Technical Report

NetApp Data Compression, Deduplication and Data Compaction

Data ONTAP 8.3.1 and Later

Karthik Viswanath, NetApp
July 2016 | TR-4476

Abstract

This technical report focuses on implementing NetApp® data compression, NetApp deduplication, and NetApp data compaction running on NetApp Data ONTAP® software version 8.3.1 and later.

Version History

Version	Date	Document Version History
Version 1.0	November 2015	Karthik Viswanath: initial version
Version 2.0	July 2016	Karthik Viswanath : updated to include 4:1 storage efficiency guarantee and data compaction details

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Overview

One of the biggest challenges for companies today is the cost of storage. Storage represents the largest and fastest growing IT expense. The NetApp portfolio of storage efficiency technologies is aimed at lowering this cost. NetApp deduplication and data compression are two key components of NetApp storage efficiency technologies that enable users to store the maximum amount of data for the lowest possible cost.

This paper mainly focuses on two NetApp features: NetApp deduplication and NetApp data compression. These technologies can work together or independently to achieve optimal savings. Both have the ability to run either as an inline process as data is written to disk or as a scheduled process. When the two are enabled on the same volume, the data is first compressed and then deduplicated. Deduplication removes duplicate compressed or uncompressed blocks in a data volume. Although compression and deduplication work well together, the savings are not necessarily the sum of the savings when each is run individually on a dataset.

[July 2016] This paper has now been updated to include NetApp data compaction, a new storage efficiency technology, and 4:1 storage efficiency guarantee program, both of which are available with ONTAP® 9.

As the title implies, this technical report covers Data ONTAP 8.3.1 and later. For information about implementation with earlier versions of Data ONTAP, refer to:

- TR-3505: NetApp Deduplication for FAS and V-Series Deployment and Implementation Guide
- TR-3505i: NetApp Deduplication and Data Compression Deployment and Implementation Guide
- TR-3958: NetApp Data Compression and Deduplication Deployment and Implementation Guide, Data ONTAP 8.1 Operating in 7-Mode
- TR-3966: NetApp Data Compression and Deduplication Deployment and Implementation Guide

The same information applies to both FAS, V-Series, and NetApp FlexArray® systems, unless otherwise noted.

Whenever references are made to volumes, the references apply to both NetApp FlexVol® volumes and Infinite Volumes.

For more information about Infinite Volumes, see TR-4178: NetApp Infinite Volume Deployment and Implementation Guide.

2 NetApp 4:1 Storage Efficiency Guarantee

Starting with ONTAP 9, NetApp guarantees that the use of NetApp's storage efficiency technologies on All Flash FAS (AFF) systems reduces the total logical used capacity needed to store customer data by 75% (that is, yield a data reduction ratio of 4:1).

The guarantee is only available for All Flash FAS systems running ONTAP 9. To qualify for the guarantee, the All Flash FAS systems should be configured with NetApp best practices.

To participate and to request additional information, refer to the FlashAdvantage 3-4-5 program or contact your NetApp sales or partner sales representative.

2.1 Storage Efficiency Ratio Visualization

As part of the storage efficiency guarantee program, to validate the efficiency guarantee, the following storage efficiency ratio visualizations are available for All Flash FAS systems running ONTAP 9.

ONTAP CLI

NetApp has introduced a new command, `` `storage aggregate show-efficiency` ``, that reports aggregate-level efficiency ratios along with the details of logical space used and physical space used. The following efficiency ratios are displayed as part of the `` `storage aggregate show-efficiency` `` command:

- Cumulative storage efficiency ratio
- Aggregate storage efficiency ratio/data compaction storage efficiency ratio
- Volume storage efficiency ratio/data reduction storage efficiency ratio (efficiency from deduplication, compression and zero-block deduplication)
- Snapshot® and FlexClone® storage efficiency ratio

Figure 1) The `storage aggregate show-efficiency` CLI command.

```
hostname::*> storage aggregate show-efficiency

Aggregate: <aggregate-name>
    Node: <node-name>

----- Cumulative Storage Efficiency ------
    Logical    Physical            Storage
       Used        Used   Efficiency Ratio
----------- ----------- ------------------
     6.45TB      1.27TB             5.09:1

--- Aggregate level Storage Efficiency ---
    Logical    Physical   Compaction             Storage
       Used        Used        Saved    Efficiency Ratio
----------- ----------- ------------ ------------------
     1.61TB      1.27TB      355.6GB             1.27:1

--------------------- Volume level Storage Efficiency ------------------------
    Logical    Physical       Dedup Compression         VBN            Storage
       Used        Used       Saved       Saved  Zero Saved   Efficiency Ratio
----------- ----------- ----------- ----------- ----------- ------------------
     6.26TB      1.42TB      2.86GB      4.85TB          0B             4.43:1

---------- Snapshot and FlexClone volume Storage Efficiency ----------
------ Snapshot -------    --- FlexClone volume --
    Logical    Physical       Logical    Physical            Storage
       Used        Used          Used        Used   Efficiency Ratio
----------- -----------    ----------- ----------- ------------------
     5.18TB      90.7GB            0B          0B            58.48:1
```

Note: The `` `aggregate show-efficiency` `` command is available only in advanced mode.

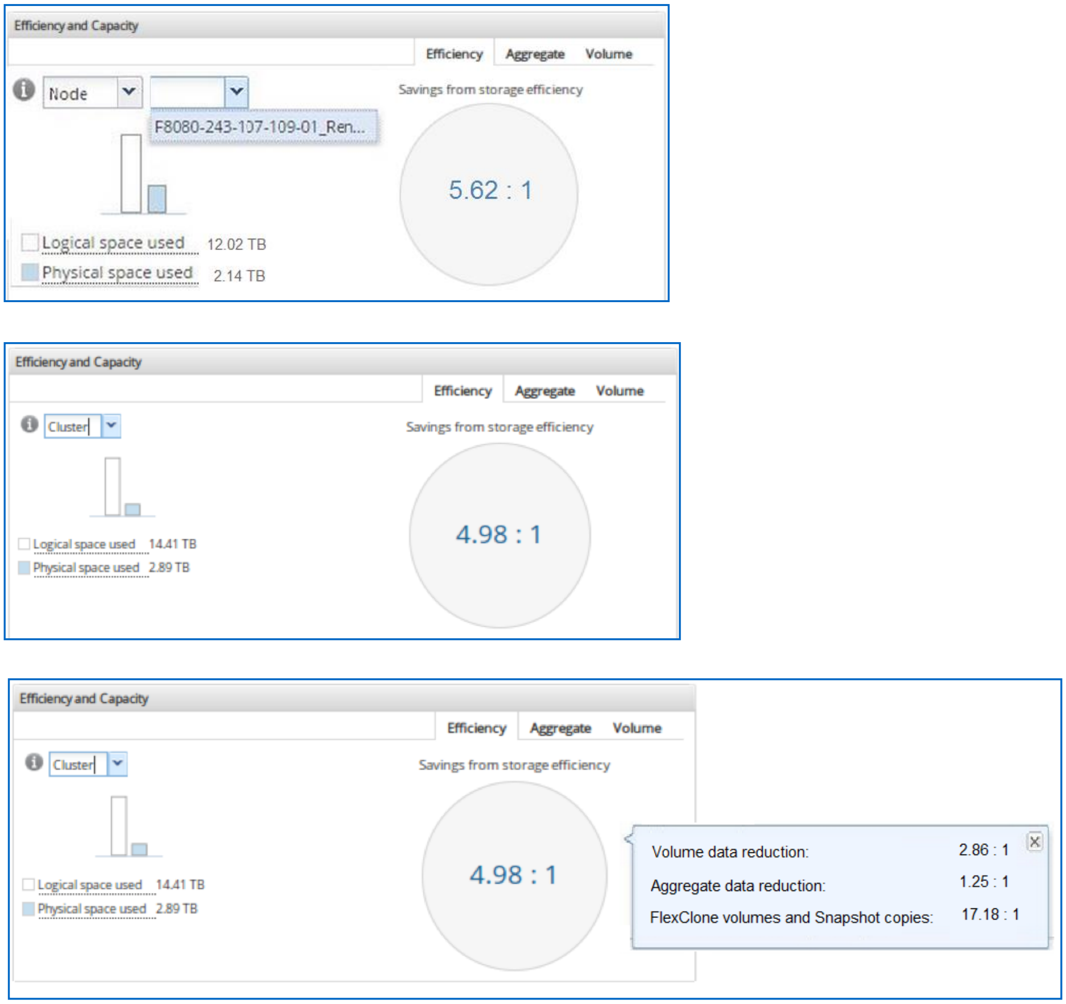
The command output is also available through a new NetApp Manageability SDK called `` `aggr-storage-efficiency-get-iter` ``.

OnCommand System Manager

On All Flash FAS systems running ONTAP 9, the System Manager dashboard shows a new tab named Efficiency under the Efficiency and Capacity panel that displays the following ratio information at cluster and node levels along with logical space used and physical space used details:

- Cumulative storage efficiency
- Aggregate storage efficiency/data compaction storage efficiency
- Volume storage efficiency/data reduction storage efficiency (deduplication, compression, and zero-block deduplication)
- Snapshot and FlexClone storage efficiency

Figure 2) OnCommand System Manager Efficiency and Capacity dashboard.

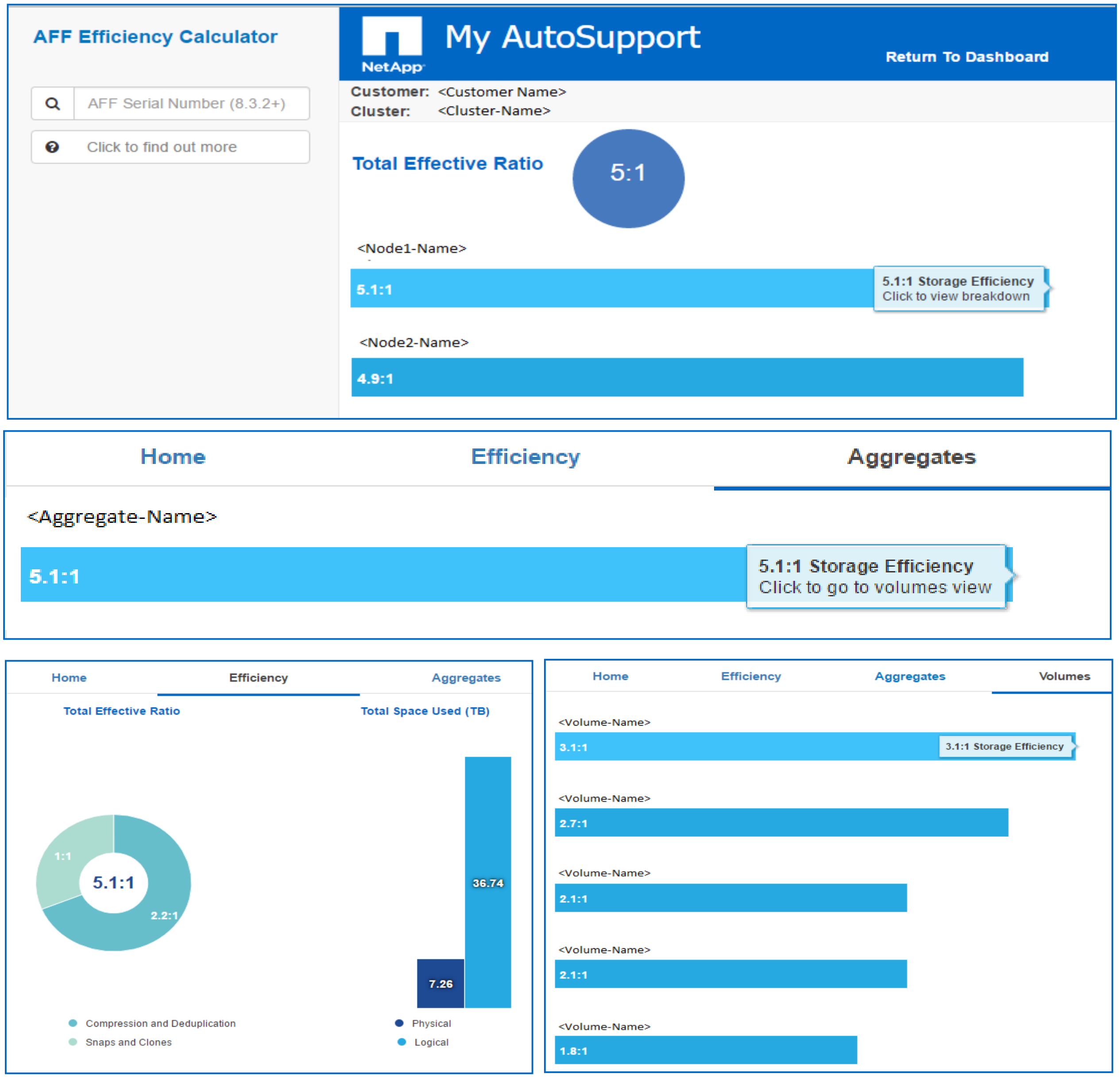


My AutoSupport

My AutoSupport® has introduced a new tool called Flash Efficiency Calculator that displays the storage efficiency ratios. It takes serial number of the controller as input and displays the following ratios:

- Cumulative storage efficiency for the cluster
- Cumulative storage efficiency for the nodes in the cluster along with:
  - Deduplication and compression node ratios
  - Snapshot and clone node ratios
  - Logical used capacity and physical used capacity for the node
  - Cumulative storage efficiency for the aggregates in the node:
    - Cumulative storage efficiency for the volumes in the aggregate

Figure 3) My AutoSupport AFF Efficiency Calculator.



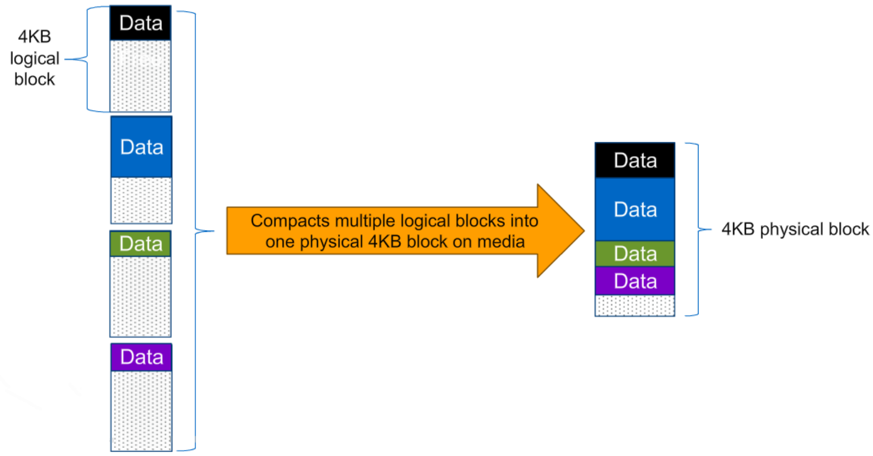
Flash Efficiency Calculator can be accessed from the My AutoSupport homepage or using this direct link.

3 NetApp Data Compaction

Starting with ONTAP 9, NetApp has introduced a new patented storage efficiency technology called data compaction that further reduces the physical used space needed to store data. It is a significant addition to our storage efficiency portfolio and complements deduplication and compression technologies.

Data compaction takes I/Os that normally consume a 4K block on physical storage and pack multiple such I/Os into one physical 4K block.

Figure 4) How data compaction works.



Data compaction is an inline operation and occurs after inline compression and inline deduplication. On an AFF system, the order of execution is as follows:

1. **Inline zero-block deduplication.** All zero blocks are detected, and no user data is written to physical storage; only metadata and reference counts are updated.
2. **Inline adaptive compression.** Compresses 8K logical blocks into 4K physical blocks; very efficient in determining compressibility of the data and doesn't waste lot of CPU cycles trying to compress incompressible data.
3. **Inline deduplication.** Opportunistically deduplicates incoming blocks to already existing blocks on physical storage.
4. **Inline adaptive data compaction.** Combines multiple <4K logical blocks into a single 4K physical block to maximize savings. It also tries to compress any 4K logical blocks that are skipped by inline compression to gain additional compression savings.

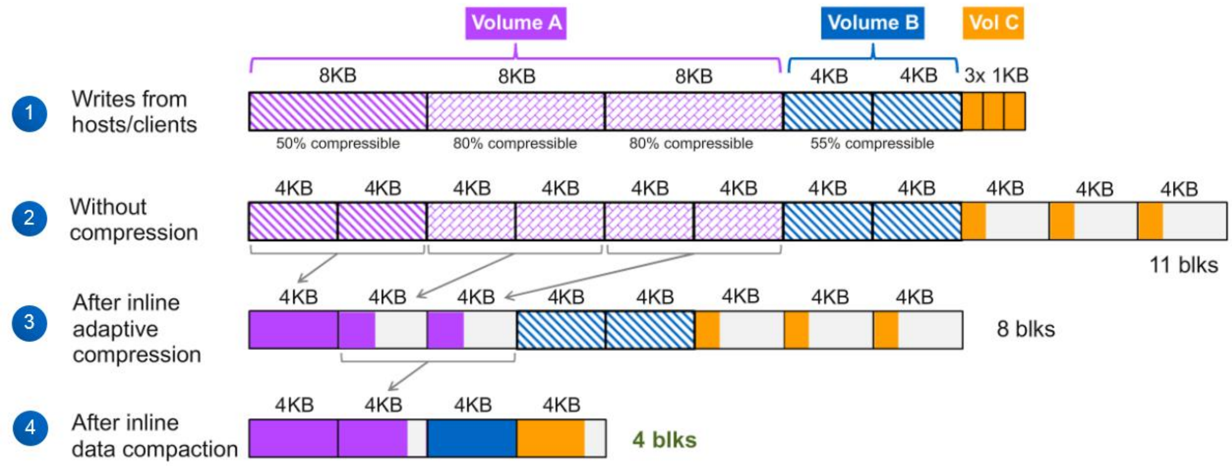
Data compaction is most beneficial for highly compressible data, very small I/Os and files (<2KB), and larger I/Os with lots of white space.

Inline data compaction is enabled by default on AFF systems and can optionally be enabled on Flash Pool™ and HDD systems. On Flash Pool systems, only writes to HDDs are compacted. Data compaction can be enabled on a per volume basis.

Data compaction can be run independently of deduplication or compression. It is always recommended to run inline adaptive data compaction along with inline adaptive compression and inline deduplication to maximize savings. Data compaction is an additive capability to inline compression as it can compress 4K blocks that are skipped by inline compression. Because inline adaptive compression skips any blocks that are 4K, the maximum compression savings that can be achieved with inline adaptive compression are 2:1. When inline adaptive data compaction is used along with inline compression, inline adaptive compression savings can be >2:1. Data compaction helps multiply compression savings.

Figure 5 shows the benefits of using inline data compaction and how inline adaptive compression savings are multiplied when used along with inline data compaction.

Figure 5) Data compaction example.



1. Figure 5 shows I/Os from three volumes: Volume A with three 8K I/Os (one of which is 50% compressible, and two are 80% compressible), Volume B with two 4K I/Os (both are 55% compressible), and Volume C with three 1K I/Os.
2. Without compression or data compaction, the incoming I/Os would have consumed a total of 11 4K blocks on physical storage. The 1K I/Os from Volume C each require 4K because the minimum block size in WAFL® is 4K.
3. If inline adaptive compression is used, the 50% compressible 8K I/O from Volume A would be compressed to 4K, and the two 80% compressible 8K I/Os from Volume A and the three 1K I/Os from Volume C would also consume 4K each on the physical storage because of the WAFL 4K block size, resulting in a total of 8 4K blocks on physical storage.
4. With inline adaptive data compaction, the two 80% compressible 8K I/Os from Volume A would be packed into a single 4K block, the two 55% compressible 4K I/Os from Volume B would be packed into another 4K block, and the three 1K I/Os from Volume C would be packed into another 4K block, resulting in a total of 4 4K blocks on physical storage.

Performance

Data compaction by itself does not have any additional performance overhead, but because data compaction also tries to compress 4K blocks that are skipped by inline compression, there could be an additional 1% to 2% CPU cycles consumed. As with adaptive compression, data compaction has built-in heuristics to efficiently detect incompressible blocks.

Space Savings

Space savings from data compaction are highly dependent on the incoming I/O, compressibility, and the existing data layout on physical storage. Even for the same dataset, it can vary from one system to another. So far, we have conducted tests on Oracle database and SQL Server 2014 to measure data compaction savings. Figures 6 and 7 summarize the results. For more information about configuration, test setup, and detailed results, refer to the following TRs:

- NetApp AFF8080A EX Storage Efficiency and Performance with Oracle Database
- NetApp AFF8080A EX Storage Efficiency and Performance with Microsoft SQL Server 2014

Figure 6) Data compaction savings: Oracle Database and ONTAP 9.

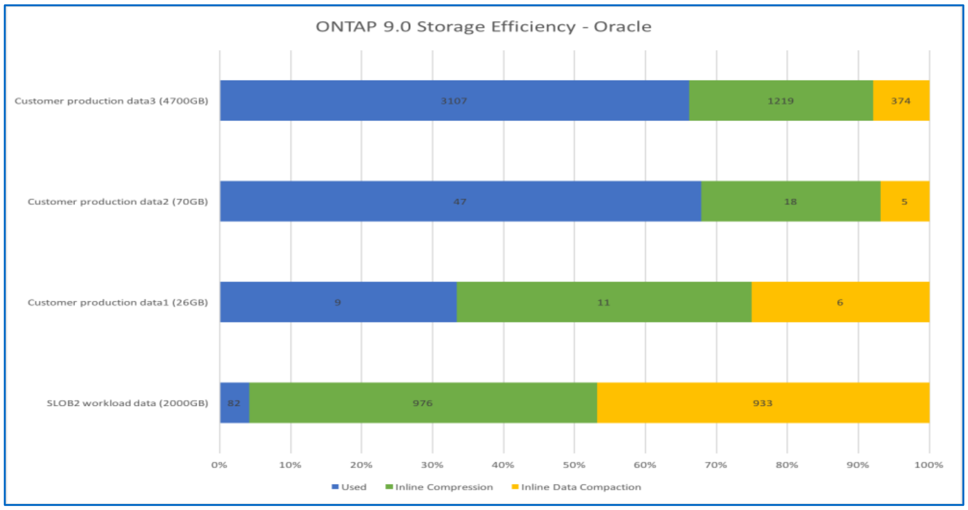
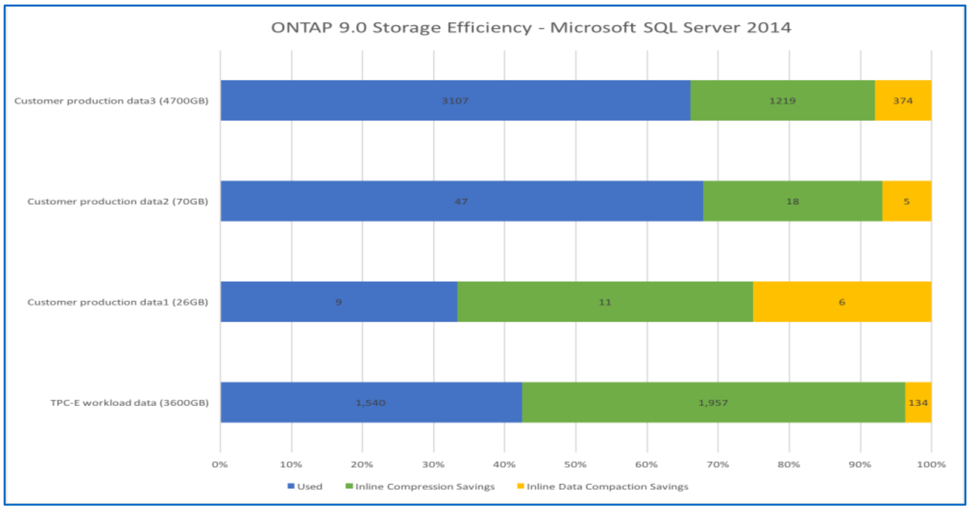


Figure 7) Data compaction savings: SQL Server 2014 and ONTAP 9.



Space Savings Estimation Tool (SSET) has been updated to include data compaction savings estimates.

Notes on Data Compaction

1. Volumes must be thin provisioned (no space guarantee).
2. It works at the aggregate (physical block) level using reference counts. It does not change logical data.
3. Blocks that are inline deduped are not compacted. Compacted blocks can be deduped by background dedupe.
4. It can reduce space consumed by Snapshot copies:
   - A CLI command to compact individual files in a Snapshot copy is available.
   - The command is only available in diagnostic mode and should be used in consultation with NetApp Support.

5. For accurate space savings reporting, aggregate space utilization should be >10%:
   - Data compaction savings are reported at an aggregate level because data compaction works at the physical block level.
   - Data compaction savings for an aggregate can be visualized using the CLI `'aggregate show-efficiency'` advanced mode command. System Manager dashboard and My AutoSupport Flash Efficiency Calculator also display aggregated data compaction savings for a node or cluster. For more details, see the "Storage Efficiency Ratio Visualization" section, earlier.

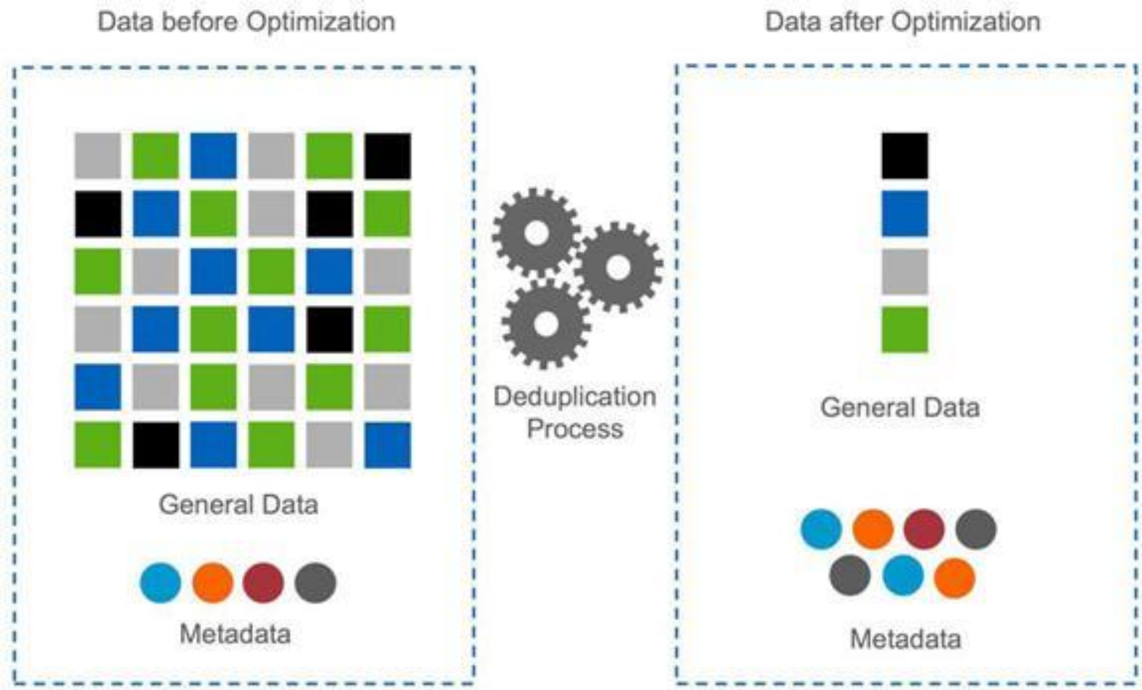
4 NetApp Deduplication

Part of NetApp's storage efficiency offerings, NetApp deduplication provides block-level deduplication within a FlexVol volume or data constituent. NetApp V-Series and FlexArray are designed for use as a gateway system that sits in front of third-party storage, allowing NetApp storage efficiency and other features to be used on third-party storage.

Essentially, deduplication removes duplicate blocks, storing only unique blocks in the FlexVol volume or data constituent. Deduplication creates a small amount of additional metadata in the process. Notable features of deduplication include:

- It works with a high degree of granularity: that is, at the 4KB block level.
- It operates on the active file system of the FlexVol volume or data constituent. Any block referenced by a NetApp Snapshot copy is not available until the Snapshot copy is deleted.
- It is a background process that can be configured to run automatically, be scheduled, or run manually through the CLI, NetApp System Manager, or NetApp OnCommand Unified Manager. Starting in Data ONTAP 8.3.2, deduplication can be run as an inline process in addition to as a background process.
- It is application transparent, and therefore it can be used to deduplicate data originating from any application that uses the NetApp system.
- It is enabled and managed by using a simple CLI or GUI such as System Manager or NetApp OnCommand Unified Manager.

Figure 8) How NetApp deduplication works at the highest level.



In summary, this is how deduplication works: Newly saved data is stored in 4KB blocks as usual by Data ONTAP. Each block of data has a digital fingerprint, which is compared to all other fingerprints in the FlexVol volume or data constituent. If two fingerprints are found to be the same, a byte-for-byte comparison is done of all bytes in the block. If there is an exact match between the new block and the existing block on the FlexVol volume or data constituent, the duplicate block is discarded, and its disk space is reclaimed.

Starting in Data ONTAP 8.3.0, NetApp added support for inline zero-block deduplication, which detects zero blocks inline and deduplicates them before they are written to disk. Inline zero-block deduplication is enabled by default on volumes having deduplication or volumes having the newly introduced inline-only efficiency policy.

In Data ONTAP 8.3.1, deduplication is primarily a postprocess operation. Starting with Data ONTAP 8.3.2, NetApp added support for inline deduplication in addition to postprocess deduplication. Users can enable inline deduplication at a volume level. The goal of inline deduplication design in Data ONTAP 8.3.2 is to provide a very low performance impact and adaptive inline dedupe that is useful in eliminating duplicate writes as they come into the system and before they are sent to disks. To that end, the inline deduplication implementation in Data ONTAP 8.3.2 removes duplicate blocks that reside in memory and in the buffer cache because of recent writes. Also, the inline deduplication implementation is opportunistic and adaptive so as not to affect client I/O performance. This implementation has the ability to throttle itself when the performance impact is beyond 1% to 2%. As a result, users might see a reduction in inline dedupe savings when the load on the system is high, but the client I/O performance is not affected.

The current implementation of inline dedupe does not eliminate the need for background dedupe. The target use cases for inline dedupe in Data ONTAP 8.3.2 are VM patch apply and copy offload. Using inline deduplication, a NetApp internal testing team achieved ~56% space savings on a VDI workload composed of a patch update of 1,000 VMs that translated to ~1TB upfront storage reduction.

Inline deduplication is supported on AFF, HDD, all SSD, and NetApp Flash Pool systems. Such deduplication is enabled by default on AFF systems. By using inline deduplication in combination with background deduplication, users can achieve maximum dedupe savings and a significant reduction in $/gigabyte and overall TCO, especially on AFF deployments.

To enable inline deduplication on volumes residing within an HDD/all-SSD (non-AFF) aggregate, the following option needs to be set on the controller – `sis.idedup_allow_non_aff_hya on.` Make sure you have done sufficient performance testing before enabling inline deduplication on HDD aggregates because spindle speed might become a bottleneck for inline deduplication performance. Contact your NetApp representative/Support if you are not sure whether to enable it on your configuration.

On certain AFF systems, there is a way to simulate inline deduplication behavior in Data ONTAP 8.3.1 as well. It is possible to use background deduplication with a very aggressive deduplication schedule to maintain storage efficiency all the time. Doing so is essentially like running deduplication inline. It has proven to be valuable for VDI and server virtualization use cases. For more information about always-on deduplication and VMware implementation, refer to TR-4428: NetApp All Flash FAS Solution for VMware Horizon 6 and vSphere Virtual Volumes, TR-4307: NetApp All Flash FAS Solution for Nonpersistent Desktops with VMware Horizon View, and TR-4335: NetApp All Flash FAS Solution for Persistent Desktops with VMware Horizon View.

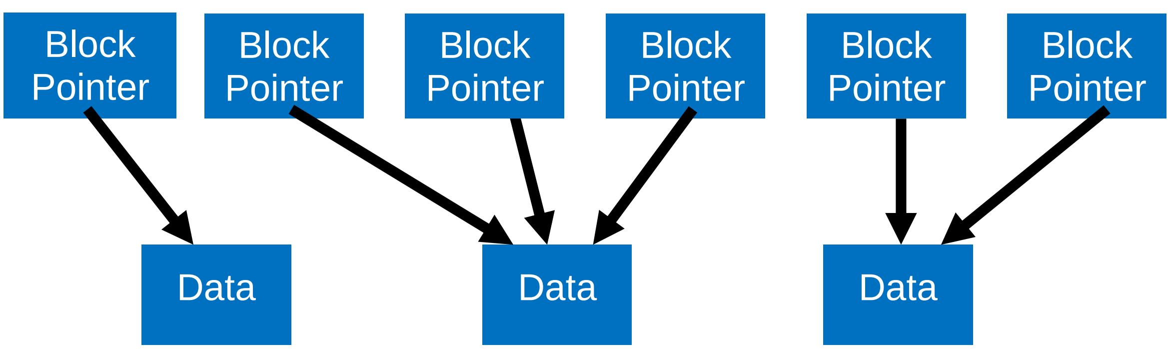
NetApp does not recommend using always-on deduplication on HDD and Flash Pool systems. That is because running deduplication continuously in the background can have an adverse effect on the performance of the system, especially when disks are a bottleneck.

4.1 Deduplicated FlexVol Volumes and Infinite Volume Data Constituents

A deduplicated volume is a FlexVol volume or data constituent that contains shared data blocks. Data ONTAP supports shared blocks in order to optimize storage space consumption. Basically, in one FlexVol volume or data constituent, there is the ability to have several references to the same data block.

In Figure 9, the number of physical blocks used on the disk is 3 (instead of 6), and the number of blocks saved by deduplication is 3 (6 minus 3). In this document, these blocks are referred to as used blocks and saved blocks.

Figure 9) Data structure in a deduplicated FlexVol volume or data constituent.



Each data block has a reference count that is kept in the volume or data constituent metadata. In the process of sharing the existing data and eliminating the duplicate data blocks, block pointers are altered. The reference count for the block that remains on disk with the block pointer is increased. The reference count for the block that contained the duplicate data is decremented. When no block pointers have a reference to a data block, the block is released.

NetApp deduplication technology allows duplicate 4KB blocks anywhere in the FlexVol volume or data constituent to be deleted, as described in the following sections.

The maximum sharing for a block is 32,767. This means, for example, that if there are 64,000 duplicate blocks, deduplication reduces that number to only 2 blocks.

4.2 Deduplication Metadata

The core enabling technology of deduplication is fingerprints. Fingerprints are unique digital signatures for every 4KB data block in the FlexVol volume or data constituent.

Inline Deduplication (Data ONTAP 8.3.2 and Later)

When data is written to the system, the inline operation scans the incoming blocks, creates a fingerprint, and stores the fingerprint in memory. The location where the fingerprints are stored in memory is called the hash store. The hash store is nonpersistent across reboots and volume offline operations. Because the goal is to dedupe across the blocks that reside in memory and the buffer cache, the hash store is small enough to be stored in memory. The hash store also is extremely fast when performing insert, lookup, and delete operations. The contents of the hash store are cycled based on a least recently used algorithm. This algorithm evicts the least recently used block to make room for the new block whenever the hash store gets full.

After the fingerprint file is created, fingerprints are checked for duplicates. When duplicates are found, a byte-by-byte comparison of the blocks in memory is done so that the in-memory blocks are indeed identical. If the blocks are found to be identical, the block's pointer is updated to the already existing in-memory data block, and the new (duplicate) in-memory data block is released.

Postprocess Deduplication

When deduplication runs for the first time on a volume with existing data, it scans the blocks in the FlexVol volume or data constituent. Deduplication then creates a fingerprint database that contains a sorted list of all fingerprints for used blocks in the FlexVol volume or data constituent.

After the fingerprint file is created, fingerprints are checked for duplicates; when duplicates are found, a byte-by-byte comparison of the blocks is done to make sure that the blocks are identical. If they are found to be identical, the block's pointer is updated to the already existing data block, and the new (duplicate) data block is released.

Releasing a duplicate data block entails updating the block pointer, incrementing the block reference count for the already existing data block, and freeing the duplicate data block. In real time, as additional data is written to the deduplicated volume or data constituent, a fingerprint is created for each new block and written to a change log file. When deduplication is run subsequently, the change log is sorted, its sorted fingerprints are merged with those in the fingerprint file, and then deduplication processing occurs.

There are two change log files. In that way, as deduplication runs and merges the fingerprints of new data blocks from one change log file into the fingerprint file, the second change log file logs the fingerprints of new data written to the FlexVol volume or data constituent during the deduplication process. The roles of the two files are then reversed the next time that deduplication runs. (For those familiar with Data ONTAP usage of NVRAM, this process is analogous to when a switch is made from one half to the other to create a consistency point.)

Here are some additional details about the deduplication metadata:

- There is a fingerprint record for every 4KB data block, and the fingerprints for all physical data blocks in the FlexVol volume or data constituent are stored in the fingerprint database file.
- Fingerprints are not deleted automatically from the fingerprint file when data blocks are freed. When a threshold of the number of new fingerprints is 20% greater than the number of data blocks used in the volume, the stale fingerprints are deleted. This step can also be taken by a manual operation using the advanced mode command volume efficiency check.

- The change log file size limit is set to 1% of the volume size except if it is a NetApp SnapVault® destination; then the change log file size limit is set to 4% of the volume size. The change log file size is relative to the volume size limit. The space assigned for the change log in a volume is not reserved.
- The deduplication metadata for a FlexVol volume or data constituent is inside the aggregate, and it is also stored in the FlexVol volume or data constituent. The copy inside the aggregate is used as the working copy for all deduplication operations. Change log entries are appended to the copy in the FlexVol volume or data constituent.
- During the upgrade of a major Data ONTAP release such as 8.3, the fingerprint and change log files are automatically upgraded to the new fingerprint and change log structure the first time volume efficiency operations start after the upgrade completes. This is a one-time operation, and it can take a significant amount of time to complete, during which you can see an increased amount of CPU on your system. You can see the progress of the upgrade using the volume efficiency `show –instance` command.
- The deduplication metadata requires a minimum amount of free space in the aggregate. This amount is equal to 3% of the total amount of physical data for all deduplicated FlexVol volumes or data constituents within the aggregate. Each FlexVol volume or data constituent should have 4% of the total amount of physical data's worth of free space, for a total of 7%.
- The deduplication fingerprint files are located inside both the volume or data constituent and the aggregate. This feature allows the deduplication metadata to follow the volume or data constituent during operations such as a NetApp volume SnapMirror® operation. If the volume or data constituent ownership is changed because of a disaster recovery operation with volume SnapMirror, deduplication automatically recreates the aggregate copy of the fingerprint database from the volume or data constituent copy of the metadata the next time it runs. This operation is much faster than recreating all the fingerprints from scratch.

Deduplication Metadata Overhead

Inline Deduplication (Data ONTAP 8.3.2 and Later)

No storage overhead is associated with deduplication in the case of inline deduplication because all the metadata is stored in memory. However, the performance overhead resulting from the entire inline deduplication operation is <1% in latency increase and throughput reduction.

Postprocess Deduplication

Although deduplication can provide substantial storage savings in many environments, a small amount of storage overhead is associated with it. The deduplication metadata for a FlexVol volume or data constituent is located inside the aggregate. A copy is also stored in the FlexVol volume or data constituent.

The guideline for the amount of extra space that should be left in the FlexVol volume or data constituent and aggregate for the deduplication metadata overhead is as follows:

- **Volume or data constituent deduplication overhead,** Each FlexVol volume or data constituent with deduplication enabled, up to 4% of the physical amount of data written to that volume or data constituent, is required to store FlexVol volume or data constituent deduplication metadata. This value never exceeds the maximum FlexVol volume or data constituent size times 4%.
- **Aggregate deduplication overhead.** Each aggregate that contains a FlexVol volume or data constituent with deduplication enabled, up to 3% of the physical amount of data contained in all of those FlexVol volumes or data constituents with deduplication enabled within the aggregate, is required to store the aggregate deduplication metadata.

Deduplication Metadata Overhead Example: Data ONTAP 8.3.x

If you have 100GB of logical data and get 50GB of savings in a single FlexVol volume, then you have 50GB of physical data. Given this, there should be 2GB (4% of 50GB) of available space in the volume and 1.5GB of space available in the aggregate. As a second example, consider a 2TB aggregate with four volumes, each 400GB in size, in the aggregate. Three volumes are to be deduplicated with the following:

Vol1: 100GB of logical data, which has 50% savings (50% of 100GB) = 50GB physical data

Vol2: 200GB of logical data, which has 25% savings (75% of 200GB) = 150GB physical data

Vol3: 300GB of logical data, which has 75% savings (25% of 300GB) = 75GB physical data

The required amount of space for deduplication metadata is as follows:

Vol1: 2GB (50GB * 4%)

Vol2: 6GB (150GB * 4%)

Vol3: 3GB (75GB * 4%)

The aggregate needs a total of 8.25GB ((3% of 50GB) + (3% of 150GB) + (3% of 75GB)) = 1.5+4.5+2.25=8.25GB) of space available in the aggregate.

The primary fingerprint database, otherwise known as the working copy, is outside the volume or data constituent in the aggregate and is therefore not captured in Snapshot copies. The change log files and a backup copy of the fingerprint database are located within the volume or data constituent and are therefore captured in Snapshot copies. Having the primary (working) copy of the fingerprint database outside the volume or data constituent enables deduplication to achieve higher space savings. However, the other temporary metadata files created during the deduplication operation are still placed inside the volume or data constituent. These temporary metadata files are deleted when the deduplication operation is complete. However, if Snapshot copies are created during a deduplication operation, these temporary metadata files can get locked in Snapshot copies, and they remain there until the Snapshot copies are deleted.

5 NetApp Data Compression

NetApp data compression is a software-based solution that provides transparent data compression. It can be run inline or postprocess and also includes the capability to compress existing data. No application changes are required to use NetApp data compression. This process is enabled and managed by using a simple CLI or GUI such as System Manager or NetApp OnCommand Unified Manager.

Data ONTAP 8.3.1 introduces adaptive compression, an inline compression method that works with I/Os of varying sizes. The performance impact is minimal, and in some cases enabling adaptive compression improves overall performance. The compression method available in Data ONTAP 8.3 and earlier is still available and is now called secondary compression.

Difference Between Adaptive Compression and Secondary Compression

Adaptive Compression

Adaptive compression combines fewer blocks of data into a compression group (8K). The compression group is then compressed and stored as a single block. When a user requests data from this compression group, less time is taken to decompress and provide that data to the user, thereby improving the read performance. In general, adaptive compression is better suited for random workloads. Adaptive compression provides relatively fewer savings as compared to secondary compression, but with better performance.

Secondary Compression

Secondary compression combines large groups of data blocks into a compression group (32K). The compression group is then compressed and stored as fewer blocks, which reduces the size of the data to a large extent, thereby increasing the free space in the storage system. In general, secondary compression is better suited for sequential workloads.

Both secondary compression and adaptive compression are supported on all types of disk media (HDD, AFF, and Flash Pool).

5.1 How NetApp Data Compression Works

NetApp data compression does not compress the entire file as a single contiguous stream of bytes. Doing so would be prohibitively expensive to service small reads or overwrites from part of a file. That is because it requires the entire file to be read from disk and uncompressed before the request can be served. Doing so would be especially difficult on large files.

To avoid this issue, NetApp data compression compresses a small group of consecutive blocks, known as a compression group. In this way, when a read or an overwrite request comes in, you need to read only a small group of blocks, not the entire file. This process optimizes read and overwrite performance and enables greater scalability in the size of the files being compressed.

Compression Groups

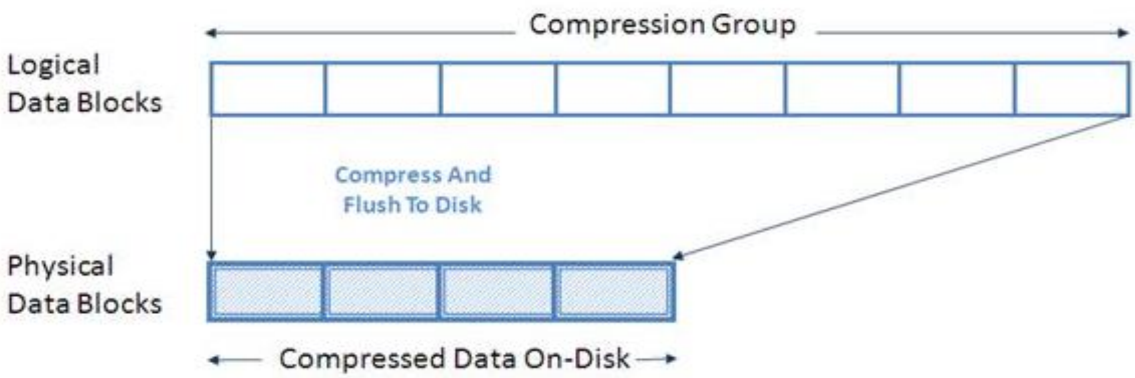
The NetApp compression algorithm divides a file into compression groups (CGs). The file must be larger than 8K or it is skipped for compression and written to disk uncompressed. Compression groups are a maximum of 8K or 32K, depending on the compression type. For secondary compression, the compression group size is 32K; for adaptive compression, the compression group size is 8K.

A compression group contains data from one file only. A single file can be contained within multiple compression groups. If a file were 60K, it would be contained within two secondary compression groups. The first would be 32K, and the second 28K. Similarly, a 15K file would be contained within two adaptive compression groups: 8K and 7K.

Compressed Writes

NetApp handles compression write requests at the compression group level. Each compression group is compressed separately. The compression group is left uncompressed unless a savings of at least 25% (in the case of 32K CGs) or at least 50% (in the case of 8K CGs) can be achieved on a per–compression group basis. This policy optimizes savings while minimizing resource overhead.

Figure 10) Compression write request handling.



Because compressed blocks contain fewer blocks to be written to disk, compression reduces the amount of write I/Os required for each compressed write operation. This reduction not only lowers the data footprint on disk, but can also decrease the time to complete backups. See the "Feature Interoperability" section for details about volume SnapMirror.

Compressed Reads

When a read request comes in, we read only the compression group (or groups) that contains the requested data, not the entire file. This approach optimizes the amount of I/O being used to service the request. When reading compressed data, only the required compression group data blocks are transparently decompressed in memory. The data blocks on disk remain compressed. This process has much less overhead on the system resources and read service times.

In summary, the NetApp compression algorithm is optimized to reduce overhead for both reads and writes.

5.2 When Data Compression Runs

NetApp data compression, both secondary compression and adaptive compression, can be configured to run either inline or postprocess, depending on the workload, performance, and savings requirements.

Inline Operations

During inline data compression, as the data is sent to the storage system, it is compressed in memory before being written to the disk. The advantage of this implementation is that it can reduce the amount of write I/O. This implementation option can affect your write performance and thus should not be used for performance-sensitive environments on HDD configurations without proper testing to understand the impact. The All Flash FAS (AFF) systems introduced with Data ONTAP 8.3.1 and the Flash Pool systems are exceptions. Inline compression can be used for primary workloads on AFF and Flash Pool systems.

For Flash Pool systems, make sure that the cache size is configured as per Flash Pool best practices. For more information about Flash Pool, see TR-4070: "Flash Pool Design and Implementation Guide."

Here are some additional details about inline compression:

- To provide the fastest throughput, inline compression compresses most new writes but skips more performance-intensive compression operations. An example of a performance-intensive compression operation is a small (<4K) partial file overwrite. Postprocess compression, if enabled, tries to compress any data that was skipped by inline compression.
- Inline adaptive compression compresses data only if the compression savings are >50%.
- Inline secondary compression compresses data only if the compression savings are >25%.
- The incompressible data detection (IDD) option is supported only on secondary compression. The enhanced compression algorithm used in Data ONTAP 8.3.1 is intelligent enough to identify incompressible data and return more quickly. There is no need to use the IDD option on 8.3.1. The option is provided for backward compatibility:
  - IDD option is not supported on AFF configurations.
  - IDD option is deprecated in ONTAP 9.

Postprocess Operations

NetApp data compression includes the capability to run postprocess compression in addition to inline compression. Postprocess compression uses the same schedule as deduplication. If compression is enabled when the volume efficiency schedule initiates a postprocess operation, the operation runs compression first, followed by deduplication. Postprocess compression also includes the capability to compress data that existed on disk prior to compression being enabled. Similar to inline compression, postprocess compression compresses data only if the compression savings are >50% (adaptive compression) or >25% (secondary compression).

If both inline and postprocess compression are enabled, then postprocess compression tries to compress only blocks that are not already compressed. These blocks include those that were bypassed by inline compression, such as small partial-file overwrites.

Postprocess compression is not supported on AFF configurations.

5.3 New Features and Behavior of Compression

New features introduced in Data ONTAP 8.3.1:

- All Flash FAS support: Compression is now officially supported on All Flash FAS series controllers (AFF80XX series controllers).
- There is newly introduced adaptive compression support.
- Read performance improvements: Numerous performance optimizations have gone into Data ONTAP 8.3.1 compression to enhance read performance, especially random read performance. The result of these optimizations is that Data ONTAP 8.3.1 inline compression performance is comparable to, or better than, Data ONTAP 8.3.0 performance.
- Complete Flash Pool integration: Starting with Data ONTAP 8.3.1, Flash Pool can cache both read and write compressed blocks.
- Subfile cloning is supported on adaptive compression–enabled volumes.
- You can size adaptive compression for AFF configurations using the NetApp sizing tool. Note: Sizing is available only for NetApp representatives.

New features introduced in Data ONTAP 8.3.2:

- Support for background compression scanner to perform in-place conversion of uncompressed data in a volume to a compressed format on AFF systems.
- Performance improvement: Compression operations now run in their own domain, which is separate from the data-serving domain, resulting in a performance improvement of up to 15% in latency and throughput.

Compression Behavior Based on Configuration

All Flash FAS (AFF) configuration:

- Supports both adaptive and secondary compression.
- Supports only inline compression.
- Adaptive inline compression is enabled by default on all new volumes (only when the effective cluster version is 8.3.1 and all the nodes in the cluster are on 8.3.1).
- New volumes have an inline-only storage efficiency policy set by default on SAN-optimized AFF systems.
- Secondary compression can be enabled manually, if required.
- Compression can be turned on and off at the volume level.
- A volume can have either adaptive compression or secondary compression but not both. Active file systems and Snapshot copies can have different compression types.

Flash Pool configuration:

- Supports both adaptive and secondary compression.

- Supports both inline compression and postprocess compression.
- Compression is not enabled by default on new volumes. Compression has to be enabled manually.
- Adaptive compression is enabled by default as long as there are SSDs in the aggregate and the effective cluster version is 8.3.1.
- Secondary compression can be enabled by specifying the compression type option while turning on compression.
- Postprocess/background compression is enabled by default. Users can choose background-only, inline-only, or inline+background compression.
- For inline-only compression, users have to manually assign the inline-only storage efficiency policy to the volume.

All-HDD configuration:

- Supports both adaptive and secondary compression.
- Supports both inline compression and postprocess compression.
- Compression is not enabled by default on new volumes. Compression has to be enabled manually.
- Secondary compression is enabled by default.
- Adaptive compression can be enabled by specifying the compression type option while turning on compression.
- Postprocess/background compression is enabled by default. Users can choose background-only, inline-only, or inline+background compression.
- For inline-only compression, users have to manually assign the inline-only storage efficiency policy to the volume.

6 General Compression and Deduplication Features

Both compression and deduplication are enabled on a per-volume basis for FlexVol volumes or Infinite Volumes. Both processes can be enabled on any number of FlexVol volumes or data constituents in a storage system. Enabling compression and deduplication on an Infinite Volume automatically enables them on all data constituents; you can’t enable them on some data constituents and not others.

You can run inline compression and inline deduplication without a scheduled or manually started background efficiency operation configured on the volume. With a new storage virtual machine, two predefined efficiency policies, inline-only and default, are automatically created. You can configure a volume to use the inline-only efficiency policy and enable inline compression to run inline compression on the volume without any scheduled or manually started background efficiency operations.

Postprocess compression and deduplication share the same scheduler and can be scheduled to run in one of five different ways:

- Inline
- Scheduled on specific days and at specific times
- Manually, by using the command line
- Automatically, when 20% new data has been written to the volume
- SnapVault software based, when used on a SnapVault destination

Only one postprocess compression or deduplication process can run on a FlexVol volume or data constituent at a time. Up to eight compression and deduplication processes can run concurrently on eight different volumes or data constituents within the same NetApp storage system. If there is an attempt to run postprocess compression or deduplication processes beyond the maximum, the additional operations are placed in a pending queue and automatically started when there are free processes.

Postprocess compression and deduplication processes periodically create checkpoints so that if the scan is interrupted, it can continue from the last checkpoint.

6.1 Compression and Deduplication System Requirements

This section discusses the requirements to use deduplication and/or compression and the maximum amount of data that is compressed and deduplicated. Although the section discusses some basic points, we assume that the NetApp storage system is already installed and running and that the reader is familiar with basic NetApp administration.

Requirements Overview

Table 1) Overview of compression and deduplication requirements for clustered Data ONTAP.

Requirement	Deduplication	Compression
Data ONTAP version	Minimum clustered Data ONTAP 8.1	
License	Neither deduplication nor compression requires a license; you enable them on a volume-by-volume basis	
Hardware	All FAS and V-Series and FlexArray systems that are supported with clustered Data ONTAP	
Volume type supported	FlexVol or Infinite Volumes	
Aggregate type supported	64-bit	
Maximum volume size	Neither compression nor deduplication imposes a limit on the maximum volume or data constituent size supported; therefore, the maximum volume or data constituent limit is determined by the type of storage system regardless of whether deduplication or compression is enabled	
Supported protocols	All	

6.2 Maximum Logical Data Size Processing Limits

The maximum logical data size that is processed by postprocess compression and deduplication is 640TB regardless of the size of the volume or data constituent created. After this logical limit is reached, writes to the volume or data constituent continue to work; however, postprocess compression and deduplication fail with the error message “maximum logical data limit has reached.”

As an example in Data ONTAP 8.3.1, if you had a FAS8080 that has a 100TB maximum volume size and you created a 100TB volume or data constituent, the first 640TB of logical data compress and deduplicate normally. However, any additional new data written to the volume or data constituent after the first 640TB is not postprocess compressed or deduplicated until the logical data becomes less than 640TB. If inline compression is enabled on the volume or data constituent, data continues to be inline compressed until the volume or data constituent is completely full.

6.3 When to Enable Compression and/or Deduplication

Choosing when to enable compression or deduplication involves balancing the benefits of space savings against the potential overhead. Your savings and acceptable overhead will vary depending on the use case. You might find some solutions suited for the primary tier and others better suited for the backup/archive tier only. The amount of system resources the processes consume and the possible savings depend on the type of data. The performance impact varies in each environment, and NetApp highly recommends that the performance impact be fully tested and understood before implementing the processes in production.

Enabling compression or deduplication and deciding which compression type to use depend on the application characteristics, workload mix, I/O size, and performance requirements. Use the Storage Space Savings Estimation Tool (SSET) to estimate a dataset's compression and deduplication savings.

General Compression Usage Guidelines

- Use inline adaptive compression:

  When the workload is predominantly random writes or a mix of sequential and random writes

  When read I/Os are the same size or smaller than write I/Os and the workload is mostly reads

- Use inline secondary compression:

  When the workload is compressible and mostly large sequential I/Os (32KB or larger)

Inline Versus Postprocess Compression

Inline compression provides immediate space savings; postprocess compression first writes the blocks to disk as uncompressed and then at a scheduled time compresses the data.

Postprocess compression is useful for environments in which you want compression savings but don't want to incur a performance penalty associated with new writes. Common use cases that fall under this category are primary workloads on all HDD configurations.

Inline compression is useful for customers who aren't as performance sensitive and can handle some impact on new write performance as well as on CPU during peak hours. This type of compression is also useful for customers who want to minimize $/gigabyte and the PE cycle of SSD drives to prolong the life of flash media. Common use cases that fall under this category are primary and secondary workloads on AFF and Flash Pool configurations and secondary workloads on all HDD configurations.

Inline Versus Postprocess Deduplication

Inline deduplication provides immediate space savings; postprocess deduplication first writes the blocks to disk and then at a scheduled time dedupes the data.

Inline deduplication is useful primarily for copy offload and VDI deployments. It is especially useful for VM patch apply use cases, VM provisioning, and database migrate operations that result in many duplicate blocks in memory that can be eliminated by inline dedupe.

Inline deduplication might not be a good candidate for entry-level systems in which there is just enough memory to meet client I/O SLAs. Postprocess deduplication can be considered in such cases. In general, NetApp does not recommend deduplication for use cases in which data is overwritten at a rapid rate.

6.4 Space Savings

This section discusses the potential storage savings from deduplication and compression technologies.

The amount of space savings is primarily a function of data type, and different data types can yield different savings ratios. The amount also depends on which space-saving technology is being used. In some cases, deduplication alone or compression alone can yield better savings ratios as compared to those of the combined savings of deduplication and compression. NetApp recommends evaluating the dataset using the Space Savings Estimation Tool (SSET) to estimate savings ratios before implementing these processes in production.

A combination of synthetic datasets and real-world datasets has been used to determine the typical space savings. It is quite possible that the savings you get do not match the numbers in Table 2. The savings ratios mentioned are only indicative.

In general, secondary compression provides better savings compared to those of adaptive compression. Adaptive compression provides better performance compared to that of secondary compression. The maximum savings that one can expect with adaptive compression is 2:1. When used in combination with data compaction (ONTAP 9 and later), adaptive compression can yield >2:1 savings.

Table 2) Typical deduplication and adaptive compression space savings.

Dataset Type	Compression Savings	Deduplication Savings	Combined Savings
Home directories	1.2:1–1.8:1	1.2–1.5:1	1.4:1–1.8:1
Software development	1.4:1–1.8:1	1.43:1	1.8:1–2:1
Virtual servers	1.5:1–2:1	2:1	2:1
Geoseismic files	1.67:1	1.03:1	1.67:1
Oracle database active data	1.65:1	N/A	1.65:1
Oracle database Archived or read-only data	1.8:1–2:1	N/A	1.8–2:1
SQL Server database	1.65:1	N/A	1.65:1
Microsoft Exchange 2007	1.54:1	1.03:1	1.54:1
Microsoft Exchange 2010	1.54:1	1.18:1	1.67:1
Video files	1.06:1	1.02:1	1.05:1

Table 3) Typical deduplication and secondary compression space savings.

Dataset Type	Compression Savings	Deduplication Savings	Combined Savings
Home directories	1.2:1–2.2:1	1.2:1–2:1	1.8:1–2.8:1
Software development	2.22:1	1.43:1	2.5:1–3.5:1
Virtual servers	2.22:1	3.33:1	3.33:1
Geoseismic files	1.67:1	1.03:1	1.67:1
Oracle database active data	1.65:1	N/A	1.65:1

Dataset Type	Compression Savings	Deduplication Savings	Combined Savings
Oracle database archived or read-only data	2.8:1–3.3:1	N/A	2.86–3.33:1
SQL Server database	1.65:1	N/A	1.65:1
Microsoft Exchange 2007	1.54:1	1.03:1	1.54:1
Microsoft Exchange 2010	1.54:1	1.18:1	1.67:1
Video files	1.06:1	1.02:1	1.05:1

In the NetApp implementation, compression is run before deduplication. Doing so provides the ability to use inline compression to get immediate space savings from compression followed by additional savings from deduplication. In our testing of other solutions, we found that better savings were achieved by running compression before deduplication.

6.5 Performance

This section discusses the performance aspects of data compression and deduplication.

Because compression and deduplication are part of Data ONTAP, they are tightly integrated with the NetApp WAFL file structure. Because of this fact, compression and deduplication are optimized to perform with high efficiency. The processes are able to leverage the internal characteristics of Data ONTAP to perform compression and decompression, create and compare digital fingerprints, redirect data pointers, and free up redundant data areas.

However, the following factors can affect the performance of compression and deduplication and the I/O performance of compressed and deduplicated volumes:

- The application and the type of dataset being used
- The data access pattern (for example, sequential versus random access, the size of the I/O)
- The amount of duplicate data
- The compressibility of the data
- The amount of total data
- The average file size
- The nature of the data layout in the volume
- The amount of changed data between compression and deduplication runs
- The number of concurrent compression and deduplication processes running
- The number of volumes or data constituents with compression and deduplication enabled
- The hardware platform: the amount of CPU in the system
- The Data ONTAP release
- The amount of load on the system
- The disk types: SSD, ATA/SAS, and the RPM of the disk
- The number of disk spindles in the aggregate

- The priority set for the volume efficiency operation assigned to the volume

Compression and deduplication can be scheduled to run during nonpeak hours. Doing so allows the bulk of the overhead on the system during nonpeak hours. When there is a lot of activity on the system, compression and deduplication run as background processes and limit the system's resource usage. When there is not a lot of activity on the system, compression and deduplication speed increases and can use all available system resources. The potential performance impact should be fully tested before implementation.

Because compression and deduplication run on a per FlexVol volume or per–data constituent basis, the more data constituents or FlexVol volumes per node that run in parallel, the greater the impact on system resources. NetApp recommends that when using compression and deduplication, you stagger the volume efficiency schedule for volumes to help control the overhead.

Storage QoS gives users the choice of running a volume efficiency operation with a priority of either best effort or background. This feature enables the administrator to define how volume efficiency operations compete for resources with user workloads and other system processes not running in the background.

When considering whether to add compression or deduplication, use standard sizing and testing methods you would use when considering adding applications to the storage system. It is important to understand how inline compression affects your system and how long postprocess operations take in your environment. It is also important to know whether you have the bandwidth to run these operations with acceptable impact on the applications running on your storage system.

We optimized compression performance considerably in Data ONTAP 8.3.1, especially for AFF and Flash Pool configurations. For AFF use cases that benefit from compression and/or deduplication, we recommend enabling inline adaptive compression and/or deduplication for primary workloads.

As an example, on Data ONTAP 8.3.1, for a 100% read workload with 8K I/O size, AFF8080 has been measured to deliver ~30% more IOPS at 1ms with compression than that of Data ONTAP 8.3.0 with no compression.

However, most workloads are a mix of reads and writes. For a mixed 50/50 R/W workload with predominantly 8K block size I/O at low to medium load with latency <1ms, AFF8080 delivers 15% to 20% higher IOPS with inline compression. A smaller platform such as the AFF8020 delivers roughly the same performance as that of Data ONTAP 8.3.0. Other platforms' results are expected to be between these ranges.

In higher-load scenarios, when latency trends to 2ms or beyond, IOPS may be 10% to 15% lower. Similarly, workloads with mixed/large block size I/O might see 10% to 15% lower performance compared with that of Data ONTAP 8.3.0.

Although we optimized compression to minimize impact on your throughput, there might still be an impact even if you use only postprocess compression. The impact can occur because some data in memory still must be uncompressed when servicing reads. The impact continues so long as the data is compressed on disk regardless of whether compression is disabled on the volume at a future point.

Because of these factors, NetApp recommends that performance with compression and deduplication be measured in a test setup and taken into sizing consideration before deploying the processes in performance-sensitive solutions, especially for HDD configurations.

7 Feature and Configuration Interoperability

Table 4 shows the support matrix information for feature and configuration combinations.

Table 4) Feature configuration interoperability.

Feature	AFF Support	Default On?	Flash Pool Support	Default On?	HDD Support	Default On?
Inline zero-block deduplication	Yes	Yes	Yes	No	Yes	No
Inline deduplication	Yes	Yes	Yes	No	Yes	No
Postprocess deduplication	Yes	No	Yes	No	Yes	No
Inline adaptive compression	Yes	Yes	Yes	No	Yes	No
Postprocess adaptive compression	Not supported		Yes	No	Yes	No
Inline secondary compression	Yes	No	Yes	No	Yes	No
Postprocess secondary compression	Not supported		Yes	No	Yes	No
Inline data compaction	Yes	Yes	Yes	No	Yes	No

8 General Workload-Based Storage Efficiency Guidelines

Table 5 shows some guidelines for when to use compression, deduplication, data compaction, and/or inline zero-block deduplication. These guidelines are not rules; your environment may have different performance requirements for specific use cases. NetApp highly recommends that the performance impact be fully tested and understood before you decide to implement these processes in production.

Table 5) Workload-based storage efficiency guidelines.

Workload	**Storage Efficiency Guidelines**		
	All Flash FAS (AFF)	Flash Pool (Sized as per Flash Pool Best Practices)	Hard Disk Drives
Database (Oracle, SQL Server)	For primary and secondary workloads, use: • Adaptive inline compression • Inline zero-block deduplication • Inline deduplication (Data ONTAP 8.3.2 and later) • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Inline zero-block deduplication • Inline deduplication (Data ONTAP 8.3.2 and later) • Inline data compaction (ONTAP 9 and later)	For primary workloads, use: • Inline zero-block deduplication For secondary workloads, use: • Adaptive inline compression • Adaptive background compression • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)

VDI and SVI	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline deduplication (Data ONTAP 8.3.2 and later) • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline deduplication (Data ONTAP 8.3.2 and later) • Inline data compaction (ONTAP 9 and later)	For primary workloads, use: • Deduplication • Inline zero-block deduplication For secondary workloads, use: • Adaptive inline compression • Adaptive background compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)
Exchange	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Set schedule to off-peak hours • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Inline secondary compression • Background secondary compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)
File services	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Adaptive background compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)
Mixed workloads	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary and secondary workloads, use: • Adaptive inline compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)	For primary workloads, use: • Deduplication • Inline zero-block deduplication For secondary workloads, use: • Adaptive inline compression • Adaptive background compression • Deduplication • Inline zero-block deduplication • Inline data compaction (ONTAP 9 and later)

For additional details about SQL implementation, refer to NetApp AFF8080 EX Performance and Server Consolidation with Microsoft SQL Server 2014.

For additional details about Oracle implementation, refer to NetApp AFF8080 EX Performance and Server Consolidation with Oracle Database and TR-3633: Oracle Databases on Data ONTAP.

For additional details about Exchange, refer to TR-3578: Microsoft Exchange Server 2007 Best Practices Guide or TR-3824: Storage Efficiency and Best Practices for Microsoft Exchange Server 2010.

For additional details about VMware implementation, refer to TR-3428: NetApp and VMware Best Practices Guide, TR-3749: NetApp and VMware vSphere Storage Best Practices, TR-4428: NetApp All Flash FAS Solution for VMware Horizon 6 and vSphere Virtual Volumes, TR-4307: NetApp All Flash FAS Solution for Nonpersistent Desktops with VMware Horizon View, and TR-4335: NetApp All Flash FAS Solution for Persistent Desktops with VMware Horizon View.

9 Best Practices

9.1 Best Practices for Optimal Savings and Minimal Performance Overhead

This section describes best practices and lessons learned from internal tests and customer deployments:

- Both deduplication and compression consume system resources and can alter the data layout on disk. Because of the application's I/O pattern and the effect of deduplication and compression on the data layout, the read and write I/O performance can vary. The space savings and the performance impact depend on the application and the data contents.
- NetApp recommends that the performance impact of deduplication and compression be carefully considered and measured in a test setup and taken into consideration for sizing before deploying the processes in performance-sensitive solutions. For information about the impact of deduplication and compression on other applications, contact the specialists at NetApp for their advice and for test results for your particular application.
- If there is only a small amount of new data, run deduplication infrequently, because there is no benefit in running it frequently in such a case, and it consumes system resources. The frequency for running deduplication depends on the rate of change of the data in the volume.
- The more concurrent compression and deduplication processes you run on a system, the more system resources are consumed.
- If NetApp Snapshot copies are required, run compression and deduplication processes before creating the Snapshot copies to minimize the amount of data that gets locked in to the copies. (Make sure that the compression and deduplication processes complete before creating the Snapshot copy.) If a Snapshot copy is created on a volume before the deduplication processes complete, the result is likely to be smaller space savings. If a Snapshot copy is created on a volume before the compression processes complete, the result is likely to be that more space is used by the Snapshot copies.
- For deduplication to run properly, leave some free space for the deduplication metadata. For information about how much extra space to leave in the FlexVol volume, data constituent, and aggregate, see section 4.2, "Deduplication Metadata."

10 Upgrade and Revert

Major and minor nondisruptive upgrades to clustered Data ONTAP are supported with both compression and deduplication.

NetApp recommends that no active postprocess compression or deduplication operations run during the nondisruptive upgrade:

- Perform the Data ONTAP upgrade during a time when compression and deduplication operations are not scheduled to run.
- Determine whether any compression and deduplication operations are active and, if they are, stop them until the Data ONTAP upgrade is complete.
- You can use the volume efficiency show command to determine whether the status of compression and deduplication is active or idle. On a system with compression and deduplication enabled, the output of the `volume efficiency show` command is similar to the following:

```
Vserver     Volume   State      Status    Progress
Vsone       vol1     Enabled    Active    122 GB Scanned, 25 GB
Compressed
Vsone       vol2     Enabled    Active    164 GB Searched
Vsone       vol3     Enabled    Active    41 GB (45%) Done
Vsone       vol4     Enabled    Idle      for 00:02:43
```

- You can use the `volume efficiency stop` command to abort the active compression and deduplication operations on the volume and use the `volume efficiency start` command to restart it.

For specific details and requirements for performing a nondisruptive upgrade on your system, refer to Upgrade Advisor in the AutoSupport tool if you have AutoSupport enabled. Otherwise, refer to the release notes for the version of Data ONTAP to which you are upgrading.

10.1 Upgrading to a Later Version of Data ONTAP

Upgrading to a later version of Data ONTAP is a nondisruptive operation regardless of whether the volumes have deduplication enabled or not. However, be aware that part of an upgrade to a later major release on a deduplication-enabled volume or data constituent includes upgrading the deduplication metadata. As a result, you might notice that volume efficiency operations take longer to complete during this operation. This is a one-time increase on each FlexVol volume or data constituent that had deduplication enabled before the upgrade and happens only after an upgrade. Subsequent volume efficiency operations complete as expected.

The process of upgrading the deduplication metadata is done in two stages. The first stage is creating new change log metafiles in the FlexVol volume or data constituents and is done automatically during the upgrade, at boot time, while the volume is coming online.

The second stage is sorting the change logs and fingerprints created from the previous version and recreating with the new format. This work is done automatically the first time volume efficiency operations are run on each FlexVol volume or data constituent that had deduplication enabled before the upgrade. The second step can take a significant amount of time to complete, during which you can see an increased amount of CPU on your system and see the progress of the upgrade using the `volume efficiency show -instance` command. The amount of time for this work to complete might be different for each FlexVol volume and data constituent. The time depends on the size of the change log and fingerprint database, the storage efficiency priority used, and the amount of available resources. During this phase there are no checkpoints, so if you need to stop deduplication, wait for this upgrade process to complete first.

Even in the case of compression, upgrading to a newer version of Data ONTAP is a nondisruptive operation.

However, because adaptive compression was newly introduced in Data ONTAP 8.3.1 and the AFF configuration supports only inline compression, compression behavior on AFF configurations is slightly different after an 8.3.1 upgrade. This difference is in comparison to compression behavior on HDD and Flash Pool configurations. Table 6 and Table 7 detail the exact behavior.

Table 6) Compression behavior after a Data ONTAP 8.3.1 upgrade: AFF configuration.

AFF: Pre-8.3.1	AFF: Post-8.3.1 Upgrade
Volume 1: no compression	Volume 1: no compression: • You can enable inline adaptive and secondary compression on Volume 1. • Existing data on the volume remains uncompressed. • In Data ONTAP 8.3.1, you cannot explicitly convert the existing data to a compressed format. Existing data is converted to a compressed format when the old data is overwritten. • Starting in Data ONTAP 8.3.2, you can compress existing uncompressed data using the background scanner. Manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 2: inline secondary compression	Volume 2: inline secondary compression: • Existing compressed data remains compressed in a secondary compression format. • If you want to enable inline adaptive compression, first rehydrate the volume and then enable inline adaptive compression. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

AFF: Pre-8.3.1	AFF: Post-8.3.1 Upgrade
Volume 3: inline secondary compression + background secondary compression	Volume 3: inline secondary compression: • Background compression is disabled because it is not supported on AFF configurations. • If you want to enable inline adaptive compression, first rehydrate the volume and then enable inline adaptive compression. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 4: background secondary compression	Volume 4: no compression: • Background compression is disabled on the volume because it is not supported on AFF configurations. • Existing compressed data remains compressed in a secondary compression format. • You can enable inline secondary compression. • If you want to enable inline adaptive compression, you have to first rehydrate the volume and then enable inline adaptive compression. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Table 7) Compression behavior after a Data ONTAP 8.3.1 upgrade: FAS configuration.

FAS: Pre-8.3.1	FAS: Post-8.3.1 Upgrade
Volume 1: no compression	Volume 1: no compression: • You can enable inline adaptive or inline secondary compression on Volume 1. • Existing data on the volume remains uncompressed. • Existing data is converted to a compressed format when the old data is overwritten or background compression is used to compress the existing data.
Volume 2: inline secondary compression	Volume 2: inline secondary compression: • Existing compressed data remains compressed in a secondary compression format. • To enable inline/background adaptive compression, first rehydrate the volume and then enable adaptive compression.
Volume 3: inline secondary compression + background secondary compression	Volume 3: inline secondary compression + background secondary compression: • To enable inline/background adaptive compression, first rehydrate the volume and then enable adaptive compression.
Volume 4: background secondary compression	Volume 4: background secondary compression: • You can enable inline secondary compression. • To enable inline/background adaptive compression, first rehydrate the volume and then enable adaptive compression.

10.2 Reverting to an Earlier Version of Clustered Data ONTAP

When you are considering reverting a system to an earlier version of clustered Data ONTAP, we typically recommend contacting NetApp Customer Success Services for assistance.

Scenarios in which customers can revert their NetApp storage systems without guidance from NetApp Customer Success Services include:

- If the NetApp storage system is new and has not yet been used, then the `volume efficiency revert_to` command can be used without guidance from NetApp Customer Success Services.
- If the NetApp storage system is being used as a test system and no production data is present, then the `volume efficiency revert_to` command can be used without guidance from NetApp Customer Success Services.

When reverting a system from clustered Data ONTAP 8.3.1 or later to Data ONTAP 8.3 or earlier, you must undo all adaptive compression savings from all nodes of the cluster before reverting. This is because adaptive compression is not supported on releases prior to Data ONTAP 8.3.1.

You should not attempt reverting in any other scenario without assistance from NetApp Customer Success Services: http://www.netapp.com/us/support/ngs-contacts.html.

The `volume efficiency revert_to` command provides sufficient guidance to successfully complete the revert process, including the process for reverting deduplicated volumes, in these scenarios. If you encounter problems during or after the revert process, contact NetApp Customer Success Services.

11 Interoperability

NetApp has several products that work with compression, deduplication, and compaction. This section discusses how compression, deduplication, and compaction interact with other NetApp products and features.

11.1 Data Protection

Snapshot Copies

Snapshot copies lock blocks on disk that cannot be freed until the Snapshot copy expires or is deleted. After a Snapshot copy of data is made on a volume, any subsequent changes to data contained in that Snapshot copy require additional disk space until the Snapshot copy is deleted or expires. The same is true with deduplication- and/or compression-enabled volumes.

Active File System (AFS) and any Snapshot copy can contain only one kind of compression: either adaptive or secondary. It is possible to have different types of compression between AFS and Snapshot copies:

- For example, AFS with adaptive compression and Snapshot1 with secondary compression

Some best practices to achieve the best space savings from deduplication- and/or compression-enabled volumes that contain Snapshot copies are:

- Run compression and deduplication before creating new Snapshot copies.
- Wait for postprocess compression and deduplication to complete before creating new Snapshot copies.
- Change the Snapshot policy to none until the compression of existing data completes.
- If running deduplication without compression, schedule deduplication only after significant new data has been written to the volume.
- Limit the number of Snapshot copies you maintain.
- If possible, reduce the retention duration of the Snapshot copies.
- Configure appropriate reserve space for the Snapshot copies.
- Delete as many Snapshot copies as possible before running compression against existing data.

Data compaction:

- Data present in Snapshot copies can be compacted by running the `'wafl pack'` command. Currently we only support packing of individual files in a Snapshot copy using the pack command.
- The command is only available in diagnostic mode and should only be used in consultation with NetApp Support.

Replication: Volume SnapMirror

Both compression and deduplication are supported with volume SnapMirror. Volume SnapMirror operates at the physical block level. Thus, when deduplication is enabled on the source, the data sent over the wire for replication remains deduplicated, and the savings are inherited at the destination. For compression, storage efficiency is preserved or lost during replication, depending on the following configuration:

- Source and destination volumes use the same compression type (either adaptive or secondary):
  - Logical replication with storage efficiency is used.
  - Storage efficiency (compressed and deduplicated blocks) is preserved during data transfer.

- Source and destination volumes use different compression types (adaptive to secondary or vice versa):
  - Logical replication without storage efficiency is used.
  - Blocks are uncompressed at the source and transferred to the destination.
  - Data is recompressed using the destination volume's compression type.

Note: NetApp always recommends using the same compression type on both source and destination volumes to prevent multiple decompression and recompression operations from occurring.

Additional notes:

- Data can be replicated from AFF systems to HDDs on FAS systems and vice versa.
- If you use compression on both the source and the destination volumes, both must be contained within a 64-bit aggregate.
- The volume SnapMirror update schedule is not tied to the compression and deduplication schedule.
- When configuring volume SnapMirror and compression and deduplication, consider the compression and deduplication schedule and the time you want to start a volume SnapMirror initialization. As a best practice, start volume SnapMirror initialization of a compressed and deduplicated volume after compression and deduplication are complete. Doing so prevents sending uncompressed and undeduplicated data and additional temporary metadata files over the network. If the temporary metadata files in the source volume are locked in Snapshot copies, they also consume extra space in the source and destination volumes.
- It is important when preparing to initialize a volume SnapMirror relationship on a compressed or deduplicated volume to decide if you want to perform the optional operation of compressing or deduplicating existing data on the primary. This is especially true if you use the `–shared-blocks true` or `–snapshot-blocks true` options because running compression against existing data with these options can result in a large number of physical-level changes on disk. If you are deduplicating existing data, doing so can result in a large number of logical changes on disk. For both compression and deduplication of existing data, SnapMirror recognizes these as changed blocks and includes them in the next data transfer to the destination volume. As a result, volume SnapMirror updates that occur after deduplication or compression of existing data with the `–shared-blocks true` or `–snapshot-blocks true` options are likely much larger than normal. For preexisting volume SnapMirror relationships, take into consideration the big surge of data involved in the transfer after compression or deduplication of existing data and plan accordingly.

Data compaction:

- SnapMirror operates at a logical level and does not have any effect on compacted data. During SnapMirror transfer, the source reads the packed 4K block containing compressed sub-4K chunks and sends the compressed sub-4K chunks to destination as is. The destination packs the precompressed sub-4K chunks and writes them to storage. The compaction savings on the destination might not be exactly the same as those of the source, but they are comparable to those of the source. The difference in savings is because the sub-4K chunks are repacked at the destination, and the same sub-4K chunks that were part of a single 4K WAFL block on the source might not end up being together in a single 4K WAFL block on the destination.
- Regardless of whether the source system supports data compaction, the data on the destination is compacted on a data compaction supported destination system.
- If the destination system does not support data compaction, the sub-4K chunks are written unpacked to storage.

Backup: SnapVault

Both compression and deduplication are supported with SnapVault. The behavior of compression and deduplication with SnapVault is similar to that of SnapMirror and depends on the efficiency settings on the source and destination volumes.

If compression is enabled on the SnapVault destination, then the savings from the source are not retained over the network transfer, but they can be regained.

If the source and destination volumes have a different compression type (source volume: adaptive compression; destination volume: secondary compression; or vice versa), then the savings from the source are not retained over the network transfer. Depending on whether the destination has inline or postprocess compression, the savings are regained.

All other configurations retain the savings over the network and on the destination.

As a best practice, if deduplication is enabled on the source, enable only deduplication on the SnapVault destination if you expect SnapVault updates to occur from Snapshot copies created before deduplication completes on the source.

As a best practice, enable only compression on the SnapVault destination if you cannot run compression on the source.

If you have compression or deduplication enabled on the destination, the process starts automatically after the transfer completes. You can't change when this process runs; however, you can change the volume efficiency priority that is assigned to the volume.

Here are some recommendations for SnapVault destinations when the source has compression enabled:

- If you require compression savings on the destination and your source has compression enabled, then do not enable compression on the SnapVault destination. The savings are already inherited on the destination. Second, if you enable compression on the SnapVault destination, then the savings are lost during the transfer, and you have to redo the savings on the destination. Further, if you ever enable compression on the destination, even if you later disable it, you never retain the savings from the source.
- Postprocess compression of existing data results in physical-level changes to the data. This result means that SnapVault recognizes the changes as changed blocks and includes them in its data transfers to the destination volume. As a result, SnapVault transfers that follow a `volume efficiency start –scan-old-data true` command are likely to be much larger than normal. If you can do so, NetApp recommends compressing existing data on the source before running baseline transfers for SnapVault. For preexisting SnapVault relationships, take into consideration the big surge of data involved in the transfer and plan accordingly.
- As a best practice, have the same compression type on the SnapVault source and destination to retain savings over the network.

Here are some recommendations for SnapVault destinations when the source does not have deduplication or compression enabled but the destination does:

- If you require compression savings on the destination and your source does not have compression enabled, then NetApp recommends using inline compression if you have sufficient CPU resources during the backup. Inline compression provides the maximum amount of savings without having a temporary impact on the Snapshot copy size. If you use postprocess compression without inline compression, then the size of the archival Snapshot copy is temporarily larger because it contains the uncompressed version of the blocks just transferred. This temporary increase in Snapshot space is removed after the deduplication process completes and a newer archival Snapshot copy is created. After you enable compression on the destination, even if you later disable it and enable it on the source, you never retain the savings from the source.

- After a SnapVault transfer completes, it automatically creates an archival Snapshot copy. If you run deduplication or compression on the destination, the archival Snapshot copy is replaced with a new one after compression and deduplication run on the destination. (The name of this new Snapshot copy is the same as that of the archival copy, but the creation time of this copy is changed.)
- The compression and deduplication schedule on the destination is tied to the SnapVault schedule and cannot be modified. The compression and deduplication schedule on the source is not tied to the SnapVault update schedule, and it can be configured independently on a volume.
- By default, every SnapVault update (baseline or incremental) kicks off the compression and deduplication process on the destination after the archival Snapshot copy is created if compression and deduplication are enabled. This feature can be modified by setting the postprocess compression and deduplication schedule to manual. When this feature is modified, deduplication metadata is not created, and postprocess compression and deduplication processes are stopped from running. This option is beneficial for SnapVault destinations that have data with good compression savings but minimal deduplication savings.
- For maximum savings, set the `incompressible data detection` option to false on a SnapVault destination if the volume efficiency schedule is set to manual.
- You cannot modify the compression and deduplication schedule on a SnapVault destination to run at a specific time or after a specific amount of data has been written to the volume. The `volume efficiency start` command is not allowed; however, the `volume efficiency start –scan-old-data true` command can be run manually on the destination.
- The SnapVault update is not tied to the compression and deduplication operation. A subsequent incremental update can run even if the compression and deduplication process on the destination volume from the previous backup is not complete. In this case, the compression and deduplication process creates a checkpoint and stops running during the SnapVault update. The process resumes after the SnapVault update is complete, although the archival Snapshot copy is not replaced. This operation can result in the active file system temporarily having fewer savings and the Snapshot copy taking more space until the Snapshot copy expires.
- You can manage deduplication and compression on SnapVault destinations using either the CLI or OnCommand products. For additional SnapVault information, refer to [TR-3487: SnapVault Design and Implementation Guide].

Data compaction:

- Similar to SnapMirror, SnapVault does not have any effect on compacted data. The network transfer is precompressed sub-4K chunks to save CPU and network bandwidth. As with SnapMirror, the destination compaction savings might be different from those of the source due to repacking of sub-4K chunks at the destination.
- Regardless of whether the source system supports data compaction, the data on the destination is compacted on the data compaction supported destination system.
- If the destination system does not support data compaction, the sub-4K chunks are written unpacked to storage.

SnapRestore

NetApp SnapRestore® functionality is supported with both compression and deduplication, and it works in the same way with either feature as it does without. When you initiate a SnapRestore operation on a volume, the restored data retains the original space savings.

If Snapshot and the destination volumes being restored have a different compression type, then the restored data does not retain the original compression space savings.

Deduplication metadata is restored to metadata corresponding to the data maintained in the Snapshot copy. This process requires additional processing during the first deduplication operation completed after using volume SnapRestore.

SnapRestore operates at logical level and does not affect data compaction.

11.2 Other NetApp Features

Deduplication, Compression, Data Compaction, and Zero-Block Deduplication Interoperability

Deduplication, compression, data compaction, and inline zero-block deduplication work seamlessly with one another. If a system has all three features enabled, the order of execution is as follows:

1. Inline zero-block deduplication
2. Inline compression
3. Inline deduplication (Data ONTAP 8.3.2 on)
4. Inline data compaction (ONTAP 9 and AFF systems only)
5. Postprocess compression
6. Postprocess deduplication

NetApp internal tests have shown that the order of execution shows results in optimal savings with the lowest performance and metadata processing overhead.

7-Mode Transition Tool (7MTT)

The 7-Mode Transition Tool preserves dedupe and compression settings when copying volumes from Data ONTAP operating in 7-Mode to clustered Data ONTAP.

To use compression, the source and the destination volumes must belong to a 64-bit aggregate. All compression and deduplication savings on the source volume are retained over the network during transition. After transition, the destination volume inherits all of the compression and deduplication attributes and storage savings from the source volume.

Transitioning deduplicated and compressed data helps to reduce the network bandwidth during transition because of these reasons:

- Shared blocks are transferred only once.
- Compression is maintained throughout the transfer.
- Compressed and deduplicated data has smaller transfer sizes as a result of compression and deduplication space savings; therefore, the transfers are completed more quickly.

You should not start compression or deduplication of existing data on the source volume during transition. If deduplication or compression is in progress, start the transition only after the deduplication or compression operation completes. Doing so avoids sending undeduplicated or uncompressed data and additional temporary metadata files over the network to the destination volume.

For deduplication and compression to take effect on any new data written on the clustered Data ONTAP volume, you must enable deduplication and compression schedules after the transition.

Deduplication maintains a partially ordered fingerprint database in the volume along with the aggregate copy. As a result, the destination system has the space savings from the source volume as well as a copy of the ordered fingerprint database. After migration, when volume efficiency is run on the new volume for the first time, the aggregate fingerprint database is automatically constructed from the copy in the destination volume. This process can result in a one-time increase in the time it takes for volume efficiency operations to complete.

The same compression type (secondary compression) is maintained for the volume after transition to clustered Data ONTAP.

Following are compression considerations when moving from 7-Mode to clustered Data ONTAP 8.3.1 or later using 7MTT.

Scenario 1: Transition from 7-Mode System to Clustered Data ONTAP FAS System

- Case 1: Compression is not enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, compression is not enabled on the volume.
  - Users can enable either inline or postprocess adaptive or secondary compression on the volume.
- Case 2: Inline secondary compression is enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, inline secondary compression is enabled on the volume.
  - Users can enable postprocess secondary compression on the volume.
  - If users want to enable adaptive compression (inline or postprocess), they must rehydrate the volume before changing the compression type.
- Case 3: Postprocess secondary compression is enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, postprocess secondary compression is enabled on the volume.
  - Users can enable inline secondary compression on the volume.
  - If users want to enable adaptive compression (inline or postprocess), they must rehydrate the volume before changing the compression type.
- Case 4: Both inline and postprocess secondary compression is enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, both inline and postprocess secondary compression is enabled on the volume.
  - If users want to enable adaptive compression (inline or postprocess), they must rehydrate the volume before changing the compression type.

Scenario 2: Transition from 7-Mode System to Clustered Data ONTAP AFF System

Note: Postprocess compression is not supported on AFF systems.

- Case 1: Compression is not enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, compression is not enabled on the volume.
  - Users can enable either inline adaptive or inline secondary compression on the volume. Only new writes to the volume are compressed using the selected compression type.
  - In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all of the old data is overwritten.
  - Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
- Case 2: Inline secondary compression is enabled on the 7-Mode volume:

- After transition to clustered Data ONTAP, inline secondary compression is enabled on the volume.
- If users want to enable inline adaptive compression, they must rehydrate the data before changing the compression type. Only new writes to the volume are compressed using the selected compression type.
- In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
- Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

- Case 3: Postprocess secondary compression is enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, postprocess secondary compression is disabled, but the existing compressed data stays compressed in a secondary compression format.
  - Users can enable inline secondary compression on the volume.
  - If users want to enable inline adaptive compression, they must rehydrate the data before changing the compression type. Only new writes to the volume are compressed using the selected compression type.
  - In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
  - Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
- Case 4: Both inline and postprocess secondary compression is enabled on the 7-Mode volume:
  - After transition to clustered Data ONTAP, only inline secondary compression is enabled on the volume. Postprocess compression is disabled on the volume, but the existing compressed data stays compressed in a secondary compression format.
  - If users want to enable inline adaptive compression, they must rehydrate the data before changing the compression type. Only new writes to the volume are compressed using the selected compression type.
  - In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
  - Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Data compaction:

When transitioning data to ONTAP 9 using 7MTT, data is compacted during transition.

Volume Move

Because NetApp supports mixed AFF-FAS clusters, we list compression points for you to consider when moving volumes within a mixed cluster.

- Between AFF nodes and HA pairs:
  - Both AFF nodes and HA pairs running a Data ONTAP 8.3.1 or later release:
    - No change after the move: all volume configurations and savings are retained.
    - All of the AFF compression features in Data ONTAP 8.3.1 and later are supported on the volume.
  - Both AFF nodes and HA pairs running a pre-8.3.1 Data ONTAP release:
    - No change after the move: all volume configurations and savings are retained.
    - All pre-8.3.1 AFF compression features are supported on the volume.
  - Source AFF nodes and HA pairs running the Data ONTAP 8.3.1 or later release and destination AFF node and HA pairs running a pre-8.3.1 release:
    - If the source volume does not have compression enabled, there is no change after the move. All pre-8.3.1 AFF compression features are supported on the volume following the move.
    - If the source volume has inline secondary compression enabled, there is no change after the move. Inline secondary compression continues to be enabled, and all pre-8.3.1 AFF compression features are supported on the volume following the move.
    - A mixed-version cluster does not support adaptive compression volumes.
  - Source node running a pre-8.3.1 Data ONTAP release and destination node running a Data ONTAP 8.3.1 or later release:
    - Refer to Table 8 for the behavior.

Table 8) Compression behavior after volume move from pre-8.3.1 AFF node to 8.3.1+ AFF node.

Before the Move	After the Move
Pre-8.3.1 AFF Node	8.3.1 or Later AFF Node
Volume 1: no compression	Volume 1: no compression: • You can enable inline and secondary compression on Volume 1. • A mixed-version cluster does not support adaptive compression. After all the nodes in the cluster are upgraded to 8.3.1 or later, you can enable adaptive compression. • Existing data on the volume remains uncompressed. • In Data ONTAP 8.3.1, you cannot explicitly convert the existing data to a compressed format. Existing data is converted to a compressed format when the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users must manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 2: inline secondary compression	Volume 2: inline secondary compression: • Existing compressed data remains compressed in a secondary compression format. • If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Before the Move	After the Move
Volume 3: inline secondary compression + background secondary compression	Volume 3: inline secondary compression: • Background compression is disabled because it is not supported on AFF configurations. • If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 4: background secondary compression	Volume 4: no compression: • Background compression is disabled on the volume because it is not supported on AFF configurations. • Existing compressed data remains compressed in a secondary compression format. • If you want, you can enable inline secondary compression. • If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

- Between FAS nodes:
  - Both FAS nodes running a Data ONTAP 8.3.1 or later release:
    - No change after the move: all volume configurations and savings are retained.
    - All Data ONTAP 8.3.1 and later FAS compression features are supported on the volume.
  - Both FAS nodes running a pre-8.3.1 Data ONTAP release:
    - No change after the move: all volume configurations and savings are retained.

- All pre-8.3.1 FAS compression features are supported on the volume.

- Source FAS node running a Data ONTAP 8.3.1 or later release and destination FAS node running a pre-8.3.1 release:
  - If the volume does not have compression enabled, there is no change after the move. All pre-8.3.1 FAS compression features are supported on the volume.
  - If the volume has inline and/or secondary compression enabled, there is no change after the move. Inline and/or secondary compression continues to be enabled, and all pre-8.3.1 compression features are supported on the volume.
  - A mixed-version cluster does not support adaptive compression volumes.
- Source FAS node running a pre-8.3.1 Data ONTAP release and destination FAS node running a Data ONTAP 8.3.1 or later release:
  - Refer to Table 9 for the behavior.

Table 9) Compression behavior after volume move from pre-8.3.1 FAS node to 8.3.1+ FAS node.

Before the Move	After the Move
Pre-8.3.1 FAS Node	8.3.1 or Later FAS Node
Volume 1: no compression	Volume 1: no compression: • You can enable either inline and/or postprocess secondary compression on Volume 1. • A mixed-version cluster does not support adaptive compression. After all the nodes in the cluster are upgraded to 8.3.1 or later, you can enable adaptive compression.
Volume 2: inline secondary compression	Volume 2: inline secondary compression: • Existing compressed data remains compressed in a secondary compression format. • If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression.
Volume 3: inline secondary compression + background secondary compression	Volume 3: inline secondary compression + background secondary compression: • If you want to enable inline or postprocess adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume.
Volume 4: background secondary compression	Volume 4: background secondary compression: • You can enable inline secondary compression. • If you want to enable inline or postprocess adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume.

- Between FAS and AFF nodes:
  - Both FAS node and AFF nodes running a Data ONTAP 8.3.1 or later release:
    - Refer to Table 10 for the behavior.

Table 10) Compression behavior after volume move from 8.3.1+ FAS node to 8.3.1+ AFF node.

Before the Move	After the Move
8.3.1 or Later FAS Node	8.3.1 or Later AFF Node
Volume 1: no compression	Volume 1: no compression: • You can enable inline adaptive or inline secondary compression on Volume 1. • Existing data on the volume remains uncompressed. • In Data ONTAP 8.3.1, you cannot explicitly convert the existing data to a compressed format. Existing data is converted to a compressed format when the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 2: inline secondary compression	Volume 2: inline secondary compression: • Existing compressed data remains compressed in a secondary compression format. • If you want to enable inline adaptive compression, first rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Before the Move	After the Move
Volume 3: inline secondary compression + background secondary compression	Volume 3: inline secondary compression: • Background compression is disabled because it is not supported on AFF configurations. • If you want to enable inline adaptive compression, first rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 4: background secondary compression	Volume 4: no compression: • Background compression is disabled on the volume because it is not supported on AFF configurations. • Existing compressed data remains compressed in a secondary compression format. • If you want, you can enable inline secondary compression. • If you want to enable inline adaptive compression, first rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. Background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Before the Move	After the Move
Volume 5: inline adaptive compression	Volume 5: inline adaptive compression: • Existing compressed data remains compressed in an adaptive compression format. • If you want to enable inline secondary compression, first rehydrate the volume. Only new writes to the volume are compressed using secondary compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.
Volume 6: background adaptive compression	Volume 6: no compression: • Background compression is disabled on the volume because it is not supported on AFF configurations. • Existing compressed data remains compressed in an adaptive compression format. • If you want, you can enable inline adaptive compression. • If you want to enable inline secondary compression, first rehydrate the volume. Only new writes to the volume are compressed using secondary compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

Before the Move	After the Move
Volume 7: inline adaptive compression + background adaptive compression	Volume 7: inline adaptive compression: • Background compression is disabled because it is not supported on AFF configurations. • If you want to enable inline secondary compression, first rehydrate the volume. Only new writes to the volume are compressed using adaptive compression. • In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten. • Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the `background scanner` command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.

- Both FAS node and AFF node running a pre-8.3.1 Data ONTAP release:
  - All volume configurations and savings are retained.
  - All pre-8.3.1 AFF compression features are supported on the volume.
- Source FAS node running a Data ONTAP 8.3.1 or later release and destination AFF node running a pre-8.3.1 Data ONTAP release:
  - If the volume does not have compression enabled, there is no change after the move. All pre-8.3.1 AFF compression features are supported on the volume.
  - If the volume had inline and/or postprocess secondary compression enabled, these processes continue to be enabled after the move. All pre-8.3.1 AFF compression features are supported on the volume.
  - A mixed-version cluster does not support adaptive compression volumes.
- Source FAS node running a pre-8.3.1 Data ONTAP release and destination AFF node running a Data ONTAP 8.3.1 or later release:
  - Refer to Table 11 for the behavior.

Table 11) Compression behavior after volume move from pre-8.3.1 FAS node to 8.3.1+ AFF node.

<table>
<tr><th>Before the Move</th><th>After the Move</th></tr>
<tr><th>Pre-8.3.1 FAS Node</th><th>8.3.1 or Later AFF Node</th></tr>
<tr><td>Volume 1: no compression</td><td>Volume 1: no compression:
You can enable inline secondary compression on Volume 1.
A mixed-version cluster does not support adaptive compression. After all the nodes in the cluster are upgraded to 8.3.1 or later, you can enable adaptive compression.
Existing data on the volume remains uncompressed.
In Data ONTAP 8.3.1, you cannot explicitly convert the existing data to a compressed format. Existing data is converted to a compressed format when the old data is overwritten.
Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the <code>background scanner</code> command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.</td></tr>
<tr><td>Volume 2: inline secondary compression</td><td>Volume 2: inline secondary compression:
Existing compressed data remains compressed in a secondary compression format.
If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression.
In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the <code>background scanner</code> command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.</td></tr>
</table>

<table>
<tr><th>Before the Move</th><th>After the Move</th></tr>
<tr><td>Volume 3: inline secondary compression + background secondary compression</td><td>Volume 3: inline secondary compression:
Background compression is disabled because it is not supported on AFF configurations.
If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression.
In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
Starting in Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the <code>background scanner</code> command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.</td></tr>
<tr><td>Volume 4: background secondary compression</td><td>Volume 4: no compression:
Background compression is disabled on the volume because it is not supported on AFF configurations.
Existing compressed data remains compressed in a secondary compression format.
If you want, you can enable inline secondary compression.
If you want to enable inline adaptive compression, first upgrade all the nodes in the cluster to 8.3.1 or later and rehydrate the volume. Only new writes to the volume are compressed using adaptive compression.
In Data ONTAP 8.3.1, the existing uncompressed data in the volume stays uncompressed until all the old data is overwritten.
Starting with Data ONTAP 8.3.2, existing uncompressed data can be compressed using the background scanner. Users have to manually run the <code>background scanner</code> command to initiate the process. The background scanner works similarly to postprocess compression, but scanning is done only once to convert the existing uncompressed data to a compressed format. The background scanner compresses the existing data using the compression type assigned to the volume.</td></tr>
</table>

- Between AFF and FAS nodes:
  - Both AFF nodes and FAS nodes running a Data ONTAP 8.3.1 or later release:
    - No change after the move: all volume configurations and savings are retained.
    - All Data ONTAP 8.3.1 and later FAS compression features are supported on the volume.
  - Both AFF nodes and FAS nodes running a pre-8.3.1 Data ONTAP release:
    - No change after the move: all volume configurations and savings are retained.

        - All pre-8.3.1 Data ONTAP 8.3.1 FAS compression features are supported on the volume.
    - Source AFF node running a Data ONTAP 8.3.1 or later release and destination FAS node running a pre-8.3.1 release:
        - If the volume does not have compression enabled, there is no change after the move. All pre-8.3.1 FAS compression features are supported on the volume.
        - If the volume had inline secondary compression enabled, inline secondary compression continues to be enabled after the move. All pre-8.3.1 AFF compression features are supported on the volume.
        - A mixed-version cluster does not support adaptive compression volumes.
    - Source AFF node running a pre-8.3.1 Data ONTAP release and destination FAS node running a Data ONTAP 8.3.1 or later release:
        - No change after the move: all volume configuration and savings are retained.
        - All pre-8.3.1 Data ONTAP FAS compression features are supported on the volume.

Data compaction:

- Data compaction and volume move operate at a physical level, and volume move uses block replication engine (BRE) to transfer data from source and destination.
- Similar to SnapMirror and SnapVault, during volume move, precompressed sub-4K chunks are sent to the destination, where they are repacked on a compaction-supported destination volume or written as is on a compaction-unsupported destination volume.

Infinite Volume

Both compression and deduplication are supported on Infinite Volumes. Compression and deduplication operate the same way on Infinite Volumes as they do on FlexVol volumes. Here are some important points to consider:

- Compression and deduplication are enabled at the Infinite Volume level. If you want to enable either deduplication or data compression, the process is enabled on the entire Infinite Volume; it cannot be enabled on some data constituents and not on others.
- When deduplication is run, the process looks for duplicate blocks within a single data constituent, not across the entire Infinite Volume. Blocks that exist in different data constituents are not deduplicated against one another. For example, if you have four data constituents, each with four copies of the same block, after deduplication completes on the Infinite Volume, each data constituent stores one physical copy of the duplicate blocks.
- When postprocess deduplication or compression operations are started on the Infinite Volume, separate operations are run on each of the data constituents up to the maximum of eight per node. If more than the maximum number of allowed compression and deduplication operations are scheduled to run at any one time, the operations are queued and run as each deduplication operation completes.
- As with FlexVol volumes, a data constituent requires free space in the volume and in the aggregate for deduplication metadata. See the Deduplication Metadata section for more details.
- The namespace constituent is not deduplicated or compressed.
- Some commands do not operate as they do on FlexVol volumes. These include the next two bullet points.
- You can choose to view aggregated space savings for an Infinite Volume, or you can see savings at the individual data constituent level. To see savings at the data constituent level, run the `volume`

`show` command, substituting the `-volume <volname>` option with `-is-constituent true`. For example, `volume show -vserver <vserver_name> -is-constituent true.`

- The `volume efficiency show` command does not currently display information about status or progress for an Infinite Volume. To see the status or progress of individual data constituents, substitute the `-volume` option for the `-is-constituent true` option. For example, `volume efficiency show -vserver <SVMname> -is-constituent true -fields progress.`

Data compaction operate the same way on Infinite Volumes as it does on FlexVol volumes.

HA Controller Configuration

HA pair controller configurations are supported with both deduplication and compression. NetApp recommends that both nodes run the same version of Data ONTAP.

Deduplication and compression run on each node independently.

Writes to the FlexVol volume or data constituents have fingerprints written to the change log. During failover or giveback to the partner node, inline compression continues as normal and deduplication change logging continues to happen. Postprocess compression and deduplication processes start at the next scheduled time on the failed node after failover or giveback completes using the updated change log, or they can be started manually. A maximum of eight concurrent compression and deduplication processes is allowed on each node on an HA pair configuration. During failover, the surviving node continues to be limited to a maximum of eight concurrent compression and deduplication operations, even though deduplication continues for both nodes.

NetApp recommends that no active postprocess compression and deduplication operations take place during the planned takeover or planned giveback:

- Perform the planned takeover or giveback during a time when compression and deduplication operations are not scheduled to run.
- Determine whether any compression or deduplication operations are active and stop them until the planned takeover or giveback completes.

For additional information about active-active controller configurations, refer to TR-3450: Active-Active Controller Configuration Overview and Best Practice Guidelines.

Data compaction is supported on HA controller configurations.

Quotas

Quotas are based on logical space usage; therefore, for compressed and deduplicated files, the logical (uncompressed and undeduplicated) size is charged against the quotas. There are several advantages to this scheme as opposed to charging quotas based on the physical (compressed and deduplicated) size of the file:

- This is in line with the general design principle of making deduplication and compression transparent to the end user.
- It is easier for system administrators to manage quotas. They can maintain a single quota policy across all volumes, whether or not compression or deduplication is enabled on them.
- Overwriting parts of the file does not fail because of quota errors when the new data being written is not as compressible as the data it is replacing.

Quotas cannot be oversubscribed on a volume. For example, a user with a quota limit of 1TB cannot store more than 1TB of data in a deduplicated and compressed volume. This point is true even if the data has been deduplicated and compressed and fits into less than 1TB of physical space on the storage system.

Because quotas operate at logical level, data compaction does not affect how quotas are reported.

Flash Cache

NetApp Flash Cache™ cards can be used to reduce the number of random disk reads of blocks that are read repeatedly. This reduction can improve the read performance in environments in which a lot of shared blocks are read repeatedly.

The amount of performance improvement deduplication-enabled volumes have with Flash Cache depends on the duplication rate, the access rate, the active dataset size, and the data layout. Adding a Flash Cache card to a system does not increase the deduplication maximum volume size for that system.

Deduplication also enables efficient use of Flash Cache. The process does this by retaining the deduplication savings on Flash Cache that exist on disk. In that way, if you have 32K duplicate blocks on disk, after you run deduplication, only one block is used on disk. If the blocks are randomly accessed, only one block is used in Flash Cache as well. This feature can significantly increase the amount of data that can be stored in Flash Cache.

The Flash Cache card can provide significant performance improvements in VMware VDI environments. The advantages provided by NetApp Flash Cache cards are further enhanced when combined with other shared block technologies, such as NetApp deduplication or NetApp FlexClone technology. For additional information about the NetApp and VMware solution, refer to TR-3705: NetApp and VMware Solution Guide.

A volume on which compression has been run can contain both compressed and uncompressed blocks. How Flash Cache works with blocks depends on whether the blocks are written on disk in a compressed or uncompressed format, regardless of whether compression is enabled or disabled. Flash Cache does not cache compressed blocks. However, the uncompressed blocks on disk and the metadata continue to benefit from the performance improvements of Flash Cache. For additional information about the Flash Cache card, refer to TR-3832: Flash Cache Best Practice Guide.

Data compaction:

- Compacted blocks are treated as any other WAFL block and are cached in Flash Cache cards.

Flash Pool

Both deduplication and compression are supported on Flash Pool aggregates.

In environments with large amounts of shared blocks that are read repeatedly or overwritten randomly, Flash Pool can significantly reduce the number of HDD reads and writes, thus improving performance. Flash Pool does this by retaining the deduplication savings on SSD that exist on HDD. If you have 32K duplicate blocks on disk, after you run deduplication, only one block is used on disk (HDD). If any requests for duplicate blocks are randomly requested, only one block is used in the Flash Pool aggregate (SSD) as well. If data is write cached, it is still evaluated for deduplication the next time deduplication is run. If the data is successfully deduplicated, it remains on SSD until it becomes cold and is ejected as normal. The amount of performance improvement with Flash Pool depends on the amount of shared blocks, the access rate, the active dataset size, and the data layout.

Starting with Data ONTAP 8.3.1, compressed blocks are eligible for both read and write caching. Any data that is sequentially written continues to be written to HDD regardless of whether compression is enabled or not.

For more information about Flash Pool, refer to TR-4070: NetApp Flash Pool Design and Implementation Guide.

Data compaction:

- Only blocks written to the HDD tier in Flash Pool are compacted. Blocks written to the SSD tier are not compacted.
- The system supports reading compacted blocks from the SSD tier.

FlexClone Volumes

FlexClone volumes support both compression and deduplication. When a FlexClone volume (cloned volume) is created:

If the parent FlexClone volume has compression or duplication enabled, the new volume inherits the compression and deduplication savings and the configuration of the parent volume, such as the compression and deduplication schedule.

The deduplication process also continues for any new data written to the clone. It also recreates the fingerprint database in the aggregate from the volume copy of the deduplication metadata files (the fingerprint database and the change log files). Doing so allows the deduplication process to continue to deduplicate between the new data and the old data.

If you compress existing data with the `-shared-blocks true` or `–snapshot-blocks true` options on the clone, then all compressible data on the parent is written as new data on the clone. This operation can significantly increase the size of the clone and thus reduce the space saving benefits of cloning.

One use case for FlexClone volumes with compression or deduplication is to quickly set up an environment with compression or deduplication to determine the possible deduplication and compression savings on a volume. This feature is useful in the following three scenarios when you do not want to make changes to the production environment:

- When you have a parent volume that does not have deduplication enabled
- When you have a parent volume that does not have deduplication or compression enabled
- When you have a parent volume that has deduplication but you want to test adding compression

Data compaction:

- FlexClone volumes operate at the logical level and are unaffected by data compaction.

FlexClone Files

FlexClone volumes at the file and LUN levels are available and are allowed on compressed and/or deduplicated volumes. Both compression and deduplication can be used to regain capacity savings after a FlexClone volume is broken.

Starting with Data ONTAP 8.3.1, subfile cloning is supported with NetApp data compression for adaptive compression–enabled volumes.

Data compaction:

- FlexClone volumes operate at the logical level and are unaffected by data compaction.

Volume Splitting

When a cloned volume is split from the parent volume, all of the deduplicated data in the clone that was part of the parent volume (not including the data that was written to the cloned volume after the clone was created) is undeduplicated after the volume split operation. If deduplication is enabled on the cloned volume, this data gets deduplicated again in subsequent deduplication operations on the clone volume. If compression is enabled on the cloned volume, then, during a split, the data from the parent is transferred to the clone as uncompressed. The data is recompressed on the clone. Both inline deduplication and inline compression operations are not attempted during volume splitting.

Data compaction: Volume splitting is unaffected by data compaction.

NDMP

Both deduplication and compression support backup to a tape through NDMP. The data sent from the source volume and written to the tape is in an uncompressed and undeduplicated format.

Reallocation (realloc)

Deduplicated data can be reallocated using physical reallocation or read_realloc space_optimized. Although data can be shared by multiple files when deduplicated, reallocate uses an intelligent algorithm to reallocate only the data the first time a shared block is encountered. Compressed data is not reallocated by reallocate or read reallocate, and NetApp does not recommend running reallocate on compressed volumes.

LUNs

When using deduplication or compression on a file-based (NFS/CIFS) environment, both deduplication and compression are straightforward and automatic. As data is written, it is compressed inline or postprocess. As duplicate blocks are freed from deduplication, they are marked as available. In both cases, as blocks of free space become available, the NetApp system recognizes these free blocks and makes them available to the volume.

Deduplication and compression on a block-based (FCP/iSCSI) LUN environment are slightly more complicated. This is because of the space guarantees and fractional reservations used by LUNs. For instance, consider a volume that contains a 500GB LUN and the LUN has LUN reserves enabled. The LUN reserve causes the space for the LUN to be reserved when the LUN is created.

Now consider that 500GB of data is written to the LUN. The consumed space is exactly 500GB of the physical disk space. If the data in the LUN is reduced through compression or deduplication, the LUN still reserves the same physical space capacity of 500GB, and the space savings are not apparent to the user.

LUN space guarantees and fractional reserves can be configured so that the use by the NetApp system of the freed blocks changes depending on the configuration. By varying the values of certain parameters, freed blocks can be returned to either the volume free pool or the aggregate free pool.

This section describes four common examples of LUN configurations and the associated compression and deduplication behavior, summarized in Table 12.

Table 12) Summary of LUN configuration examples.

	A (Default)	B	C	D
LUN space guarantee value	Yes	Yes	No	No
Volume fractional reserve value	100	0	0	0
LUN (fractional) overwrite reserved	Yes	Yes	No	No
Volume space guaranteed	Yes	Yes	Yes	No
Compression and deduplication savings are returned to which free pool?	Neither	Neither	Volume free pool	Aggregate free pool

Definitions

- **LUN (fractional) overwrite reserve.** The space that Data ONTAP reserves is available for overwriting blocks in a LUN when the space guarantee = yes. Because this space is reserved, any savings from compression or deduplication are not available for other use.
- **Volume free pool.** This term refers to the free blocks in the parent volume of the LUN. These blocks can be assigned anywhere in the volume as needed.
- **Aggregate free pool.** This term refers to the free blocks in the parent aggregate of the LUN. These blocks can be assigned anywhere in the aggregate as needed.

Configuration A: Default LUN Configuration

The default configuration of a NetApp LUN follows. (The best practice for all NetApp LUNs is for you to turn the controller Snapshot copy off, delete all scheduled Snapshot copies, and set the snap reserve to 0.)

LUN space reservation value = on	Default
Volume fractional reserve value = 100	Default
Volume guarantee = volume	Default
Snap reserve = 0% (default = 5%)	`vol modify -percent-snapshot-space 0`
Autodelete = off	Default
Autosize = off	Default
Try_first = volume_grow	Default

This configuration is used so that overwrites to the LUN never fail, even if the LUN is overwritten entirely. When a LUN is created with LUN space reservation on and volume guarantee set to volume, no apparent compression or deduplication savings are observed by the storage administrator.

Configuration B: LUN Configuration with Volume Free Space Management

If the user wants to remove LUN overwrite reserve space, the user can do so with the following configuration:

LUN space reservation value = on	Default
Volume fractional reserve value = 0	`vol modify -fractional-reserve 0`
Volume guarantee = volume	Default
Snap reserve = 0%	`vol modify -percent-snapshot-space 0`
Autodelete = on	`snapshot autodelete modify –enabled true`
Autosize = on	`vol modify –max-autosize X` –autosize-increment Y,

where X represents the maximum size to which the volume can grow, and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow	Default

The differences between this configuration and configuration A are that the value of the fractional reserve is set to zero, and both autodelete and autosize are on. As a result, in this configuration, no blocks are set aside for LUN overwrites. To mitigate the possibility of overwrite failures caused by a full volume, NetApp recommends turning on Snapshot autodelete and volume autosize. Snapshot autodelete frees space in a volume by automatically deleting old Snapshot copies. Volume autosize expands the volume when it reaches a predefined threshold.

In this configuration, if compression or deduplication was run, there would be no space savings even if a space reclamation tool was run. The reason is that with LUN space reservation on, 100% of the LUN space is allocated at the time of LUN creation whether those blocks contain data or are empty.

Configuration C: LUN Configuration for Maximum Volume Space Savings

If the user wants to make freed blocks available to the volume free pool, the user can do so with the following configuration:

LUN space reservation value = off `lun modify -space-reserve disabled lun_path`

Volume fractional reserve value = 0 `vol modify -fractional-reserve 0`

Volume guarantee = volume Default

Snap reserve = 0% (default = 5%) `vol modify -percent-snapshot-space 0`

Autodelete = on `snapshot autodelete modify –enabled true`

Autosize = on `vol modify –max-autosize X` –autosize-increment Y, where X represents the maximum size to which the volume can grow, and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow Default

The difference between this configuration and configuration B is that the LUN is not space reserved. At the time of creation, the LUN takes up no space from the volume. Only when data is written to the LUN is space allocated. This process allows volumes to support more LUNs than they physically have space for. With LUN space guarantees off, the value for the volume fractional reserve is ignored for all LUNs in this volume. However, because autodelete and autosize are both on, the volume expands rather than runs out of space if the total allocated blocks approach the total size of the volume.

In this configuration, after compression and deduplication are run, savings are not seen unless a space reclamation tool such as NetApp SnapDrive® software is run. If space reclamation is run after compression and deduplication complete, then the freed blocks are available to the volume.

Configuration D: LUN Configuration for Maximum Volume and Aggregate Space Savings

This configuration provides optimal space savings by returning all freed blocks to the aggregate free pool. This process is accomplished with the following configuration:

LUN space reservation value = off `lun modify -space-reserve disabled lun_path`

Volume fractional reserve value = 0 `vol modify -fractional-reserve 0`

Volume guarantee = none `vol modify -space-guarantee none`

Snap reserve = 0% (default = 5%) `vol modify -percent-snapshot-space 0`

Autodelete = on `snapshot autodelete modify –enabled true`

Autosize = on `vol modify –max-autosize X` –autosize-increment Y, where X represents the maximum size to which the volume can grow and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow Default

The difference between this configuration and configuration C is that the volume is not space reserved. This configuration automatically allocates the free blocks from the volume into the aggregate free pool, where the blocks can be reallocated to any other volumes or LUNs in the aggregate.

The advantage of this configuration is that it provides the highest efficiency in aggregate space provisioning. The configuration also uses Data ONTAP thin provisioning features, volume autosize, and Snapshot autodelete to help administer the space in the solution.

In this configuration, any blocks in the volume that are freed by compression and deduplication are automatically allocated to the aggregate free space pool. If space reclamation is performed on the LUN, then any freed blocks from the LUN are also allocated to the aggregate free pool.

For more information about LUN configuration options, see TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide.

Additional Resources

- NetApp Space Savings Estimation Tool v4.0:
- https://fieldportal.netapp.com/content/437497

  These links can be accessed only by NetApp employees and partners. If you are a NetApp customer and would like to have access to the tool, talk to your NetApp partner or account representative.

  We have initiated the process to post the tool to the Tool Chest. After the process is done, partners and customers will have direct access to the tool.

- TR-4514: NetApp AFF8080A EX Storage Efficiency and Performance with Oracle Database

  NetApp AFF8080A EX Storage Efficiency and Performance with Oracle Database

- TR-4516: NetApp AFF8080A EX Storage Efficiency and Performance with Microsoft SQL Server 2014

  NetApp AFF8080A EX Storage Efficiency and Performance with Microsoft SQL Server 2014

- TR-4403: NetApp AFF8080 EX Performance and Server Consolidation with Microsoft SQL Server 2014

  http://www.netapp.com/us/media/tr-4403.pdf

- TR-4415: NetApp AFF8080 EX Performance and Server Consolidation with Oracle Database

  http://www.netapp.com/us/media/tr-4415.pdf

- TR-3633: Oracle Databases on Data ONTAP

  http://www.netapp.com/us/media/tr-4415.pdf

- TR-4428: NetApp All Flash FAS Solution for VMware Horizon 6 and vSphere Virtual Volumes

  http://www.netapp.com/us/media/tr-4428.pdf

- TR-4307: NetApp All Flash FAS Solution for Nonpersistent Desktops with VMware Horizon View

  http://www.netapp.com/us/media/tr-4307.pdf

- TR-4335: NetApp All Flash FAS Solution for Persistent Desktops with VMware Horizon View

  http://www.netapp.com/us/media/tr-4335.pdf

- TR-4037: Introduction to NetApp Infinite Volume
  http://www.netapp.com/us/media/tr-4037.pdf
- TR-4070: NetApp Flash Pool Design and Implementation Guide
  http://www.netapp.com/us/media/tr-4070.pdf
- TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices
  http://www.netapp.com/us/media/tr-3978.pdf
- WP-7022: The NetApp Storage Efficiency Guide
  http://www.netapp.com/us/media/wp-7022-1110.pdf
- TR-3742: Using FlexClone to Clone Files and LUNs
  http://www.netapp.com/us/media/tr-3742.pdf
- TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide
  http://www.netapp.com/us/media/tr-3965.pdf
- TR-3450: High-Availability Pair Controller Configuration Overview and Best Practices
  http://www.netapp.com/us/media/tr-3450.pdf
- TR-3747: Best Practices for File System Alignment in Virtual Environments

http://www.netapp.com/us/media/tr-3747.pdf

- TR-3705: NetApp and VMware VDI Best Practices
  http://www.netapp.com/us/media/tr-3705.pdf
- TR-3770: 2,000-Seat VMware View on NetApp Deployment Guide Using NFS: Cisco Nexus Infrastructure
  http://www.netapp.com/us/media/tr-3770.pdf
- TR-3428: NetApp and VMware Virtual Infrastructure 3 Storage Best Practices
  http://www.netapp.com/us/media/tr-3428.pdf
- TR-3749: NetApp and VMware vSphere Storage Best Practices
  http://www.netapp.com/us/media/tr-3749.pdf
- TR-3886: Understanding and Using vStorage APIs for Array Integration with NetApp Storage
  http://www.netapp.com/us/media/tr-3886.pdf
- TR-3702: NetApp Storage Best Practices for Microsoft Virtualization
  http://www.netapp.com/us/media/tr-3702.pdf
- WP-7053: The 50% Virtualization Guarantee* Program Technical Guide
  http://www.netapp.com/us/media/wp-7053.pdf
- TR-3732: Citrix XenServer and NetApp Storage Best Practices
  http://www.netapp.com/us/media/tr-3732.pdf
- TR-3584: Microsoft Exchange 2007 Disaster Recovery Model Using NetApp Solutions
  http://www.netapp.com/us/media/tr-3584.pdf
- TR-3651: Microsoft Exchange 2007 SP1 Continuous Replication Best Practices Guide
  http://www.netapp.com/us/media/tr-3651.pdf
- TR-3824: Storage Efficiency and Best Practices for Microsoft Exchange Server 2010
  http://www.netapp.com/us/media/tr-3824.pdf
- TR-3979: Oracle on NetApp Clustered Data ONTAP Solution Guide
  http://www.netapp.com/us/media/tr-3979.pdf
- TR-3843: Storage Savings with Domino and NetApp Deduplication
  http://www.netapp.com/us/media/tr-3843.pdf

Contact Us

Let us know how we can improve this technical report.

Contact us at docfeedback@netapp.com.

Include TECHNICAL REPORT 4476 in the subject line.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp®
www.netapp.com