QSAN
Official Document

Dedup and Compression

White Paper

April 2025

Announcement

Copyright

Trademarks

TABLE OF CONTENTS

FIGURES

TABLES

NOTICES

Information contained in this document has been reviewed for accuracy. But it could include typographical errors or technical inaccuracies. Changes are made to the document periodically. These changes will be incorporated in new editions of the publication. QSAN may make improvements or changes in the products. All features, functionality, and product specifications are subject to change without prior notice or obligation. All statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

PREFACE

Technical Support

Do you have any questions or need help trouble-shooting a problem? Please contact QSAN Support, we will reply to you as soon as possible.

- Via the Web: https://www.qsan.com/technical_support
- Via Telephone: +886-2-77206355
- (Service hours: 09:30 - 18:00, Monday - Friday, UTC+8)
- Via Skype Chat, Skype ID: qsan.support
- (Service hours: 09:30 - 02:00, Monday - Friday, UTC+8, Summer time: 09:30 - 01:00)
- Via Email: support@qsan.com

Information, Tip, and Caution

This document uses the following symbols to draw attention to important safety and operational information.

INFORMATION

INFORMATION provides useful knowledge, definition, or terminology for reference.

TIP

TIP provides helpful suggestions for performing tasks more effectively.

CAUTION

CAUTION indicates that failure to take a specified action could result in damage to the system.

1. INTRODUCTION

This is an era of data explosion. Although having petabytes of data for research, AI, and analytics sounds beneficial, excessive data can overload storage and hinder high-performance computing. However, simply deleting data isn’t an option because patterns and sequences in data are essential for analytics. This is where data reduction comes in. It transforms raw data to reduce its size while preserving its essential characteristics.

Data scientists use several techniques for data reduction:

1. **Reorganize:** Clean the data by removing corrupt or duplicate entries, reorganizing data to eliminate unnecessary objects, and identifying non-essential information.
2. **Encoding:** Create a smaller representation of the data by encoding features and patterns. This can be "lossy" (reducing size by losing some fidelity) or "lossless" (keeping the data identical but smaller).

Data reduction is a technique used to decrease the amount of data stored and transmitted by eliminating redundancy information. This process aims to enhance storage efficiency, reduce costs, and improve performance in data handling and analysis. Regardless of the approach, data reduction is a complex process in which engineers and scientists must make tradeoffs between space savings, processing savings, and data fidelity.

Some common types of data reduction are:

1. **Thin Provisioning:** Allocating storage space dynamically as needed, rather than pre-allocating it. Although more computationally intensive, this method reduces inefficiencies like disk fragmentation.
2. **Deduplication:** Removing duplicate data, which can include deleting records that are identical or represent the same information.
3. **Compression:** Applying algorithms to reduce the storage space of data. This can be done when data is moved into storage or to data-at-rest for greater space savings.

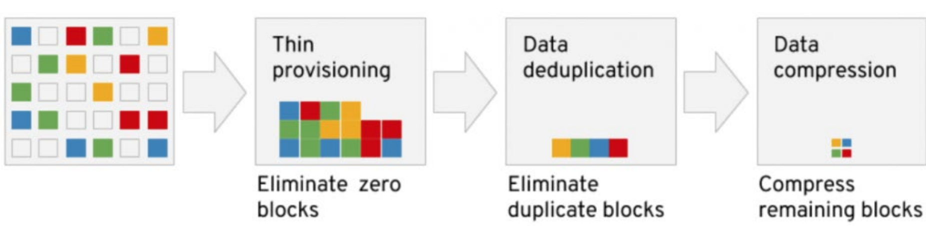

Figure 1-1 Data Reduction Processing (figure from Red Hat blog)

By leveraging these technologies, data reduction can significantly improve the efficiency of storage and network bandwidth utilization. In this document, we will describe data deduplication and compression in depth.

1.1. What Is Data Deduplication

Data deduplication is a data compression technique used to eliminate redundant copies of data, thereby reducing the amount of storage space required. This process ensures that only unique instances of data are stored, while duplicates are replaced with references or pointers to the original data. Deduplication can occur at various levels, such as file-level or block-level, depending on the granularity of the data being analyzed.

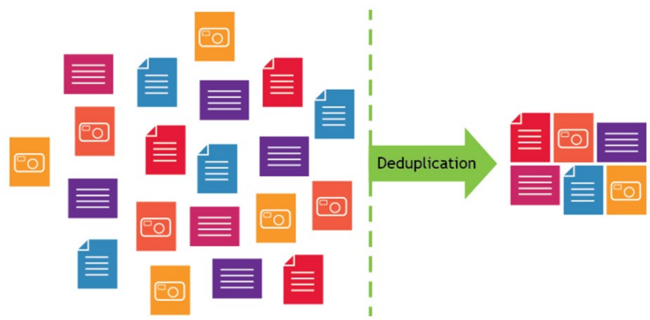

Figure 1-2 Data Deduplication (figure from GK_RAJ post)

There are two levels of deduplication:

1. **File-Level Deduplication:** Compares entire files and removes duplicates. Useful in environments where identical files are frequently saved.
2. **Block-Level Deduplication:** Breaks files into smaller blocks and removes duplicate blocks. More granular and effective than file-level deduplication, especially for large datasets with minor differences.

1.1.1. The Process of Data Deduplication

Data deduplication works by identifying and eliminating redundant copies of data, ensuring that only unique instances are stored. Here's a simplified explanation of how the process typically works:

1. **Data Chunking:** The data is divided into smaller chunks or segments. This can be done using fixed-size chunks or variable-size chunks, depending on the deduplication method.
2. **Hashing:** Each chunk is processed through a hashing algorithm to generate a unique identifier, called a hash value or fingerprint. This identifier represents the content of the chunk.
3. **Comparison:** The hash value of each chunk is compared against a database of existing hash values. If the hash value is not found in the database, the chunk is unique and needs to be stored. If the hash value already exists, it indicates that the chunk is a duplicate.
4. **Storage:** New chunks (with unique hash values) are stored in the storage system. Their hash values are added to the database for future comparisons. Instead of storing duplicate chunks, references (or pointers) to the already stored unique chunks are created.
5. **Metadata Management:** Metadata is maintained to keep track of the references and the structure of the original data. This allows the system to reconstruct the original data when needed.

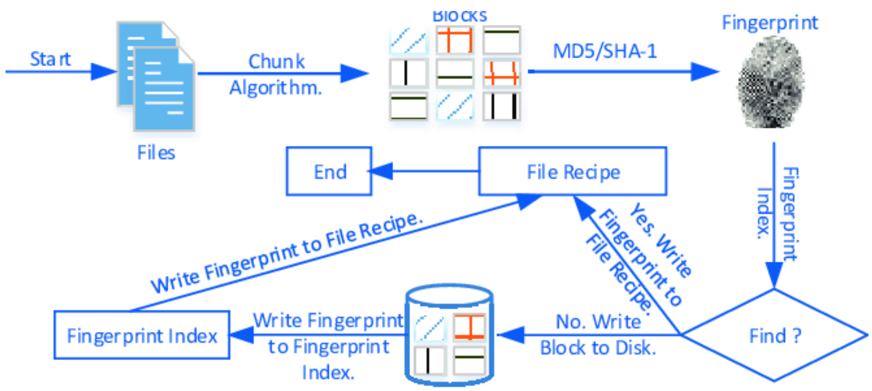

Figure 1-3   The Process of Data Deduplication *(figure from ResearchGate article)*

Data deduplication can be implemented in two primary ways: inline deduplication and post-process deduplication. Each method has its own advantages and use cases, which are determined by the specific requirements of the storage environment and performance needs.

1. **Inline Deduplication:** It occurs during the data write process. As data is being written to the storage system, it is analyzed in real-time, and duplicate data is identified and eliminated before it is stored.
2. **Post-process Deduplication:** It occurs after the data has been written to storage. The data is first written in its original form, and then a separate process scans the data, identifies duplicates, and eliminates them.

Both inline and post-process deduplication offer valuable benefits for reducing storage space, but they cater to different needs. Inline deduplication provides immediate space savings and is beneficial in high-redundancy environments, although it may introduce some write latency. Post-process deduplication avoids impacting write performance but achieves space savings after the initial write, making it suitable for archival and systems where write performance is paramount.

1.1.2. Benefits and Challenges of Deduplication

Listed below are some benefits of using deduplication:

- **Storage Efficiency:** Significantly reduces the amount of storage space required by eliminating redundant data.
- **Cost Savings:** Lowers storage infrastructure costs by maximizing the use of existing storage resources.
- **Improved Backup and Recovery:** Speeds up backup and recovery processes by reducing the amount of data that needs to be transferred and stored.
- **Enhanced Data Management:** Simplifies data management by reducing the volume of data to be handled.

Here are some challenges of using deduplication:

- **Processing Overhead:** Initial deduplication can require significant processing power and time.
- **System Performance:** Continuous deduplication processes can impact system performance if not managed properly.
- **Complexity:** Implementing and maintaining a deduplication system can be complex and may require specialized tools and expertise.

Data deduplication is a powerful technique for optimizing storage efficiency and reducing costs by eliminating redundant copies of data. By storing only unique instances and replacing duplicates with references, deduplication significantly decreases the storage footprint and enhances data management. Despite the challenges of implementation and potential impact on system performance, the benefits of deduplication make it an essential component of modern data storage strategies.

1.2. What Is Compression

Compression in storage is a technique used to reduce the size of data by encoding it more efficiently. This process helps to save storage space, reduce costs, and improve data transfer speeds. Compression can be applied to various types of data, including text, images, audio, and video, and is commonly used in both file storage and data transmission.

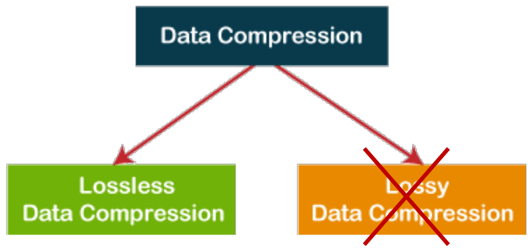

Figure 1-4 Types of Data Compression

There are two types of data compression:

1. **Lossless Compression:** Reduces the size of data without losing any information. The original data can be perfectly reconstructed from the compressed data. It is used for text documents, executable files, and any data where integrity and exact replication are critical. For examples: ZIP files, PNG images, and FLAC audio files.
2. **Lossy Compression:** Reduces the size of data by removing some of the information, which may result in a loss of quality. The original data cannot be perfectly reconstructed. It is used for multimedia files where a reduction in quality is acceptable for significant space savings, such as photos, music, and videos. For examples: JPEG images, MP3 audio files, and MPEG video files. Lossy compression cannot be used in storage because the file cannot be restored to its original form.

1.2.1. The Process of Data Compression

Data compression works by encoding and decoding data, ensuring that lossless data compression compresses without any loss of data quality and that the decompressed file is an exact copy of the original file. Here's a simplified explanation of how the process typically works:

1. **Data Analysis:** Analyze the data to identify patterns, repetitions, and redundancies.
2. **Encoding:** Apply an algorithm to replace repetitive elements with shorter representations. Common algorithms include Huffman coding, LZW (Lempel-Ziv-Welch), and RLE (Run-Length Encoding).

3. **Storage or Transmission:** Store or transmit the compressed data along with a dictionary or key for decompression.

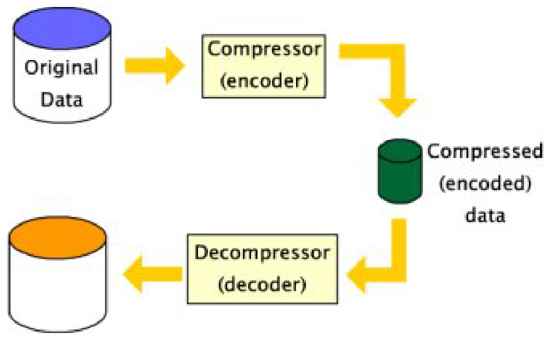

Figure 1-5 The Process of Data Compression (figure from GITTA)

Likewise, data compression can be implemented inline and post-processed. Both inline and post-process compression offer valuable benefits for reducing storage space, but they cater to different needs and scenarios. Inline compression provides immediate space savings and can enhance write performance, though it may introduce latency and require complex implementation. Post-process compression avoids initial write latency and allows for flexible scheduling, but space savings are realized later and require additional temporary storage capacity.

1.2.2. Benefits and Challenges of Compression

Listed below are some benefits of using compression:

- **Storage Efficiency:** Reduces the amount of storage space required for data.
- **Cost Savings:** Lowers storage infrastructure costs and reduces data transfer expenses.
- **Faster Data Transfer:** Compressed data takes less time to transmit over networks, improving performance.
- **Improved Backup and Recovery:** Smaller backup files speed up backup and recovery processes.

Here are some challenges of using compression:

- **Processing Overhead:** Compressing and decompressing data requires additional processing power and time.

- **Complexity:** Implementing and managing compression algorithms can be complex and may require specialized knowledge.

Data compression is a powerful technique for optimizing storage and transmission efficiency. By encoding data more efficiently, compression reduces storage space requirements and improves data transfer speeds.

1.3. Deduplication vs. Compression

In deduplication, the data is clustered based on the common blocks in them. A single version of each block is retained while the other occurrences hashed or referred to using pointers. On the other hand, in compression, additional data, spaces, etc. are eliminated to reduce the data file size. Therefore, deduplication focuses on removing duplicates; compression focuses on reducing size. Deduplication is more effective for backup and archiving; compression is more effective for file transfer and storage.

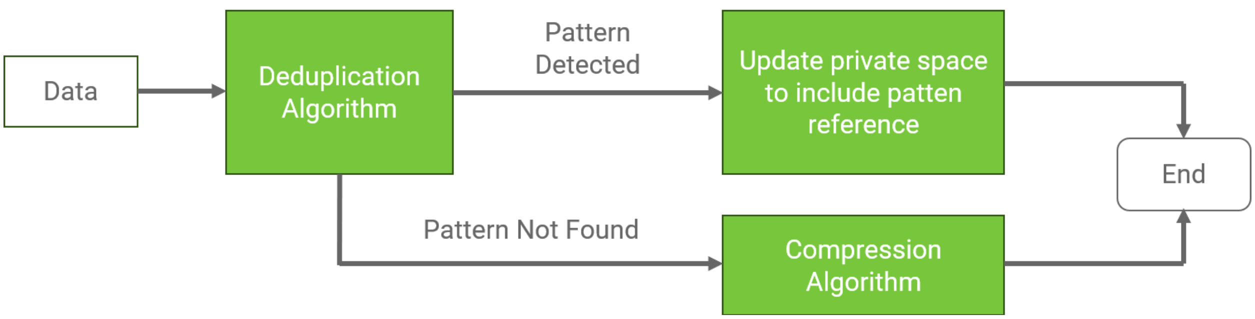

Figure 1-6 Complementary Use of Deduplication and Compression

For size reduction ratios, compression claims to reduce data size to a ratio of 2.5:1, as claimed by some programs based on available data file types. However, through deduplication, the data will change significantly. Reduction ratios can range from 4:1 to 10:1, or even higher for certain material types.

Table 1-1 Deduplication vs. Compression

	DEDUPLICATION	**COMPRESSION**
Definition	Eliminates redundant copies of data	Reduces the size of data by encoding it more efficiently
Method	Identifies and stores unique data chunks, replaces duplicates with references	Encodes data to use fewer bits, using algorithms to represent data more compactly
Efficiency	Highly effective for environments with large amounts of duplicate data	Effective for most types of data, with varying degrees of efficiency
Space Savings	Depends on the amount of duplicate data, reduction ratios can range from 4:1 to 10:1	Consistent space savings, can be significant depending on the data type and algorithm, reduce data size to a ratio of 2.5:1
Processing Overhead	Can be high, especially for inline deduplication	Varies; real-time (inline) compression can introduce latency
Use Cases	• Backup storage • Cloud storage • VDI (Virtual Desktop Infrastructure)	• File storage • Data transmission • Multimedia files
Complementary Use	Can be combined with compression to further optimize storage efficiency	Can be used with deduplication for additional storage reduction benefits

Both deduplication and compression have their own set of advantages and limitations. For complementary use, the two can be used in conjunction for maximum benefit.

1.4. Factors Influencing Types

When setting capacity savings goals for data reduction methods, you must consider the following deterministic factors that affect deduplication and compression efficiency ratios, which are critical to storage performance:

- **Data Type:** It plays a key role in determining the degree of deduplication or compression. Snapshots (such as in VDI environments) are expected to provide higher capacity optimization rates. For databases, there is already a certain level of redundancy elimination at the application level. Therefore, the database may not generate significant savings through deduplication and compression.
- **Data Change Rate:** The less data changes, the easier and faster it is to perform deduplication and compression. Each change made to data and saved to disk requires a separate check cycle to identify duplicate data sets and perform compression algorithms. This subsequently increases server overhead.
- **Data Backup Frequency:** Especially in the case of full backup, the higher the backup frequency, the more redundant data will be generated. This will save more space during deduplication and compression.

The efficiency of the data reduction can be measured as the ratio of original data size to reduced data size. For example, when the deduplication ratio is 10:1, 100 GB of original data only requires 10 GB of storage capacity, thus saving 90% of space. The higher the ratio, the greater the capacity savings. This depends on the ability of software to perform deduplication and compression, as well as the other factors mentioned above.

1.5. Data Reduction Usage Guide

Although data reduction has limited applications, it is still more cost-effective in most cases. According to the SNIA report, the TCO (Total Cost of Ownership) percentage per TB of "Traditional HDDs : NVMe SSDs" is approximately 45.37%. When the application data reduction rate reaches 54.63% or higher, the deduplication strategy will achieve better TCO than deploying a new storage device, as shown in the figure below. Since the application is more suitable for deduplication strategy, the better TCO can be accomplished. In addition, data reduction strategies may result in performance degradation, but the experience of deploying SSDs in a storage environment will still become more powerful.

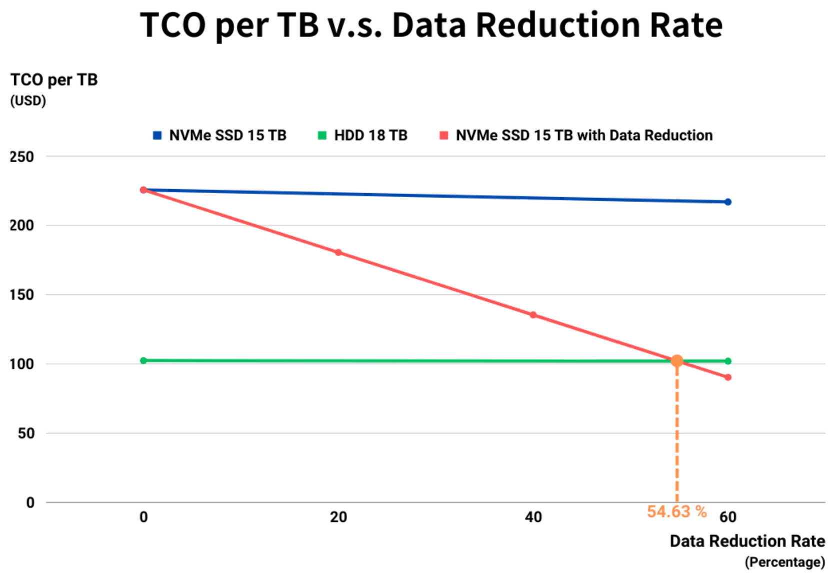

Figure 1-7 TCO per TB vs. Data Reduction Rate

2. THEORY OF OPERATION

The data reduction feature incorporates various space efficiency algorithms to minimize the storage space required for datasets. This feature includes deduplication and compression algorithms. The data reduction works the same for block and file storage volumes. It uses software algorithms to analyze and achieve space savings within storage resources and occurs inline between memory cache and disk drives. This chapter introduces the theory of data reduction operations, breaks down several operations, and describes them individually.

Figure 2-1 Data Reduction Overview

2.1. Write Data

When data is written to the storage, it is first saved in the memory cache. Before storing writes in the memory cache, the storage checks if there is enough space within the target storage resource and allocates space for I/O. To ensure the fastest response, the data reduction algorithm for write I/O is not executed at this stage, and the write is immediately acknowledged to the host.

For storage that support data reduction, the process occurs during active memory cache cleanup or when the memory flushes cache pages to the disk drives. During this process, multiple files or blocks are combined and processed through the algorithm. Based on the algorithm's results, space within the volume is allocated, and data is written to the disk drive if necessary.

2.2. Data Reduction Process

Figure below outlines the data reduction capabilities with deduplication and compression enabled in the volume. It combines three techniques — zero-block elimination, data deduplication, and data compression — to reduce data footprint.

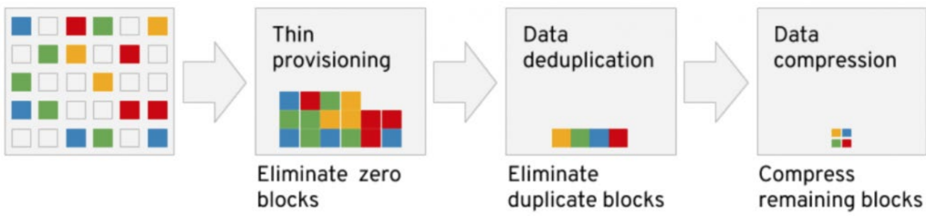

Figure 2-2 Data Reduction Processing (figure from Red Hat blog)

These are critical stages in the data reduction process, allowing the engine to reduce the data footprint on storage. It applies these stages inline and dynamically. Each algorithm in the data reduction function is discussed below.

1. **Zero Block Elimination:** During the initial phase, any block consisting entirely of zeros will be identified and recorded in the metadata only. At this stage, the process only allows data containing non-all zeros to be filtered to the next processing stage.
2. **Deduplication:** In the second phase, incoming data is processed to determine if it is redundant (data that has been previously written). The redundancy of this data is checked through metadata maintained by the deduplication service core module provided as part of the data reduction engine. Any data blocks found to be redundant will not be written out. Instead, the metadata will be updated to point to the original copy of the chunk already stored on the media.
3. **Compression:** After the initial zero elimination and deduplication phases are complete, compression is applied to individual data. The compressed data is then packed into fixed-length chunks and stored on the media. This also speeds up the performance of reading data from storage since a single physical block can contain many compressed blocks.

2.3. Read Data

When a read operation is requested from a storage resource with data reduction enabled, the system first determines where the data is located. It checks if the data is in the memory cache or on the disk drive, and whether it is in its original size or has been deduplicated or compressed. If the data is in the memory cache in its original form, it is sent directly to the host. If the data is in its original form on the disk drive, a standard read operation is performed, copying the data into the memory cache and then sending it to the host.

If the data has been reduced in size, it must be restored to its original form in the memory cache before being sent to the host. For deduplicated data, if they contain a common pattern, they are recreated. The data is then sent to the host. If the data is compressed, it must be uncompressed before being sent to the host. If compressed data is already in the memory cache, it is uncompressed to a temporary location, sent to the host, and the temporary location is cleared. If the compressed data is on a disk drive, it is read into the memory cache, uncompressed to a temporary location, and sent to the host. Data is never uncompressed on the disk itself, as this would negate the storage savings.

2.4. Overwrite Data

When an update is received for a previously written block or file, the system checks if it has space savings and if the update can achieve further savings. The data goes through the data reduction process to determine any potential space savings. If the new block or file matches a known pattern, the private space within the resource is updated with the new pattern information. If the old one had compression savings or was stored in its original form, those space are freed for reuse. If deduplication isn't possible, the data is then sent through the compression logic.

If compression reduces the data size, the system decides where to store the data within the disk drive. If the new data size is larger than before, new space is allocated. If the size hasn't changed or is smaller, the data can be written to the existing allocated block. This approach prevents fragmentation, improving performance and space savings. If a new block is allocated, the old one is freed for reuse. If no deduplication or compression savings are possible, the data is written in its original size, possibly overwriting the original data.

3. CONFIGURE

This chapter will discuss how to setup data reduction.

3.1. Support Firmware Version

The following summarizes the firmware versions and limitations that support data reduction.

Firmware Version

- XEVO
  - Firmware version 3.0.0 and later with **license**
- QSM
  - Firmware version 4.1.4 and later with **license**

3.2. License

The deduplication and compression license must be purchased to enable the feature. There is also a 30-day free trial license for evaluation.

3.2.1. Enable Deduplication and Compression License

The deduplication and compression function is optional. Before using it, you must enable deduplication and compression license. Select the **System** tab and the **Maintenance** subtab, and then click the **Licenses** pane to allow users to active licenses.

Click the **Request License** button to download the file and send to your local sales to obtain a License Key. After getting the license key, click the icon to select it, and then click the **Apply** button to enable. Each license key is unique and dedicated to a specific system. If the license is active, the status will show as **Enable**. After enabling the license, the system must reboot manually to take effect.

3.3. Constraint

There are some caveats regarding the data reduction feature.

- Deduplication and compression require a license.
- Only all-flash pools are supported, even including spare disks.
- Deduplication and compression will be enabled or disabled simultaneously.
- Enable deduplication and compression when creating a pool, and then enable them when creating volumes.
- Support disk roaming.
- After the trial license expires, all volumes with deduplication and compression features will be deactivated.

The total capacity of the pool will be limited depending on the system memory size. The following are the deduplication and compression pool parameters.

Table 3-1 Dedup and Compression Pool Parameters

ITEM	VALUE
The maximum pool capacity per system	System memory x 1,024
The maximum number of pools per system	2
The maximum number of volumes per system	System memory / 4

The following is the mapping table between deduplication and compression pool memory, capacity, and volume number.

Table 3-2 Dedup and Compression Pool Mapping Table

SYSTEM MEMORY PER CONTROLLER	MAXIMUM DECUP & COMPRESSION POOL CAPACITY PER SYSTEM	MAXIMUM NUMBER OF VOLUMES PER SYSTEM
8 GB	8 TB	2
16 GB	16 TB	4
32 GB	32 TB	8
64 GB	64 TB	16
128 GB	128 TB	32
256 GB	256 TB	64
512 GB	512 TB	128
1,024 GB	1,024 TB	256

3.4. Configure Data Reduction

For the operation of deduplication and compression pool, please refer to the XEVO 3 or QSM 4 Software Manual on the QSAN website.

4. TEST RESULTS

This chapter will introduce the test results of data reduction.

4.1. Space Saving Test

This test demonstrates the space savings achieved through data reduction.

Test Equipment and Configuration

- Server
  - Model: ASUS RS700-E6/ERS4 (CPU: Intel Xeon E5620 2.4 GHz / RAM: 24 GB)
    10 GbE HBA: Broadcom BCM57810 NetXtreme II 10 GbE
    OS: Windows Server 2019
- Storage
  - Model: QSAN XN8126D
    Memory: 16 GB per controller
    Firmware Version: 4.1.0
    SSD: 7 x KIOXIA PX05SMB Series, PX05SMB080, 800 GB, SAS 12 Gb/s
  - All-flash Pool: 1 x RAID 6 pool with 7 SSDs in Controller 1
    Volume: 1 x 500 GB with deduplication and compression enabled
- Test Files
  - Large video file: 20 GB
  - Fragmented small files: more than 110K files, each file size between 1 KB and 218 KB, totaling 6.69 GB
  - VMware VMDK files: 21.4 GB total

Test Scenario and Result

1. Create a pool with deduplication and compression enabled, and then create a volume with it also enabled.
2. Write the large video file to the volume and record the occupied capacity as 1st write.
3. Write the same large video file to different directory, and then record the occupied capacity as 2nd write.

4. Repeat steps 2 and 3 for fragmentary small files and VMware VMDK files.

Table 4-1 Space Saving Test Result

TEST ITEM	1ST WRITE	2ND WRITE ORIGINAL / ACTUALLY	SPACE SAVING RATE	REDUCTION RATIO
Large Video File	20 GB	40 GB / 36 GB	10%	1.1 : 1
Fragmentary Small Files	6.69 GB	13.38 GB / 7 GB	49%	5.1 : 1
VMware VMDK Files	21.4 GB	42.8 GB / 29.6 GB	86%	6.9 : 1

4.2. Performance Test

This test demonstrates the impact of data reduction on performance.

Test Equipment and Configuration

- Server
  - Model: ASUS RS700-E6/ERS4 (CPU: Intel Xeon E5620 2.4 GHz / RAM: 24 GB)
    10 GbE HBA: Broadcom BCM57810 NetXtreme II 10 GbE
    OS: Windows Server 2019
- Storage
  - Model: QSAN XN8126D
    Memory: 16 GB per controller
    Firmware Version: 4.1.0
    SSD: 7 x KIOXIA PX05SMB Series, PX05SMB080, 800 GB, SAS 12 Gb/s
  - All-flash Pool: 1 x RAID 6 pool with 7 SSDs in Controller 1
    Volume: 1 x 500 GB with deduplication and compression enabled, another 1 x 500 GB with it disabled
- Test Files
  - Large video file: 20 GB

- Fragmented small files: more than 110K files, each file size between 1 KB and 218 KB, totaling 6.69 GB

Test Scenario and Result

1. Create a pool with deduplication and compression enabled, and then create 2 volumes. One volume has it enabled and the other volume has it disabled.
2. Copy the large video file to these 2 volumes and record the transfer speed.
3. Repeat steps 2 for fragmentary small files.

Table 4-2 Performance Test Result

TEST ITEM	ENABLE	DISABLE	PERFORMANCE IMPACT	TIME SAVING
Large Video File	350 MB/s	750 ~ 800 MB/s	-53% ~ -58%	N/A
Fragmentary Small Files	700 ~ 800 KB/s (takes 2 hours)	1 ~ 1.5 MB/s (takes 1.5 hours)	-20% ~ -53%	25%

5. CONCLUSION

Data reduction is a vital strategy in data management, enabling organizations to handle increasing data volumes more effectively. By implementing techniques like deduplication and compression, organizations can achieve significant cost savings, improve performance, and optimize their storage infrastructure. This holistic approach ensures that data storage remains efficient and manageable in the face of growing data demands.

6. APPENDIX

6.1. Apply To

Deduplication and compression require a **License** to enable the feature. We also provide a 30-day free **Trial License** for evaluation. The deduplication and compression function is applicable to the following models.

- XEVO
  - Firmware version 3.0.0 and later with **license**
- QSM
  - Firmware version 4.1.4 and later with **license**

6.2. Reference

Document

- XEVO 3 Software Manual
- QSM 4 Software Manual