe Configuration via 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 run a deduplication job. However, in order to configure and modify dedupe configuration settings, they must have the deduplication role privileges.

Running SmartDedupe

SmartDedupe can be run either on-demand (started manually) or via a predefined schedule. This is configured via the cluster management 'Job Operations' section of the WebUI.



Edit Job Type Details

Job Type Details

Name
Dedupe

Description
Dedupe blocks in filesystem

☑ Enable this job type

Default Priority: 4

Default Impact Policy: LOW

Schedule
○ Manual
● Scheduled

Currently set to: None

Daily

Daily Schedule

Run policy every: 1 Weekday(s)

○ Run one policy per specified day
● Run multiple policies per specified day

Run policy every: 12 Hours
Beginning at: 12:00 AM
Ending at: 11:59 PM

Figure 6: SmartDedupe Job Configuration and Scheduling via the OneFS WebUI

Note: In OneFS 7.2.1 and prior, the dedupe job will always run at a low impact level. OneFS 8.0 and beyond allows the dedupe job's impact level and priority to be reconfigured from the default

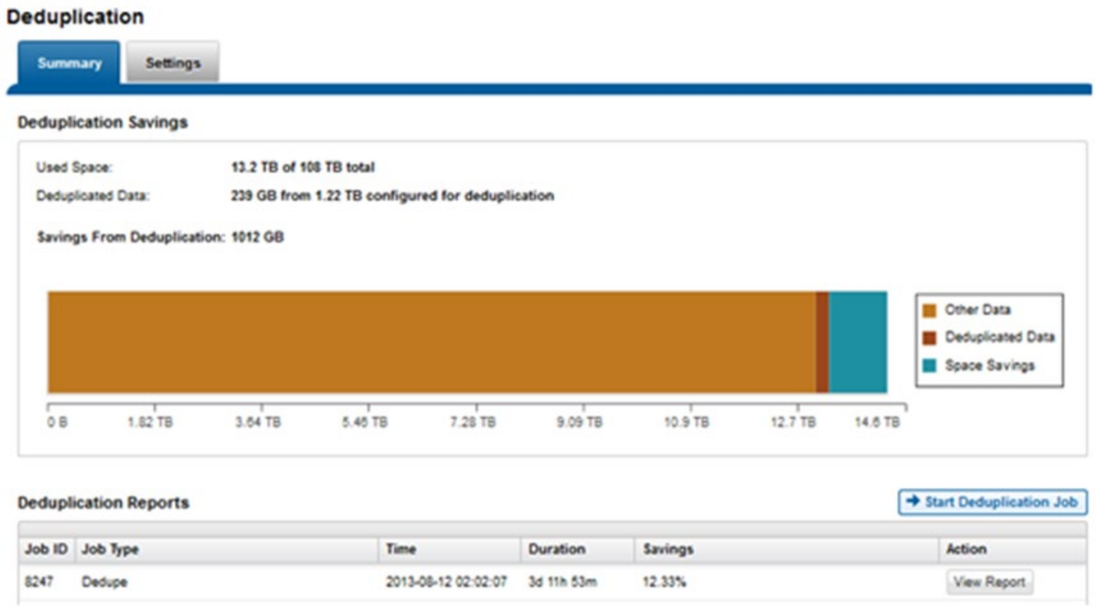
Isilon recommends scheduling and running deduplication during off-hours, when the rate of data change on the cluster is low. If clients are continually writing to files, the amount of space saved by deduplication will be minimal because the deduplicated blocks are constantly being removed from the shadow 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 navigating to **File System Management** > **Deduplication** > **Summary**.



Deduplication Savings

Used Space: 13.2 TB of 108 TB total

Deduplicated Data: 239 GB from 1.22 TB configured for deduplication

Savings From Deduplication: 1012 GB

Deduplication Reports

Job ID	Job Type	Time	Duration	Savings	Action
8247	Dedupe	2013-08-12 02:02:07	3d 11h 53m	12.33%	View Report

Figure 7: SmartDedupe Cluster Capacity Savings WebUI Chart

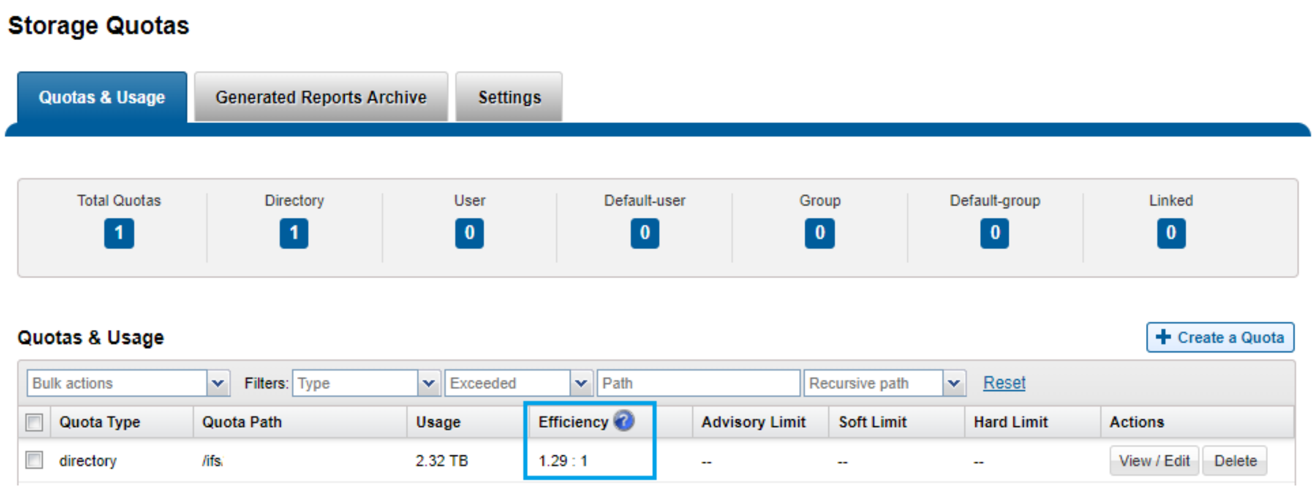
In addition to the bar chart and accompanying statistics (above), which graphically represents the data set and space efficiency in actual capacity terms, the dedupe job report overview field also displays the SmartDedupe savings as a percentage.

SmartDedupe space efficiency metrics are also provided via the ‘`isi dedupe stats`’ CLI command:

```
# isi dedupe stats
      Cluster Physical Size: 676.8841T
          Cluster Used Size: 236.3181T
  Logical Size Deduplicated: 29.2562T
             Logical Saving: 25.5125T
Estimated Size Deduplicated: 42.5774T
  Estimated Physical Saving: 37.1290T
```

Figure 8: SmartDedupe Efficiency Statistics via the CLI

In OneFS 8.2.1 and later, Isilon 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 the desired data set 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 navigating to ‘File System > SmartQuotas > Quotas and Usage’.



Storage Quotas

Total Quotas	Directory	User	Default-user	Group	Default-group	Linked
1	1	0	0	0	0	0

Quotas & Usage

□	Quota Type	Quota Path	Usage	Efficiency	Advisory Limit	Soft Limit	Hard Limit	Actions
□	directory	/ifs	2.32 TB	1.29 : 1	--	--	--	View / Edit Delete

Figure 9: OneFS WebUI SmartQuotas Quotas and Usage Status Detailing Efficiency Ratio.

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

```
# isi quota quotas list
Type      AppliesTo  Path           Snap  Hard  Soft  Adv  Used    Efficiency
-----------------------------------------------------------------------------
directory DEFAULT    /ifs           No    -     -     -    2.3247T 1.29 : 1
-----------------------------------------------------------------------------
Total: 1
```

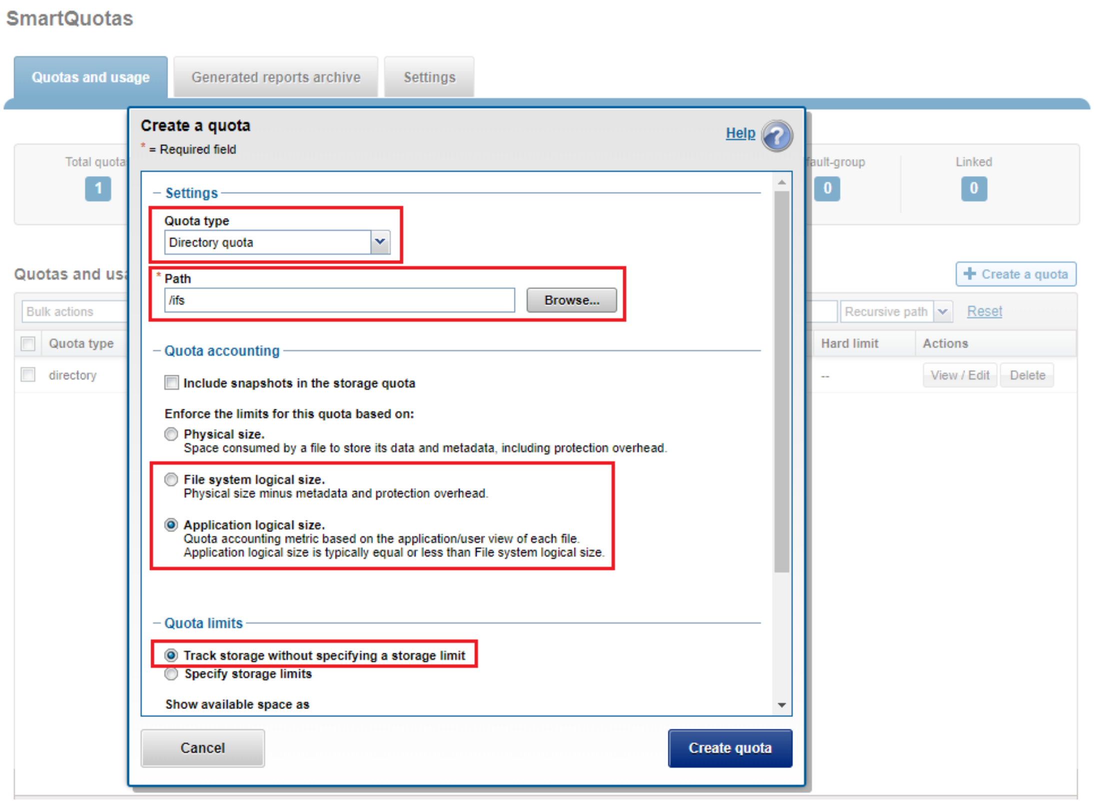
More detail, including both the physical (raw) and logical (effective) data capacities, is also available via the 'isi quota quotas view <path> <type>' CLI command. For example:

```
# isi quota quotas view /ifs directory
                        Path: /ifs
                        Type: directory
                   Snapshots: No
 Thresholds Include Overhead: No
                       Usage
                           Files: 4245818
         Physical(With Overhead): 1.80T
           Logical(W/O Overhead): 2.33T
Efficiency(Logical/Physical): 1.29 : 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:

Navigate to 'File System > SmartQuotas > Quotas and Usage' and select 'Create a Quota'. In the create pane, field, set the Quota type to 'Directory quota', add the preferred top-level path to report on, select 'File system logical size' for Quota Accounting, and set the Quota Limits to 'Track storage without specifying a storage limit'. Finally, select the 'Create Quota' button to confirm the configuration and activate the new directory quota.



Create a quota

* = Required field

Settings

Quota type: Directory quota

* Path: /ifs

Quota accounting

□ Include snapshots in the storage quota

Enforce the limits for this quota based on:

○ Physical size.
Space consumed by a file to store its data and metadata, including protection overhead.

○ File system logical size.
Physical size minus metadata and protection overhead.

● Application logical size.
Quota accounting metric based on the application/user view of each file.
Application logical size is typically equal or less than File system logical size.

Quota limits

● Track storage without specifying a storage limit

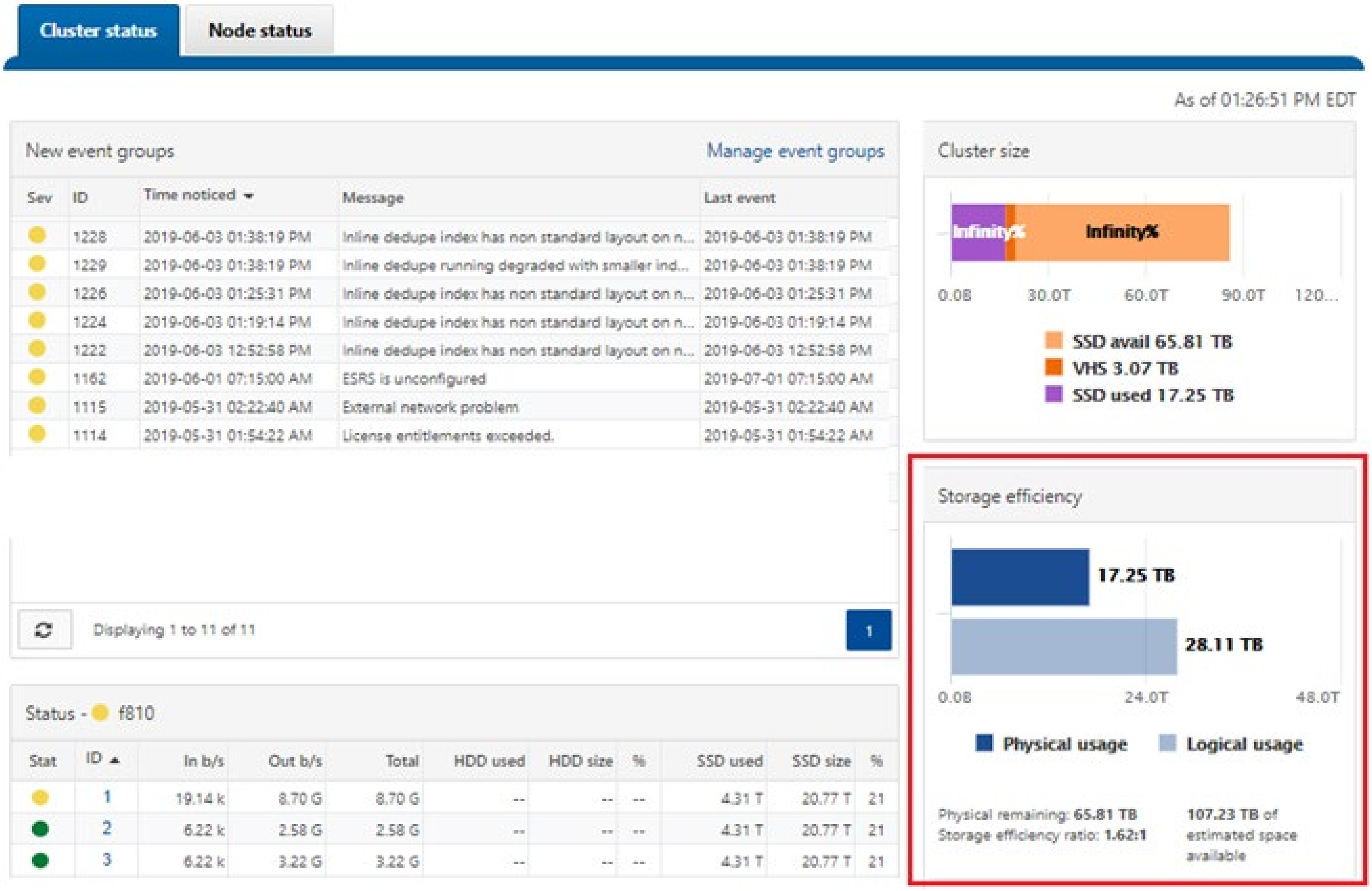
○ Specify storage limits

Show available space as

Figure 10: 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 in-line data reduction. This is in contrast to a history of stats over time, as reported in the ‘isi statistics data-reduction’ CLI command output, described above. As such, the efficiency ratio for the entire quota directory will reflect what is actually there. via the platform API as of OneFS 8.2.2.

The OneFS WebUI cluster dashboard also now displays a storage efficiency tile, which shows physical and logical space utilization histograms and reports the capacity saving from in-line 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 navigating to ‘File System > Dashboard > Cluster Overview’.



As of 01:26:51 PM EDT

New event groups

Manage event groups

Sev	ID	Time noticed	Message	Last event
●	1228	2019-06-03 01:38:19 PM	Inline dedupe index has non standard layout on n...	2019-06-03 01:38:19 PM
●	1229	2019-06-03 01:38:19 PM	Inline dedupe running degraded with smaller ind...	2019-06-03 01:38:19 PM
●	1226	2019-06-03 01:25:31 PM	Inline dedupe index has non standard layout on n...	2019-06-03 01:25:31 PM
●	1224	2019-06-03 01:19:14 PM	Inline dedupe index has non standard layout on n...	2019-06-03 01:19:14 PM
●	1222	2019-06-03 12:52:58 PM	Inline dedupe index has non standard layout on n...	2019-06-03 12:52:58 PM
●	1162	2019-06-01 07:15:00 AM	ESRS is unconfigured	2019-07-01 07:15:00 AM
●	1115	2019-05-31 02:22:40 AM	External network problem	2019-05-31 02:22:40 AM
●	1114	2019-05-31 01:54:22 AM	License entitlements exceeded.	2019-05-31 01:54:22 AM

Displaying 1 to 11 of 11

Status - ● f810

Stat	ID	In b/s	Out b/s	Total	HDD used	HDD size	%	SSD used	SSD size	%
●	1	19.14 k	8.70 G	8.70 G	--	--	--	4.31 T	20.77 T	21
●	2	6.22 k	2.58 G	2.58 G	--	--	--	4.31 T	20.77 T	21
●	3	6.22 k	3.22 G	3.22 G	--	--	--	4.31 T	20.77 T	21

Physical remaining: **65.81 TB**
Storage efficiency ratio: **1.62:1**

107.23 TB of estimated space available

Figure 11: OneFS WebUI Cluster Status Dashboard – Storage Efficiency Summary.

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 via 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.

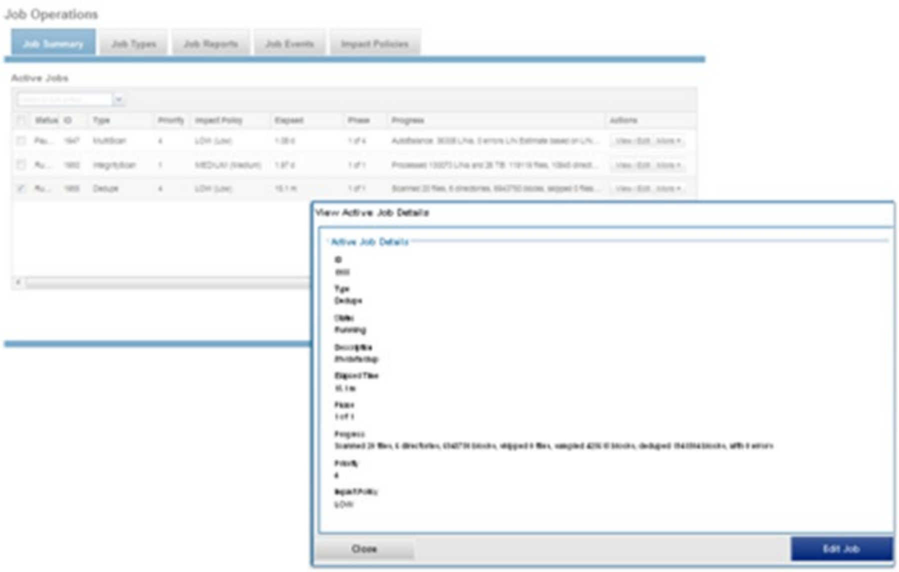
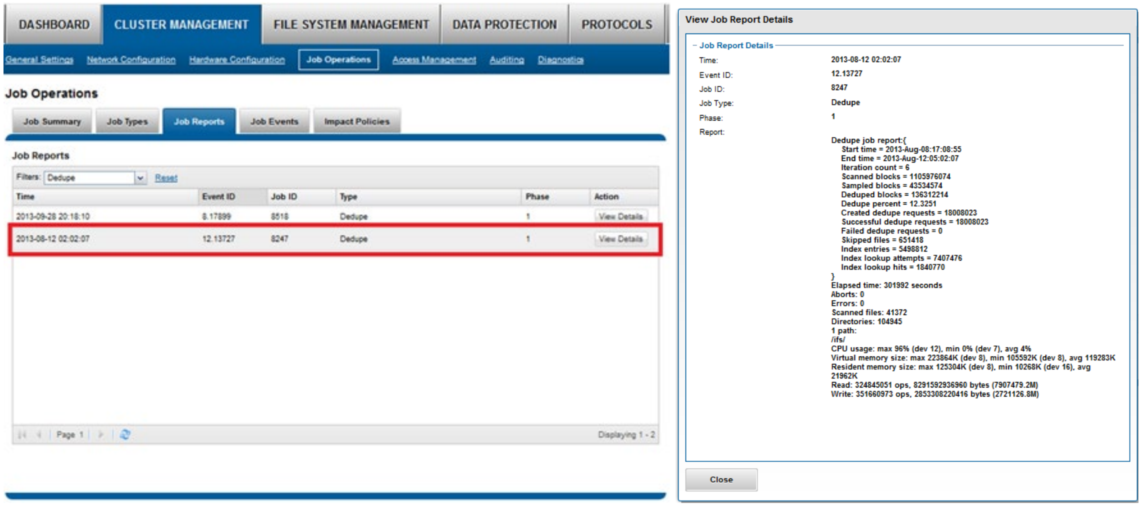


Figure 12: Example Active Job Status Update

SmartDedupe Job Reports

Once the SmartDedupe job has run to completion, or has been terminated, a full dedupe job report is available. This can be accessed from the WebUI by navigating to **Cluster Management > Job Operations > Job Reports** and selecting 'View Details' action button on the desired Dedupe job line item.



Job Operations

Job Reports

Filters: Dedupe

Time	Event ID	Job ID	Type	Phase	Action
2013-09-28 20:18:10	8.17899	8518	Dedupe	1	View Details
2013-08-12 02:02:07	12.13727	8247	Dedupe	1	View Details

View Job Report Details

Job Report Details

Time: 2013-08-12 02:02:07
Event ID: 12.13727
Job ID: 8247
Job Type: Dedupe
Phase: 1
Report:

```
Dedupe job report:{
   Start time = 2013-Aug-08:17:08:55
   End time = 2013-Aug-12:05:02:07
   Iteration count = 6
   Scanned blocks = 1105976074
   Sampled blocks = 43534574
   Deduped blocks = 136312214
   Dedupe percent = 12.3251
   Created dedupe requests = 18008023
   Successful dedupe requests = 18008023
   Failed dedupe requests = 0
   Skipped files = 651418
   Index entries = 5498812
   Index lookup attempts = 7407476
   Index lookup hits = 1840770
}
Elapsed time: 301992 seconds
Aborts: 0
Errors: 0
Scanned files: 41372
Directories: 104945
1 path:
/ifs/
CPU usage: max 96% (dev 12), min 0% (dev 7), avg 4%
Virtual memory size: max 223864K (dev 8), min 105592K (dev 8), avg 119283K
Resident memory size: max 125304K (dev 8), min 10268K (dev 16), avg
21962K
Read: 324845051 ops, 8291592936960 bytes (7907479.2M)
Write: 351660973 ops, 2853308220416 bytes (2721126.8M)
```

Figure 13: Example WebUI Dedupe Job Report

The job report contains the following relevant dedupe metrics.

Report Field	Description of Metric
Start time	When the dedupe job started.
End time	When the dedupe job finished.
Scanned blocks	Total number of blocks scanned under configured path(s).
Sampled blocks	Number of blocks that OneFS created index entries for.
Created dedupe requests	Total number of dedupe requests created. A dedupe 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 dedupe requests that completed successfully.
Failed dedupe requests	Number of dedupe requests that failed. If a dedupe request fails, it does not mean that the also job 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 hasn't been modified since. Another reason for a file to be skipped is if it's less than 32KB in size. Such files are considered too small and don't 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 hasn't 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.

Table 1: Dedupe Job Report Statistics Table

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

Note: From an execution and reporting stance, the Job Engine considers the 'dedupe' job to comprise of a single process or phase. The Job Engine events list will report that Dedupe Phase1 has ended and succeeded. This indicates that an entire SmartDedupe job, including all four internal dedupe phases (sampling, duplicate detection, block sharing, & index update), has successfully completed. For example:

```
# isi job events list --job-type dedupe

Time                Message

-------------------------------------------------------

2019-02-01T13:39:32 Dedupe[1955] Running
```

```
2019-02-01T13:39:32 Dedupe[1955] Phase 1: begin dedupe

2019-02-01T14:20:32 Dedupe[1955] Phase 1: end dedupe

2019-02-01T14:20:32 Dedupe[1955] Phase 1: end dedupe

2019-02-01T14:20:32 Dedupe[1955] Succeeded
```

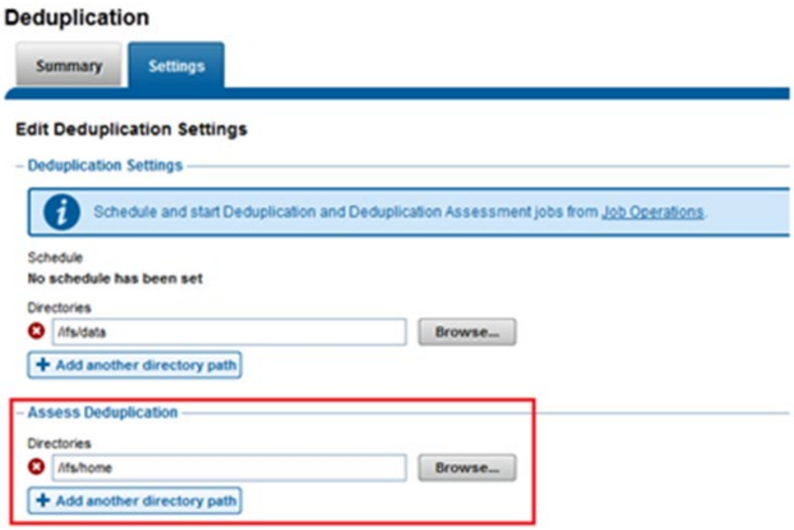
Figure 14: Example Command Line (CLI) Dedupe Job Events List

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

Space Savings Estimation with the SmartDedupe Assessment Job

To complement the actual Dedupe job, a dry-run Dedupe Assessment job is also provided to help estimate the amount of space savings that will be seen by running deduplication on a particular directory or set of directories. The dedupe assessment job reports a total potential space savings. The dedupe assessment does not differentiate the case of a fresh run from the case where a previous dedupe 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. Isilon recommends that the user should run the assessment job once on a specific directory prior to starting an actual dedupe job on that directory.

The assessment job runs similarly to the actual dedupe job but uses a separate configuration. It also does not require a product license and can be run prior to purchasing SmartDedupe in order to determine whether deduplication is appropriate for a particular data set or environment.



Deduplication

Edit Deduplication Settings

Deduplication Settings

Schedule and start Deduplication and Deduplication Assessment jobs from Job Operations.

Schedule
No schedule has been set

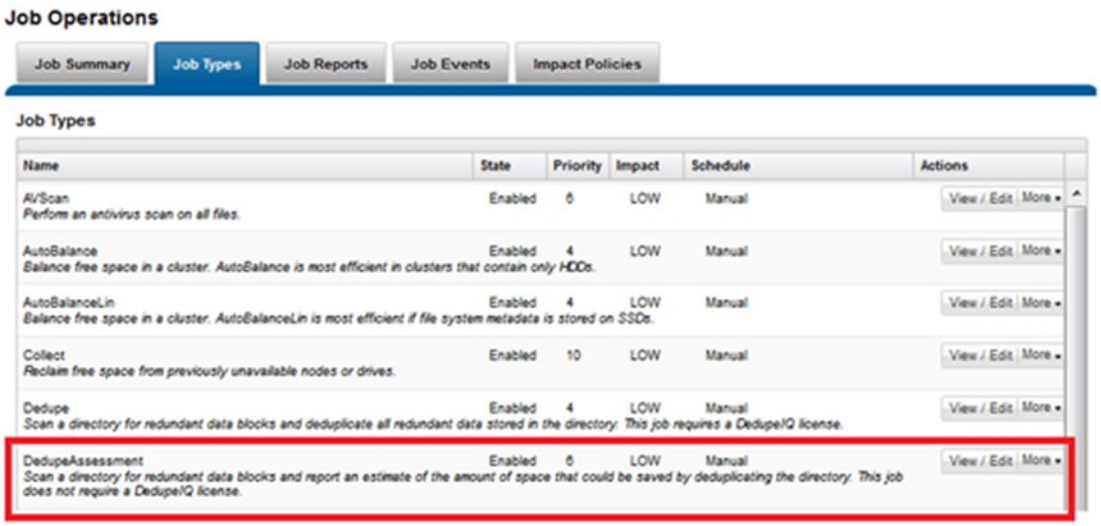
Directories
/ifs/data

Assess Deduplication

Directories
/ifs/home

Figure 15: Deduplication Assessment Job Configuration

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



Name	State	Priority	Impact	Schedule	Actions
AVScan *Perform an antivirus scan on all files.*	Enabled	6	LOW	Manual	View / Edit More ▾
AutoBalance *Balance free space in a cluster. AutoBalance is most efficient in clusters that contain only HDDs.*	Enabled	4	LOW	Manual	View / Edit More ▾
AutoBalanceLin *Balance free space in a cluster. AutoBalanceLin is most efficient if file system metadata is stored on SSDs.*	Enabled	4	LOW	Manual	View / Edit More ▾
Collect *Reclaim free space from previously unavailable nodes or drives.*	Enabled	10	LOW	Manual	View / Edit More ▾
Dedupe *Scan a directory for redundant data blocks and deduplicate all redundant data stored in the directory. This job requires a DedupeIQ license.*	Enabled	4	LOW	Manual	View / Edit More ▾
DedupeAssessment *Scan a directory for redundant data blocks and report an estimate of the amount of space that could be saved by deduplicating the directory. This job does not require a DedupeIQ license.*	Enabled	6	LOW	Manual	View / Edit More ▾

Figure 16: Dedupe Assessment Job Control via the OneFS WebUI

Performance with SmartDedupe

As with most things in life, deduplication is a compromise. In order to gain increased levels of storage efficiency, additional cluster resources (CPU, memory and disk IO) are utilized to find and execute the sharing of common data blocks.

Another important performance impact consideration with dedupe is the potential for data fragmentation. After deduplication, files that previously enjoyed contiguous on-disk layout will 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 will not share blocks across node pools or data tiers and will not attempt to deduplicate files smaller than 32KB in size. On the other end of the spectrum, the largest contiguous region that will be matched is 4MB.

Because deduplication is a data efficiency product rather than performance enhancing tool, in most cases the consideration will be around cluster impact management. This is from both the client data access performance front, since, by design, multiple files will be sharing common data blocks, and also from the dedupe job execution perspective, as additional cluster resources are consumed to detect and share commonality.

The first deduplication job run will often take a substantial amount of time to run, since 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 will typically improve 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 data set with a high rate of change. If a deduplication job is paused or interrupted, it will automatically resume 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 1TB of data per day, per node.

SmartDedupe Licensing

SmartDedupe is included as a core component of Isilon OneFS but requires a valid product license key in order to activate. This license key can be purchased through your Isilon account team. An unlicensed cluster will show a SmartDedupe warning until a valid product license has been purchased and applied to the cluster.

License keys can be easily added via the ‘Activate License’ section of the OneFS WebUI, accessed by navigating via Cluster Management > Licensing.

Note: The SmartDedupe dry-run estimation job can be run without any licensing requirements, allowing an assessment of the potential space savings that a dataset might yield before making the decision 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 mentioned above, the deduplication assessment job can be run to help predict the likely space savings that deduplication would provide on a given data set.

Virtual machines files often contain duplicate data, much of which is rarely modified. Deduplicating similar OS type virtual machine images (for example VMware VMDK files, etc, 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 can also deduplicate iSCSI LUNs, since deduplication does not treat a LUN (a directory containing a group of extent files) differently from regular files.

Isilon 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 won't 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 won't 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.

Below are some examples of typical space reclamation levels that have been achieved with SmartDedupe.

Note: These dedupe space savings values are provided solely as rough guidance. Since no two data sets are alike (unless they're replicated), actual results can vary considerably from these examples.

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%

Figure 17: Typical Workload Space Savings with SmartDedupe

SmartDedupe Best Practices

For optimal cluster performance, Isilon recommends observing the following SmartDedupe best practices. Please note that some of this information may be covered elsewhere in this paper.

- Deduplication is most effective when applied to data sets with a low rate of change – for example, archived data.
- Enable SmartDedupe to run at subdirectory level(s) below /ifs.
- Avoid adding more than ten subdirectory paths to the SmartDedupe configuration policy,
- SmartDedupe is ideal for home directories, departmental file shares and warm and cold archive data sets.
- Run SmartDedupe against a smaller sample data set first to evaluate performance impact versus space efficiency.
- Schedule deduplication to run during the cluster's low usage hours – i.e. overnight, weekends, etc.

- After the initial dedupe job has completed, schedule incremental dedupe 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 dedupe 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, it is important to verify that the logical data size (i.e. 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 data set prior to enabling a snapshots schedule.
- Where possible, perform any snapshot restores (reverts) before running a deduplication job. And run a dedupe job directly after restoring a prior snapshot version.

SmartDedupe Considerations

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

- Since 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 will not permit block sharing across different hardware types or node pools to reduce the risk of performance asymmetry.
- SmartDedupe will not share blocks across files with different protection policies applied.
- OneFS metadata, including the deduplication index, is not deduplicated.
- OneFS will not deduplicate redundant information within a file, only across different files.
- Deduplication is a long running process that involves multiple job phases that are run iteratively.
- SmartDedupe will not attempt to deduplicate files smaller than 32KB in size.
- Dedupe job performance will typically improve significantly on the second and subsequent job runs, once the initial index and the bulk of the shadow stores have already been created.
- SmartDedupe will not deduplicate the data stored in a snapshot. However, snapshots can certainly be created of deduplicated data.
- If deduplication is enabled on a cluster that already has a significant amount of data stored in snapshots, it will take time before the snapshot data is affected by deduplication. Newly created snapshots will contain deduplicated data, but older snapshots will not.
- SmartDedupe will automatically run with a “low-impact” Job Engine policy. In OneFS 7.x releases, this cannot be manually reconfigured. However, this restriction is lifted in OneFS 8.0 and beyond.
- From OneFS 8.0 onwards, SmartDedupe deduplicates common blocks within the same file, resulting in even better data efficiency.

SmartDedupe and OneFS Feature Integration

SyncIQ Replication and SmartDedupe

When deduplicated files are replicated to another Isilon cluster via SyncIQ, or backed up to a tape device, the deduplicated files are inflated (or rehydrated) back to their original size, since they no longer share blocks on the target Isilon 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. Because of this, 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, it is important to 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 Isilon cluster, you can configure deduplication for the target directories of your replication policies. Although this will deduplicate data on the target directory, it will not allow SyncIQ to transfer shadow stores. Deduplication is still performed post-replication, via 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 won’t be deduplicated unless deduplication is provided by the backup vendor’s DMA software. However, compression is often provided natively by the backup tape or VTL device.

Snapshots and SmartDedupe

SmartDedupe will not deduplicate the data stored in a snapshot. However, snapshots can be created of deduplicated data. If a snapshot is taken of a deduplicated directory, and then the contents of that directory are modified, the shadow stores will be transferred to the snapshot over time. Because of this, more space will be saved on a cluster if deduplication is run prior to enabling snapshots.

If deduplication is enabled on a cluster that already has a significant amount of data stored in snapshots, it will take time before the snapshot data is affected by deduplication. 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 back to normal files if they are overwritten by a snapshot revert. However, once the snapshot revert is complete, deduplication can be run on the directory again and the resulting space savings will persist on the cluster.

SmartLock and SmartDedupe

SmartDedupe is also fully compatible with Dell EMC Isilon SmartLock, Isilon’s data retention and compliance product. Dedupe delivers storage efficiency for immutable archives and write once, read many (or WORM) protected data sets.

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 additional space used by the shadow store will not be accounted for by the quota.

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 deduped file that is moved by SmartPools to a different pool or tier will retain the shadow references to the shadow store on the original pool. This breaks the rule for deduping across different disk pool policies, but it is less impactful to do this than rehydrate files that are moved. Further dedupe activity on that file will no longer be allowed to reference any blocks in the original shadow store. The file will need to be deduped against other files in the same disk pool policy. If the file had not yet been deduped, the dedupe index may have knowledge about the file and will still think 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 deduped, the dedupe index will have knowledge of the shadow store only. Since 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), then a similar situation occurs and the SmartDedupe job will discover this and purge knowledge of the shadow store from the dedupe index.

In-line Compression and SmartDedupe

SmartDedupe post process dedupe is compatible with in-line compression , currently available on both the F810 all-flash and H5600 hybrid platforms, and vice versa. In-line compression is able to compress OneFS shadow stores. However, in order for SmartDedupe to process compressed data, the SmartDedupe job will have to decompress it first in order to perform deduplication, which is an addition resource overhead.

In-line Deduplication and SmartDedupe

While Isilon has offered a native file system deduplication solution for several years, until OneFS 8.2.1 this was always accomplished by scanning the data after it has been written to disk, or post-process. With in-line data reduction, deduplication is now 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.

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

When a match is found, in-line dedupe performs a byte-by-byte comparison of each block to be shared to avoid the potential for a hash collision. Data is prefetched prior 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 then shared by writing the matching data to a common shadow store and creating references from the original files to this shadow store.

In-line dedupe 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, in-line dedupe will merge the results into regions and process them efficiently.

In-line dedupe is also detects dedupe 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, in-line dedupe 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 dedupe worker thread. As such, it is possible to deduplicate a data set 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.

InsightIQ and SmartDedupe

InsightIQ, Isilon's 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 mentioned above, an enterprise's data typically contains substantial quantities of redundant information. And home directories, file shares and data archives are great example of workloads that consistently yield solid deduplication results. Each time a spreadsheet, document or email attachment is saved by multiple employees, the same file is stored in full multiple times, taking up valuable disk capacity. Isilon 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 data sets.

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, etc.

These are all excellent use cases for deduplication, since 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 OneFS SmartLock-protected immutable archives and other WORM data sets, 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 which are frequently scanned or indexed, metadata read acceleration on SSDs

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 dedupe job has successfully completed, subsequent incremental dedupe 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 enable Isilon 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% range or higher. This is in contrast to the industry mean of around 50-60% 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

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

Isilon SmartDedupe's integration with the industry's leading 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, Isilon SmartDedupe is easy to estimate, configure and manage, and provides enterprise data efficiency within a single, 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 EMC sales representative or authorized reseller to learn more about how Isilon scale-out NAS storage solutions can benefit your organization.

Shop Dell EMC Isilon to compare features and get more information.

Learn more about Dell EMC Isilon solutions

Contact a Dell EMC Expert

View more resources

Join the conversation with #DellEMCStorage