White Paper

EMC VNX DEDUPLICATION AND COMPRESSION

Maximizing effective capacity utilization

Abstract

This white paper discusses the capacity efficiency technologies delivered in the EMC® VNX™ series of storage platforms. High-powered deduplication and compression capabilities for file and block storage are delivered standard with the VNX Operating Environment.

July 2012

EMC²
where information lives®

Part Number h8198.2

Table of Contents

Executive Summary 4
Audience 4
Technology Introduction 4
VNX Data Deduplication and Compression for File 5
Space reduction process 7
Minimizing client impact 8
Client input/output to space-reduced files 10
Deploying VNX Deduplication and Compression for File data 11
Using deduplication settings 11
Enabling deduplication and compression on a file system 13
Viewing the deduplication state 13
Changing the deduplication state 14
Viewing deduplication statistics 16
Compress operations for Block 17
Decompression operations 20
Deploying VNX Compression for Block data 21
Performance 24
Migration to VNX with Deduplication and compression example 25
Customer A 26
Customer B 26
Limits and interoperability 27
Limits 27
Interoperability 28
Special cases for LUN Migration and SAN Copy 29
Resource consumption and performance 30
Conclusion 33
References 33
Appendix A: NTFS file system conditioning with SDelete 34
Appendix B: Compression states 36

Executive Summary

Capacity-optimization technologies play a critical role in today's environment where companies need to do more with less. The EMC® VNX™ series of storage arrays is well-equipped to meet users' needs in this regard. Intelligent and automated deduplication and compression features are provided in the VNX Operating Environment at no additional cost in the VNX5300™ and higher models. (This feature is not available in the smalest model, the VNX5100™).

With VNX deduplication and compression, users can significantly increase storage utilization for file and block data. Often, effective utilizations are increased two to three times compared with traditional storage.

Management is simple and convenient. Once the capacity-optimization technologies are turned on, the system intelligently manages capacity-optimization processes as new data is written. With Unisphere™, users can manage block and file data from within a single screen. In addition, users can deploy many of the features from VMware vCenter™ through the EMC Virtual Storage Integrator for VMware vSphere™: Unified Storage Management feature.

This white paper discusses the capacity-optimization capabilities of VNX series systems, how they are best deployed, and how they fit in with other deduplication technologies in the storage environment.

Audience

This white paper is intended for anyone interested in understanding the deduplication and compression functionality included with the VNX series of storage systems.

Technology Introduction

There are many capacity-optimization technologies in the industry. Each technology varies in its efficacy based on the type of data being processed, amount of data, and data access patterns. Deduplication systems, like the EMC Avamar® and Data Domain® offerings, are designed to process massive amounts of data at high speed. When applied to backup data sets, these systems can reduce required capacity by tens and even hundreds of times the data set's aggregate size. Avamar and Data Domain serve the same basic need—backup to disk—but each implementation provides unique benefits.

VNX systems are high-performing primary-storage devices for file and block data. File data is accessed on the VNX system by using the CIFS, NFS, or FTP protocols. Block data is accessed by using the Fibre Channel (FC), Fibre Channel over Ethernet (FCoE), or internet SCSI (iSCSI) protocols. Capacity optimization on these systems is an asynchronous operation, occurring after new data is written, in an effort to maximize server I/O performance. Avamar and Data Domain offerings allow the use of Data

Domain Boost software to redirect data directly to the Data Domain system. This can significantly increase backup performance by distributing parts of the dedupe process to the backup server. DD Boost will transfer the source data, in an efficient transfer method for processing by Data Domain system in place of performing intensive deduplication processing only on the client. For more information on Avamar and Data Domain, refer to the *"EMC Avamar Integration With EMC Data Domain"* paper on Powerlink.

Table 1. High-level comparison of VNX deduplication and compression with Avamar and Data Domain backup-to-disk systems

VNX – Multipurpose storage platform with storage efficiency features	Avamar and Data Domain – Dedicated backup/archival storage platforms
Post-process data reduction—Device is sized for original data size. Capacity is released gradually as it's processed.	Inline data reduction—Device is sized for reduced data. All incoming data is reduced immediately.
Relatively low to moderate deduplication processing throughput.	Very high deduplication processing throughput.
Low impact, and less aggressive capacity optimization—single instancing of files with compression. Compression for block data.	Most aggressive deduplication—variable block.
Capacity optimization is a low-priority task	Deduplication is a high-priority task.

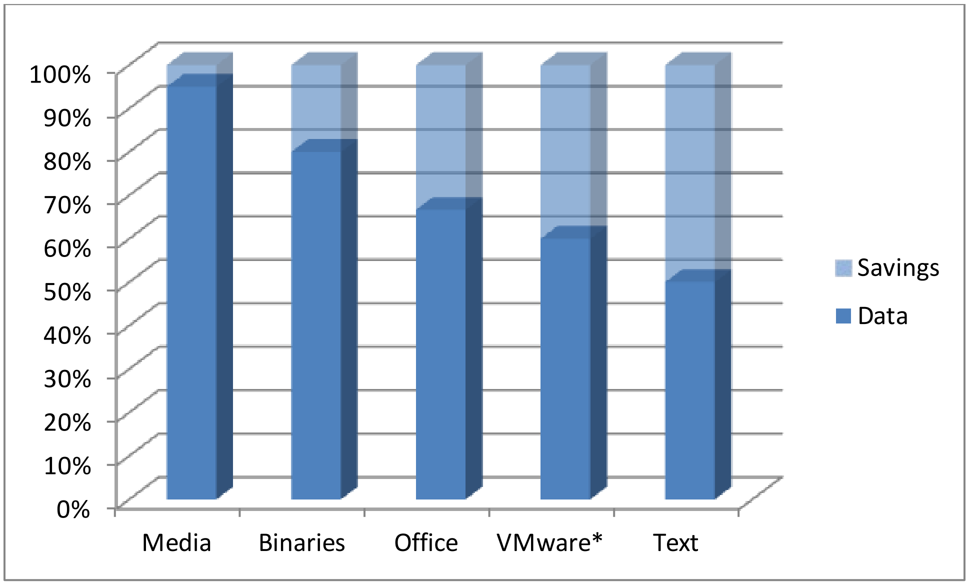
VNX Data Deduplication and Compression for File

VNX systems can increase capacity efficiency by as much as three times when compared to traditional systems without advanced capacity efficiency features (shown in the figures below). VNX achieves this through a combination of capacity efficiency technologies including thin-LUN Virtual Provisioning™, compression, and file-level single instancing. All deduplication and compression features discussed in this paper are available on the VNX5300 and higher models (deduplication currently available on File data only).

VNX systems are built to handle the I/O demands of large numbers of Flash drives. Performance-optimization features such as FAST Cache and FAST_VP move the busiest data onto the highest-performing drives, increasing the system's IOPS-per-dollar figure. However, Flash and high-speed SAS drives have a high cost per gigabyte. The selected use of capacity efficiency features such as deduplication and compression plays a complementary role in lowering overall cost by increasing effective utilization rates.

VNX achieves capacity optimization in slightly different manners for file and block data. Compression is just one element of the VNX capacity-optimization features for both file and block data. Compression is a fundamental capacity efficiency technique used in many solutions because it benefits most data types.

The efficiency benefit of VNX compression for several data types is shown in Figure 1.



*Virtual machines' OS image disks without data. Virtual disks used for data will be as compressible as the data stored on them.

Figure 1. Compression rates of common file types

When migrating from traditional systems to those utilizing capacity efficiency technologies, the initial capacity savings can be much larger than the nominal data-compression rate alone. This is due to the other optimizations used: single instancing of file data and thin-LUN Virtual Provisioning for block data. Figure 2 shows the efficiency of VNX capacity-optimized volumes over RAID Group LUNs.

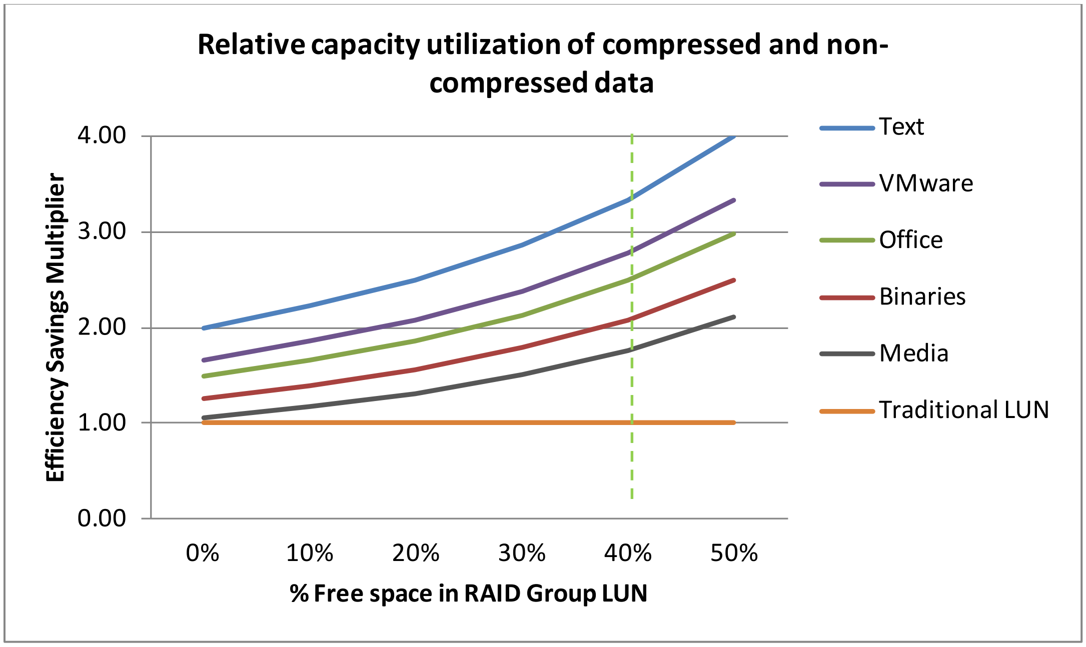


Figure 2. Relative capacity efficiencies of RAID Group LUNs

Figure 2 represents a model of effective capacity utilizations. The graph shows how different data types benefit from VNX capacity optimizations. These are compared against the "RAID Group LUN" case, which could be any data type on a volume without capacity optimization. The x-axis represents how much of the *user capacity* (the amount presented to servers) is free or unused space.

For example, assume a system with 1,000 GB usable capacity has 600 GB of data, which equates to 60 percent capacity utilization (40 percent free space). In the case of block storage, a RAID Group LUN would "stovepipe" that unused capacity to the assigned server. As Figure 2 illustrates, if the data were office files and capacity optimization were used, effective capacity utilization would be increased 2.5x, to 150 percent. (This is shown by a dotted line in the chart.) Other data types that are more compressible can deliver even higher effective utilizations.

Over 100 percent effective utilization means you can store more data than there is usable capacity. This is possible because through compression, the data is stored using less capacity than it normally would require. For file data, capacity savings from compression and single instancing are returned to the VNX file system for use by other files. For block data, thin-LUN Virtual Provisioning is used to return unused capacity to the storage pool for use by other LUNs. The bottom line is that capacity that would normally be allocated to seperate servers, but not used, is available for other data when capacity optimization is used.

Space reduction process

VNX File Deduplication and Compression has a flexible policy engine that specifies data for exclusion from processing and decides whether to deduplicate specific files based on their age. When enabled on a file system, VNX File Deduplication and Compression periodically scans the file system for files that match the policy criteria, and then compresses them.

VNX File Deduplication and Compression employs SHA-1 (Secure Hash Algorithm) for its file-level deduplication. SHA-1 can take a stream of data less than 264 bits in length and produce a 160-bit hash, which is designed to be unique to the original data stream. The likelihood of different files being assigned the same hash value is extremely low. Optionally, you can also employ a byte-by-byte comparison to confirm identical files detected by SHA-1 or disable file-level deduplication in general. If a user wanted to switch compression types for a specific file it would first have to be decompressed and then re-compressed with the preferred algorithm, default or deep compression.

The compressed file data is checked to determine whether the file was already identified.

- If the compressed file data was not already identified, it is copied into a hidden portion of the file system. The space that the file data occupied in the user portion of the file system is freed and the file's internal metadata is updated to reference the copy of the data in the hidden portion of the file system.

- If the data associated with the file was already identified, the space it occupies is freed and the internal file metadata is updated. Note that VNX detects non-compressible files and stores them in their original form. However, these files can still benefit from file-level deduplication.

Minimizing client impact

VNX performs all deduplication processing as a background asynchronous operation that acts on file data *after* it is written into the file system. It does not process data while file data is written into the file system. This avoids latency in the client data path because access to production data is sensitive to latency.

In addition to doing all the processing in the background, you can configure VNX File Deduplication and Compression to avoid processing the *hot data* in the file system. Hot data is any file that clients are actively using. Note that hot data is defined by how recently clients accessed or modified the files. By not processing active files, you avoid introducing any performance penalty on the files that clients and users are accessing. Surveys of file system data profiles show that typically only a small amount of the data in a file system is in active use. This means that VNX File Deduplication and Compression processes the bulk of the data in a file system without affecting the production workload.

A comprehensive data management strategy often involves archiving files that are not used for some time to an alternative tier of storage. This can be done with another product such as the Cloud Tiering Appliance (CTA). If you use this strategy, VNX File Deduplication and Compression maximizes storage efficiency for those files that are no longer actively used but are active enough not to qualify for archiving. If you combine both deduplication and archiving, you can potentially create a multi-tiered storage solution that provides greater storage efficiency.

The policy engine uses a defined default policy to scan the files in a file system in which deduplication is enabled. The default policy is a result of the investigation and analysis of how typical files age from active to inactive use in different industry and sector types.

However, the default policy may not meet every company's information lifecycle needs. VNX File Deduplication and Compression provides the flexibility and granular control to allow administrators to define their own hot data parameters, although EMC recommends careful planning and analysis before changing the policy. Deduplication policies are set at the Data Mover and file system levels.

Administrators can configure several policy settings to determine what constitutes active and inactive files in their environment. Changes made at the Data-Mover level change the policy settings for all deduplication-enabled file systems mounted on that Data Mover. Changes made at the file system level provide further customization of individual file systems, overriding the changes made at the Data Mover level.

Automated scheduling and self-throttling (based on CPU load) reduce the impact of deduplication on the VNX. Each Data Mover scans and deduplicates only one file system at a time. If VNX detects that the CPU load of the Data Mover exceeds a user-defined threshold, the process throttles its activity to a minimal level until the CPU load falls below a low-activity threshold. This means that the deduplication and reduplication processes effectively consume CPU cycles that would otherwise be idle. As a result, it does not affect the system's ability to satisfy client activity.

VNX File Deduplication and Compression targets inactive files and avoids new files that are considered active. For this reason ,this feature does not need to scan frequently to find aged files. Administrators can adjust this frequency, and they can prompt the system to scan a specific file system immediately, if required.

Backups with NDMP PAX use a filter to determine whether to store space-reduced files in their compressed format or in their original form. Filtering is used so that the restore time is not affected for files that provide a minimal amount of savings. Settings can be changed per file system or at the Data Mover level. Note some settings are only available at the file system level.

Setting	Definition	Default value
Access Time	Length of time in days that the file has not been accessed	15 days
Modification Time	Length of time in days that the file has not been modified	15 days
Minimum Size	Files less than this size will not be deduplicated	24 KB
Maximum Size	Files greater than this size will not be deduplicated	8 TB
File Extension Exclude List	Files with the specified extensions will not be deduplicated	None
Pathname Exclude List	Directories with the specified pathname will not be deduplicated	None
Minimum Scan Interval	Frequency with which the deduplication policy engine will scan a deduplication-enabled file system	7 days
SavVol High Water Mark	Usage capacity percentage of the file system's SavVol at which the space reduction process will not proceed	90%
CPU % High Water Mark	If the CPU reaches this level, deduplication will throttle down	75%
CPU % Low Water Mark	If deduplication is throttled down and the CPU level returns to this level, deduplication will throttle back up	40%
CIFS Compression Enabled	Enables CIFS compression	On

Setting	Definition	Default value
Backup Data High Water Mark	Percentage of the logical size of the file that the space-reduced size should be for NDMP to back up the file in its space-reduced format	90%
Case Sensitive	Defines whether case-sensitive (for NFS environments) or case-insensitive (for CIFS environments) string comparisons will be used during scans.	Off
Compression Method	Indicates whether the compression algorithm is set to fast or deep. This option is valid for VNX systems that use version 7.1 and later.	Fast
Duplicate Detection Method	Detection method for deduplication, options are sha1, byte, or off	Sha1

Table 2. VNX File Deduplication and Compression settings

Client input/output to space-reduced files

The VNX File Deduplication and Compression feature does not affect the client input/output (I/O) to files that have not been deduplicated. The feature does not introduce any additional overhead for access to files that it has not processed. The default policy is designed to filter out files that have frequent I/O access and thus avoid adversely affecting the time required to access those files.

Read access to deduplicated files is satisfied by decompressing the data in memory and passing it back to the client. VNX does not decompress or alter any data on disk in response to client read activity. In addition, random reads require decompression of the requested portion of the file data and not of the entire file data. Reading a file that is compressed can take longer than reading a file that is not compressed because of the decompression activity. However, the opposite may also be true. Reading a compressed file is sometimes faster than reading a file that is not compressed. This is because less data needs to be read from the disk, which more than offsets the increased CPU activity associated with decompressing the data.

A client request to write to or modify a deduplicated file causes the requested portion of the file to reduplicate (decompress) in the file system. At the same time, the deduplicated data must be preserved for the remaining references to the file. The following three factors mitigate this effect:

- Most applications do not modify files. They typically make a local copy, modify it, and when finished, write the entire new file back to the file server, discarding the old copy in the process. Therefore, reduplication on the file server does not occur. The file is just replaced.

- VNX avoids processing active files (accessed or modified recently) based on policy definitions. Therefore, deduplicated files are less likely to be modified and, if they are, performance is less likely to be a critical factor.
- When a client writes to a deduplicated file, the Data Mover writes only the individual blocks of text that have changed. The entire file is not decompressed and reduplicated on the disk until the sum of the number of individual changed blocks and the number of blocks in the corresponding deduplicated file is greater than the logical file size.

Deploying VNX Deduplication and Compression for File data

The VNX Operating Environment for File offers several convenient methods for managing deduplication. There are user-defined deduplication policies available in the Unisphere software as well as integrated options within VMware® vCenter and Windows Explorer. User-defined policy attributes identify which files to deduplicate and compress. Users can set these controls at the file-system or Data Mover level.

You have the ability to enable or disable CIFS compression by using the Microsoft Windows compression attribute. Enabling this feature allows the user to see compressed files displayed in a different color in Windows Explorer than non-compressed files.

Newly introduced in the VNX OE 7.1 is the option to select deep file compression. This compression method is optimized for space efficiency rather than speed. This alternate compression method is designed to produce up to 30% more space savings than the space savings produced by the fast (default) method, though sacrificing decompression speed and CPU utilization on the Data Mover to attain additional space savings. This alternate method is intended to be used on file systems with “cold data” that is infrequently accessed where space savings is much more important than file access speed. File archiving is a prime use case for deep compression. It is not recommended for compressing VMs on NFS file systems due to the increased decompression time.

In VMware environments, file-level compression can be invoked within the EMC VSI for VMware vSphere: Unified Storage Management plug-in. Compression is the term used within vCenter, but this includes compression and single instancing. Using the plug-in, compression can be enabled at the NFS datastore level, individual virtual machine level, or virtual disk level. Right clicking on a cluster, host, datastore, or VM, displays the compression options. When compression is enabled on the datastore, all virtual disks in the datastore are processed. When compression is enabled on a virtual machine, all existing virtual disks associated with that VM are processed.

Using deduplication settings

In the Unisphere software, you configure policy settings at the Data Mover and file system levels. To access the **Deduplication Settings** at the Data Mover level, select **Storage** in the navigation pane. Then click **Deduplication settings** on the tree on the right-hand side under File Storage. Figure 3 shows the Data Mover **Deduplication Settings**.

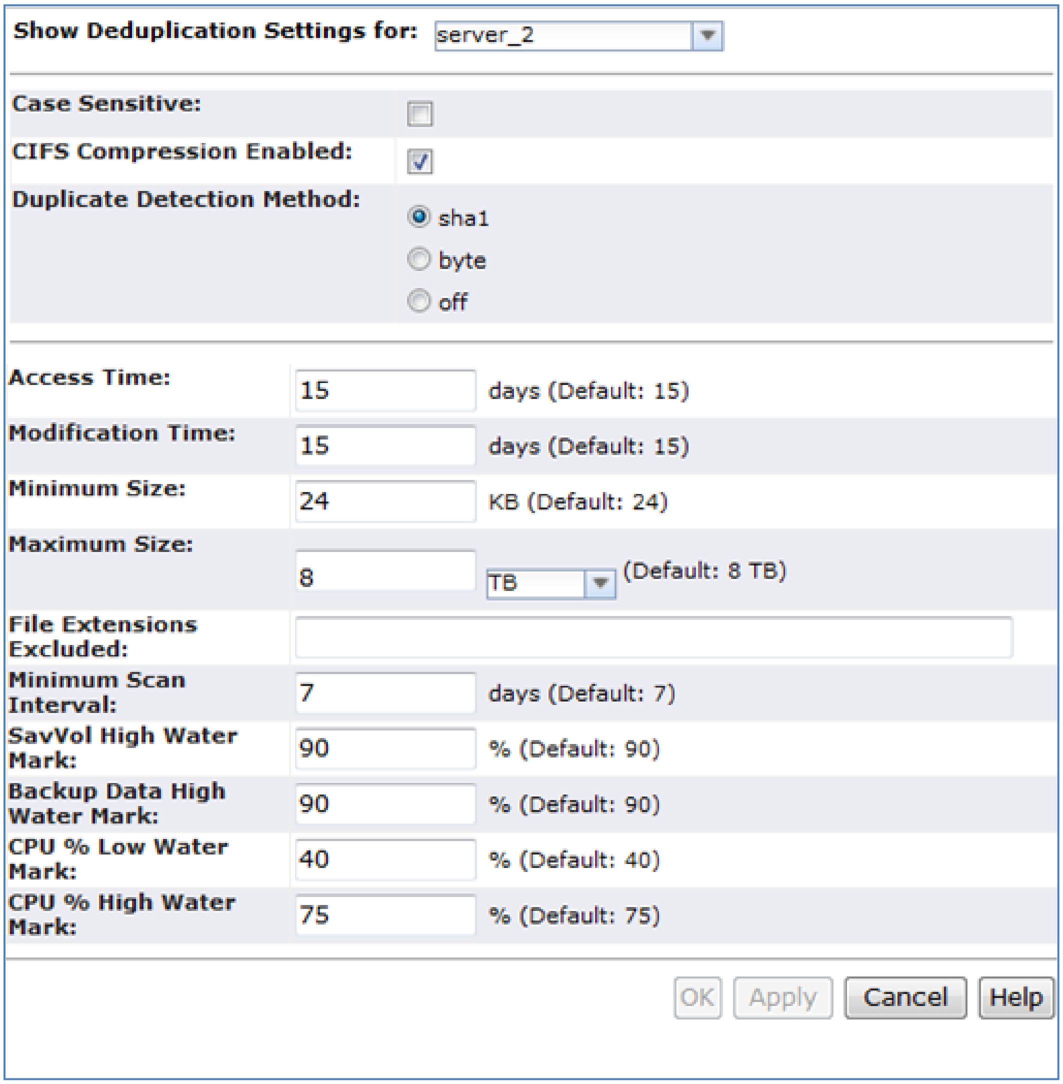


Figure 3. Deduplication Settings at the Data Mover level

Figure 4shows the **Deduplication Settings** tab at the file system level, which is in the **File System Properties** window of a deduplication-enabled file system.

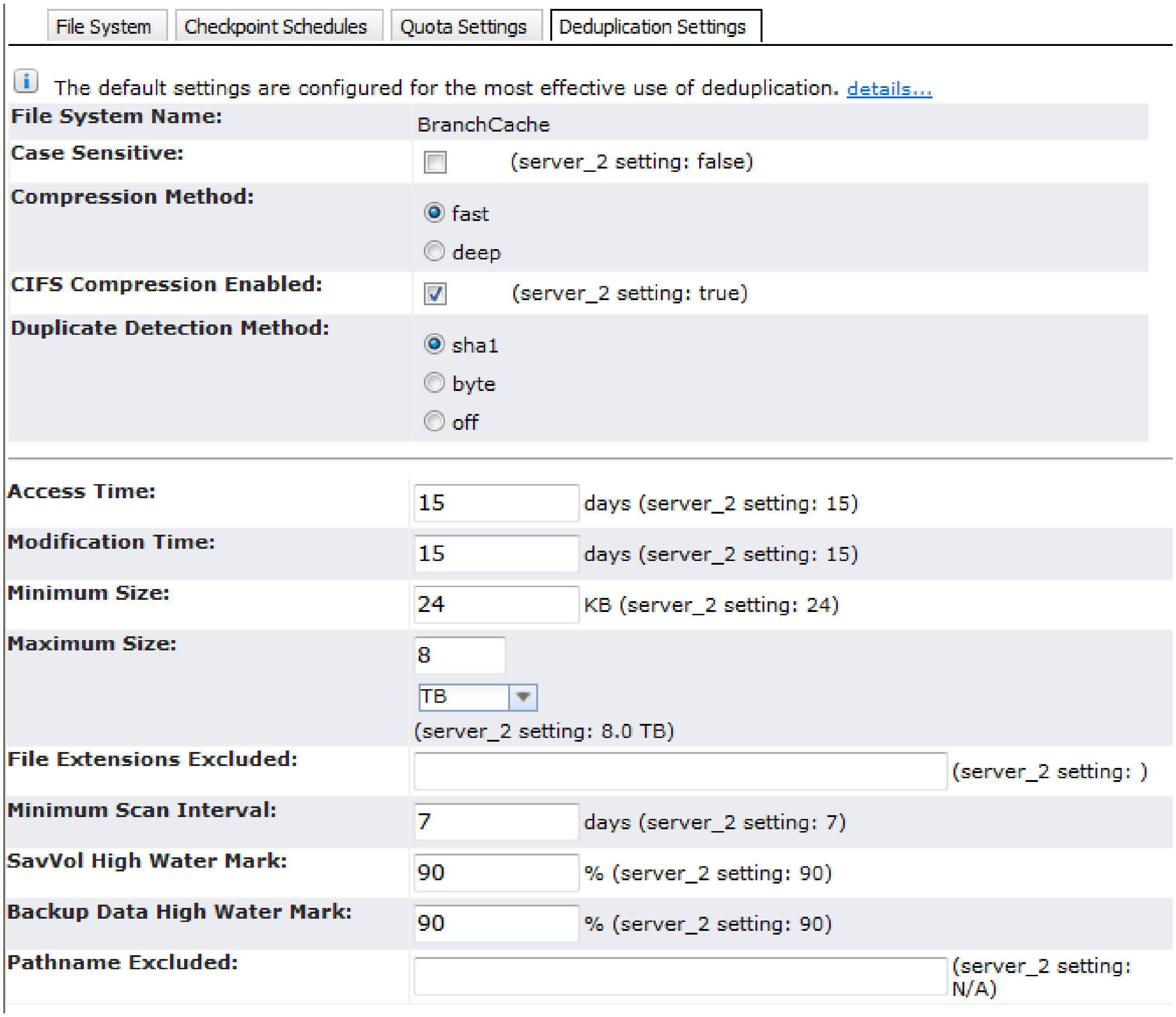


Figure 4. Deduplication Settings tab at the file system level

The deduplication settings for Data Movers and file systems are similar. However, you can configure the CPU low and high watermark settings only at the Data-Mover level, whereas you configure the pathname exclusion setting only at the file-system level.

Enabling deduplication and compression on a file system

When you create a new file system, you can enable VNX File Deduplication and Compression on that file system in the Unisphere **New File System** window. For existing file systems, you can select **On** in the Unisphere **File System Properties** window, as shown in Figure 5

Viewing the deduplication state

After you enable deduplication on a file system, VNX File Deduplication and Compression periodically scans it and looks for files to deduplicate. You can use the CLI fs_dedupe command to query the state of the deduplication process for each file system, or view the state in the Unisphere **File System Properties** window as shown in Figure 5

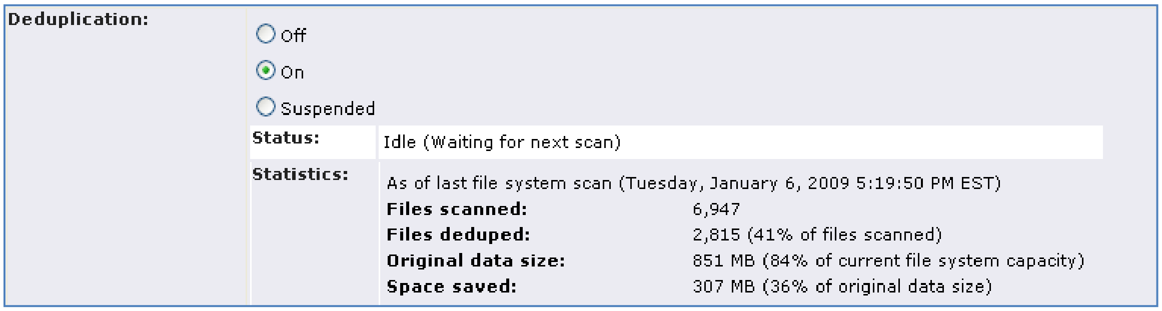


Figure 5. VNX File Deduplication and Compression state in Unisphere

Changing the deduplication state

You can change the state of deduplication in the file system properties. See Figure 5above. When changing the compression method new files will be compressed with that new method, however existing compressed files will not be changed. When files are decompressed and recompressed they will use the new method.

The default VNX File Deduplication and Compression state for a file system is Off. In this state, the file system has no deduplicated files and the policy engine does not scan it for files to deduplicate.

When VNX File Deduplication and Compression is in the On state, the file system may contain deduplicated files, and the policy engine scans the file system for more files to deduplicate on its next scheduled run.

The Suspended state means that the VNX File Deduplication and Compression processing is paused for the file system. In the Suspended state, the file system may contain deduplicated files. However, the policy engine does not scan for additional files to deduplicate.

You can change deduplication states at any time. Changing the deduplication state from On or Suspended to Off reduplicates all deduplicated files in the file system. Before processing the request to turn off deduplication, the system checks whether there is sufficient space in the file system to complete this process. If there is insufficient space, the system informs you about the amount of additional space that is required to complete the operation, and recommends that you extend the file system.

Figure 6 shows the Properties dialog boxes for a datastore and a virtual machine with compression enabled. Checkboxes for enable/disable as well as savings due to compression are available in each dialog box.

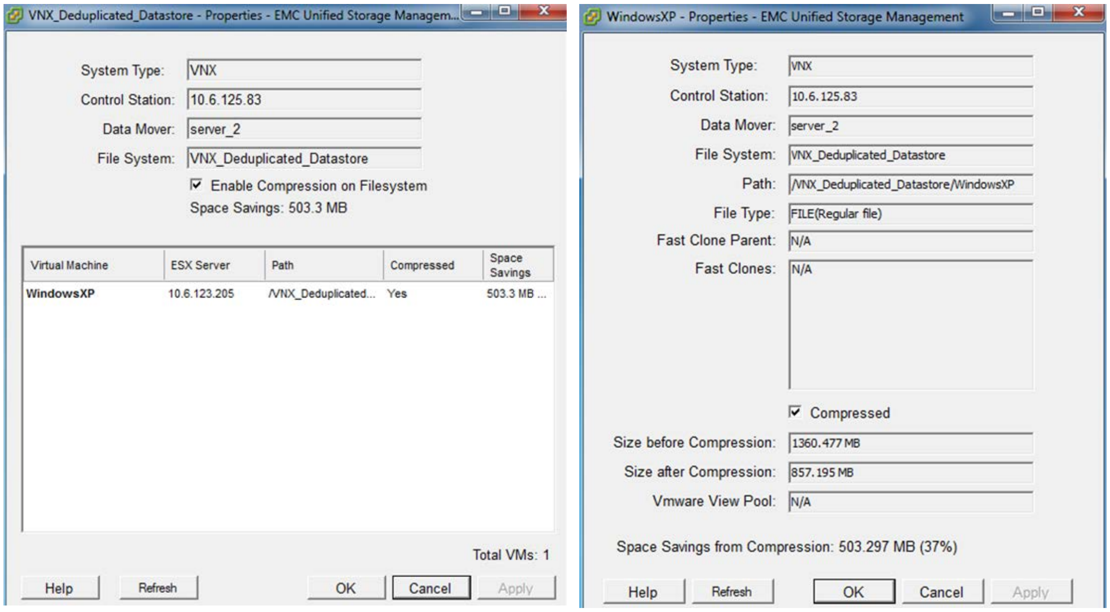


Figure 6. Compression options and space savings in Datastore and VM Properties dialog boxes

When processing virtual disks, the file compression feature is aware of the virtual-disk structure and only processes the .vmdk file. Swap and temp spaces are excluded because it is not practical to process these files. This optimization allows the virtual disks on NFS datastores compressed through the vCenter plug-in to remain compressed, even when active. The system in this case ingests new data to the compressed file asynchronous to the write.

Users can manage the compression of files and directories on CIFS shares in a similar fashion to virtual disk files in vCenter. Compression in this case also includes single instancing. Windows users can enable file compression at the share, directory, or individual-file level from within Windows Explorer. Files compressed in this fashion also remain compressed with new changes ingested as necessary.

Figure 7shows a compressed file, file1, displayed in blue within Windows Explorer.

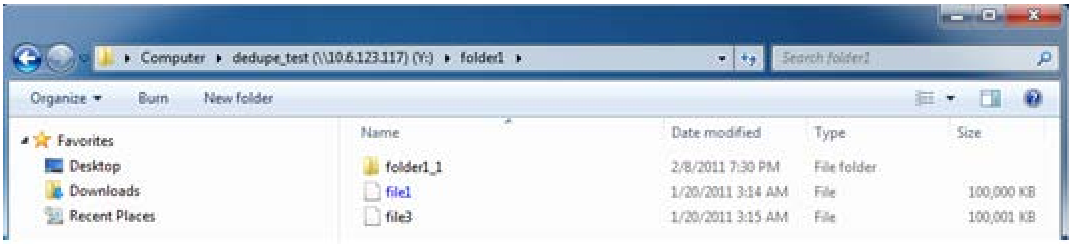


Figure 7. A compressed file in Windows Explorer

Figure 8shows the properties for file 1. Enable compression for an individual file using the Advanced Attributes dialog box.

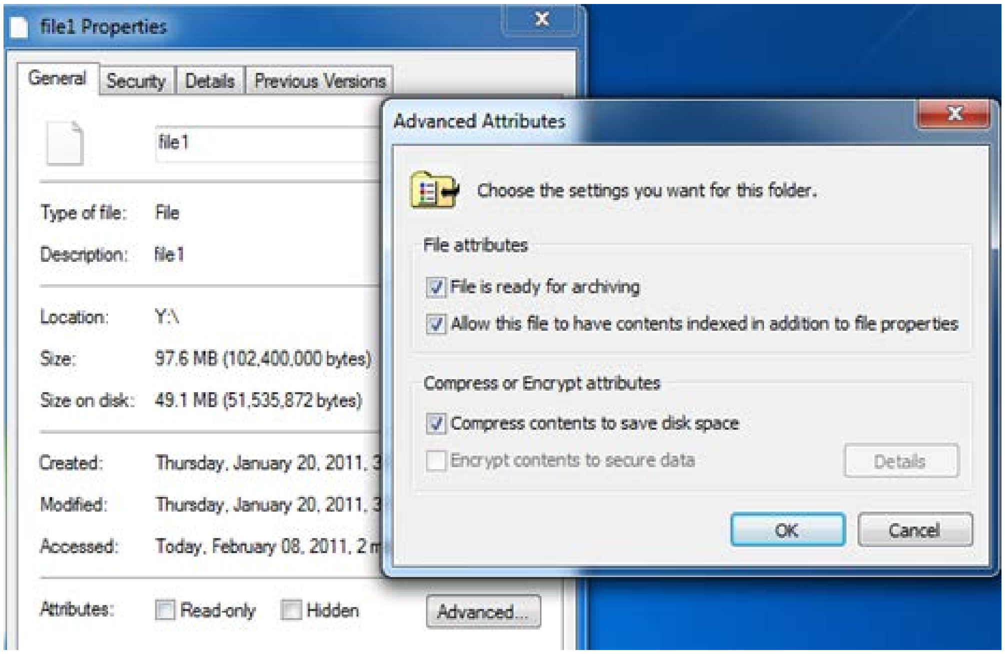


Figure 8. The properties of a compressed file, file1

With policy-based management, users have the freedom to define default deduplication behavior. The additional functionality provided within vCenter and Windows allows users to manage files explicitly over and above the general deduplication policy.

Viewing deduplication statistics

As shown previously in Figure 5, VNX displays the results of the deduplication process on the file system data with the following statistics:

- **Timestamp** of the last successful scan of the file system.
- **Files scanned** — Total number of files that the deduplication policy engine looked at when it last scanned the file system.
- **Files deduped** — Number of files that the deduplication policy engine processed to save space. It also shows the percentage of deduplicated files versus scanned files.
- **Original data size** — Space required to store the data in the file system if it is not deduplicated. This number might exceed the capacity of the file system, in which case the file system is said to be overprovisioned. This is shown by the ratio of the original data size to the file system capacity, which is also displayed.
- **Space saved** — Amount and percentage of space saved by deduplication. This is calculated by subtracting the actual space used to store data after deduplication from the original data size.

After the first scan, statistics are reported as static values based on the last successful scan.

Compress operations for Block

Compresion on the block side differs from file side deduplication and compression. As Compression process data is in 64 KB increments, compressed data is written to the LUN if at least 8 KB of the 64 KB can be saved. If the resulting savings from compression is less than 8 KB, the data is written to the LUN uncompressed. The VNX will not compress if there is not sufficient savings, which shows the efficiency of the product.

Although you can enable compression on any LUN type, when the initial compression process completes, the LUN will become thin because thin LUN technology allows the compression process to return space to the pool. For thin LUNs, 8 KB is the smallest unit of capacity allocation.

Capacity is allocated from a pool to a thin LUN in 1 GB slices. Over time, if enough 8 KB blocks are freed by compression, 1 GB slices can be returned to the pool for use by any LUN in the pool. This process starts when enough capacity is saved to free a slice and may continue after the compression process has completed.

As of the EMC® VNX™ Operating Environment (OE) for block release 5.32, file version 7.1, the compression method for Block has changed. Now the system will compress data in place. This means we no longer take up additional space when doing compression.

After compressing a LUN, users will see the consumed capacity of both the LUN and pool decrease. Consumed capacity in this case is reduced due to savings from both thin LUNs and compression. Examples of this can be seen in Figure 9 and Figure 10. Users will not see any change in reported capacity usage at the host level as compression is transparent to the host.

Figure 9 (top) shows the compression of a thick LUN. The user capacity of the LUN is 250 GB, but 259 GB are consumed in the pool since thick LUNs consume capacity equal to the user capacity plus metadata. After compression (bottom), the LUN only consumes 34 GB of space in the pool, for a savings of just over 220 GB. This savings represents the benefits of both moving the LUN from a thick LUN to a thin LUN and compressing the data.

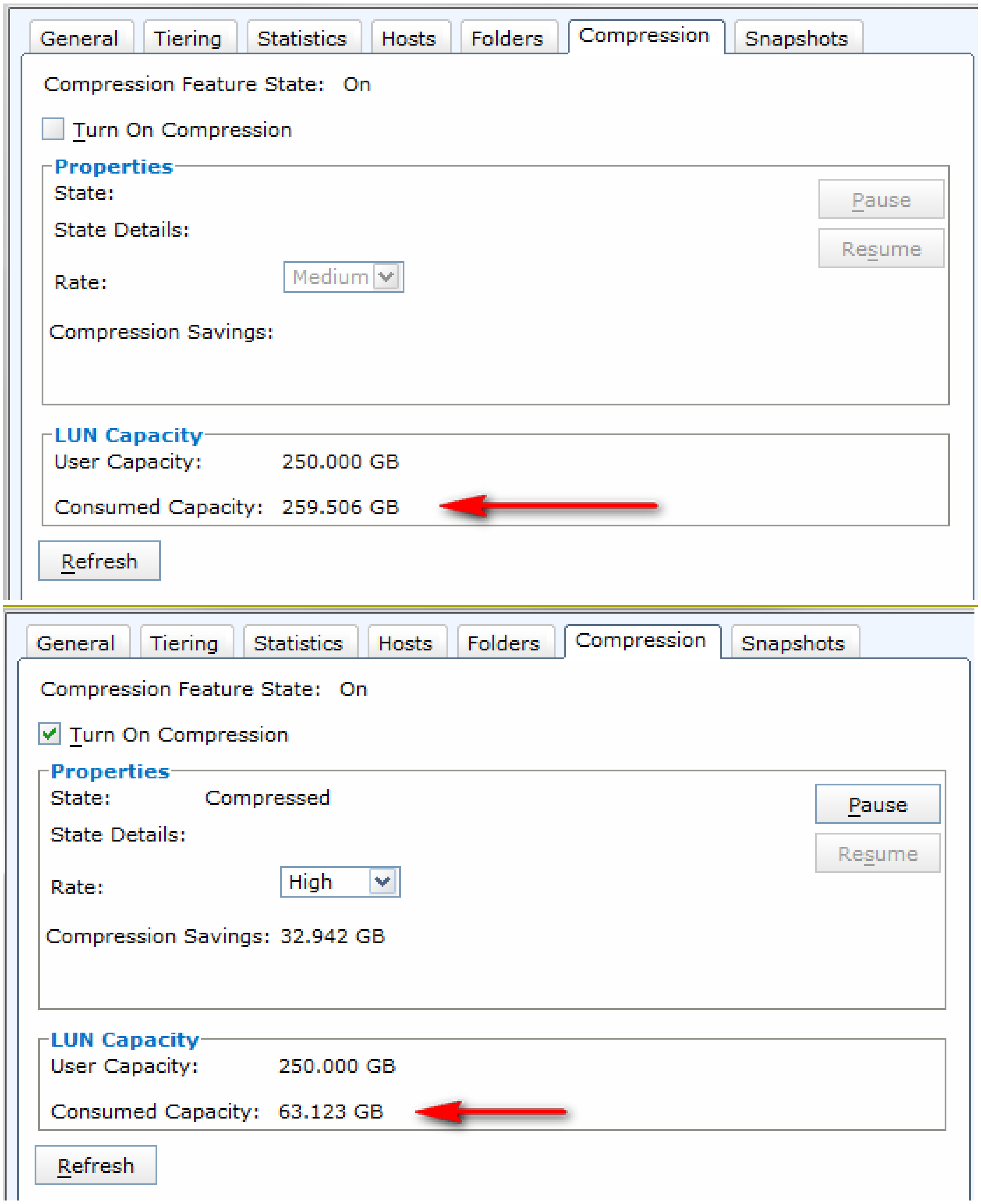


Figure 9. LUN Properties – Compression tab before and after compression

Figure 10 shows the change in consumed capacity and resulting change in Percent Full. The capacity saved from compressing the LUN is returned to the pool for use by any other LUN in the pool. Server-visible metrics like User Capacity, Subscribed Capacity, and Percent Subscribed remain unchanged.

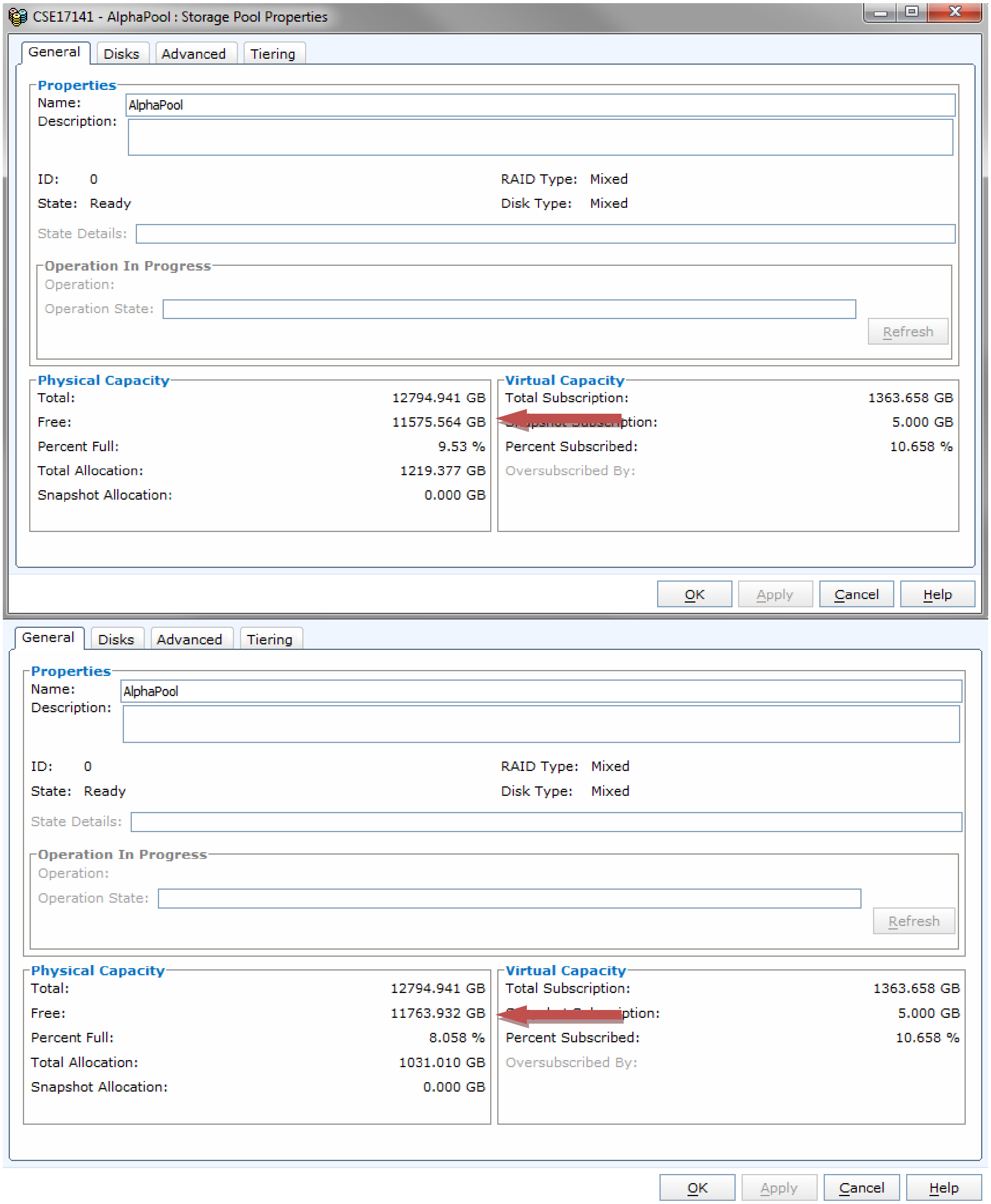


Figure 10. Pool properties before and after compression

There are user settings for the compression rate of High, Medium, and Low in the LUN Properties dialog box under the Compression tab as shown in Figure 9. The compression rate setting determines how aggressively compression is performed, not the level of data reduction. The compression rate setting applies to initial compression, subsequent compression of new data, and decompression operations. The default rate setting is Medium, which can be changed at any time.

An online migration of RAID group LUNs to thin LUNs is performed when compression is enabled. All aspects of the migration are handled for the user with no interruption to server I/O. Since RAID group LUNs do not already reside in a pool, the user is prompted to select a destination pool for the compressed LUN. Figure 11 shows the popup “Turn On Compression” dialog box that allows the user to select an existing pool or launch the Create Pool dialog box to create a new destination pool. Users can also set the rate of the migration in this dialog box. Keep in mind this is for migration and not the in place conversion that takes place when LUNs are already in the pool.

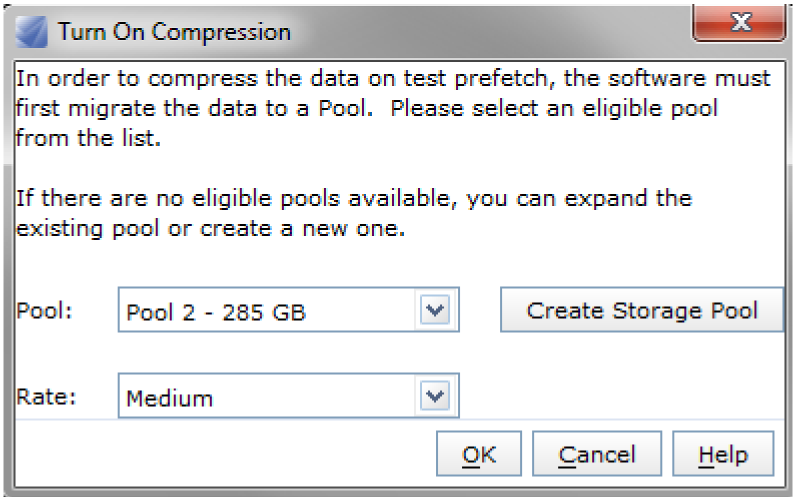


Figure 11. Destination pool selection for RAID group LUN compression

All currently eligible pools are listed in the Pool pull-down menu. Pools are only shown if they have enough free capacity to accommodate the user capacity of the RAID group LUN. Capacity equal to the user capacity of the RAID group LUN is reserved in the pool to ensure the process can complete. After the migration is complete, the original RAID group LUN is unbound and its capacity is available to create new LUNs in the RAID group.

Decompression operations

The decompression process restores compressed data to its original size. When compression is disabled on a compressed LUN, the entire LUN is processed in the background. When the decompression process completes, RAID Group LUNs and Thick LUNs are migrated back to Thick LUNs. Thin LUNs are migrated to a new Thin LUN with a defrag process which might give better performance after the decompression. The LUN, while remaining a thin LUN, is fully allocated to preserve the consumed capacity of the original LUN type.

If the pool becomes about 91 percent full, the compression state of the LUN will become **system paused**. Host I/O can continue to the LUN uninterrupted while compression is in this state. This behavior is implemented as a safeguard to ensure there is pool capacity available for new host I/O. If the user decides to continue decompression without adding capacity to the pool, the user can manually override this safeguard and resume decompression. However, the compression state will again become **system paused** if the pool reaches 98 percent full. This safeguard cannot be overridden.

LUN Migration offers a compelling alternative to decompression. This is an attractive option when a LUN is more active than expected or in anticipation of a known period of high I/O activity. In these cases, in addition to decompressing data, a change in LUN type may be warranted and/or different physical spindles may be used. All of these changes can be addressed with a single online LUN Migration operation. Users can migrate a compressed LUN to a RAID Group LUN or a LUN in another pool for example as long as the target LUN's user capacity is the same or larger than the compressed LUN's user capacity

Deploying VNX Compression for Block data

Within block storage, there is no notion of a "file," therefore compression is a practical approach to capacity optimization that offers significant space savings benefits for many data types. You can easily manage compression by using either the Unisphere software or the CLI at both the LUN and system levels. Once enabled, the system automatically manages the processing of new data based on the amount of new data coming in compared to system-defined thresholds.

Block data compression is tightly integrated with Thin LUNs. When compression is enabled on a LUN, the LUN becomes a thin LUN if it is not already. The software automatically handles transition of non-thin LUNs to thin LUNs. As thin LUN blocks are freed, they can be returned to the pool for use by other LUNs in the pool.

Note that LUNs with block compression enabled should *not* be used for VNX file volumes. Use VNX file deduplication and compression exclusively for file data. More granular control is available for file data rather than block, so the system can identify inactive data to process versus active data. Optimizations such as virtual disk file awareness are only available in the file implementation.

Block data compression is intended for relatively inactive data that requires the high availability of the VNX system. Consider static data repositories or copies of active data sets that users want to keep on highly available storage. Block compression is fully compatible with replication software delivered in the VNX Local and Remote Protection Suites, so there are many use cases where these products may be used to create a compressed copy of a data set.

Figure 12shows the LUN table of a VNX system in the Unisphere software. Users can configure optional columns for the LUN attributes “thin” and “compression.”

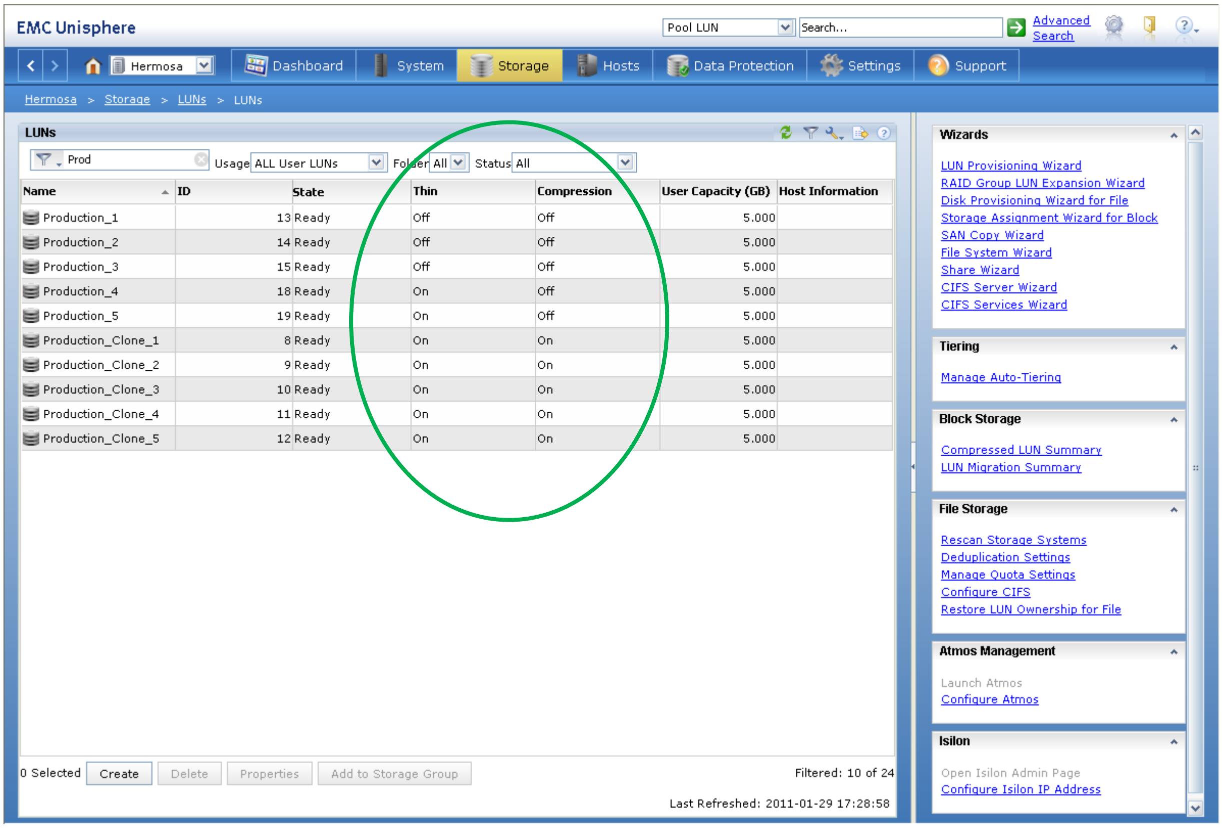


Figure 12. VNX LUN table in the Unisphere software

Controls for block compression are available in the LUN Properties dialog box under the Compression tab, as shown in Figure 13. In this dialog box, users can enable and disable compression *for the LUN.*

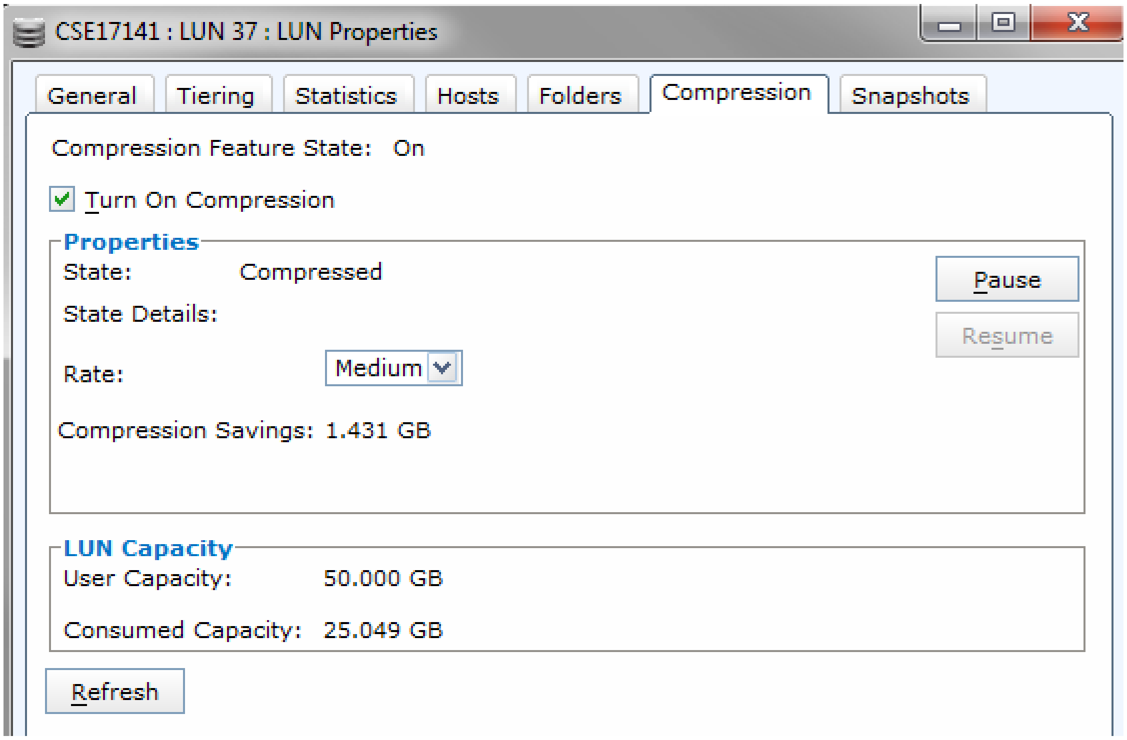


Figure 13. LUN Properties dialog box, Compression tab

You can also set the compression rate on this dialog. In this case, rate refers to the speed at which the compression process operates, *not* the level of compression effort. The options are High, Medium (default), and Low. LUN-level Pause and Resume capabilities are available on the right side of the dialog box. The compressions savings are also listed under the rate.

User capacity is the capacity as presented to the server. Consumed capacity is the amount of physical capacity allocated to the LUN. If compression is enabled, it represents the end result of both thin provisioning and compression.

The Compressed LUNs Summary dialog box shown in Figure 14 provides a consolidated view of block compression activity for all LUNs. It also provides system-level compression control by using the Pause button and Resume button at the bottom of the dialog box.

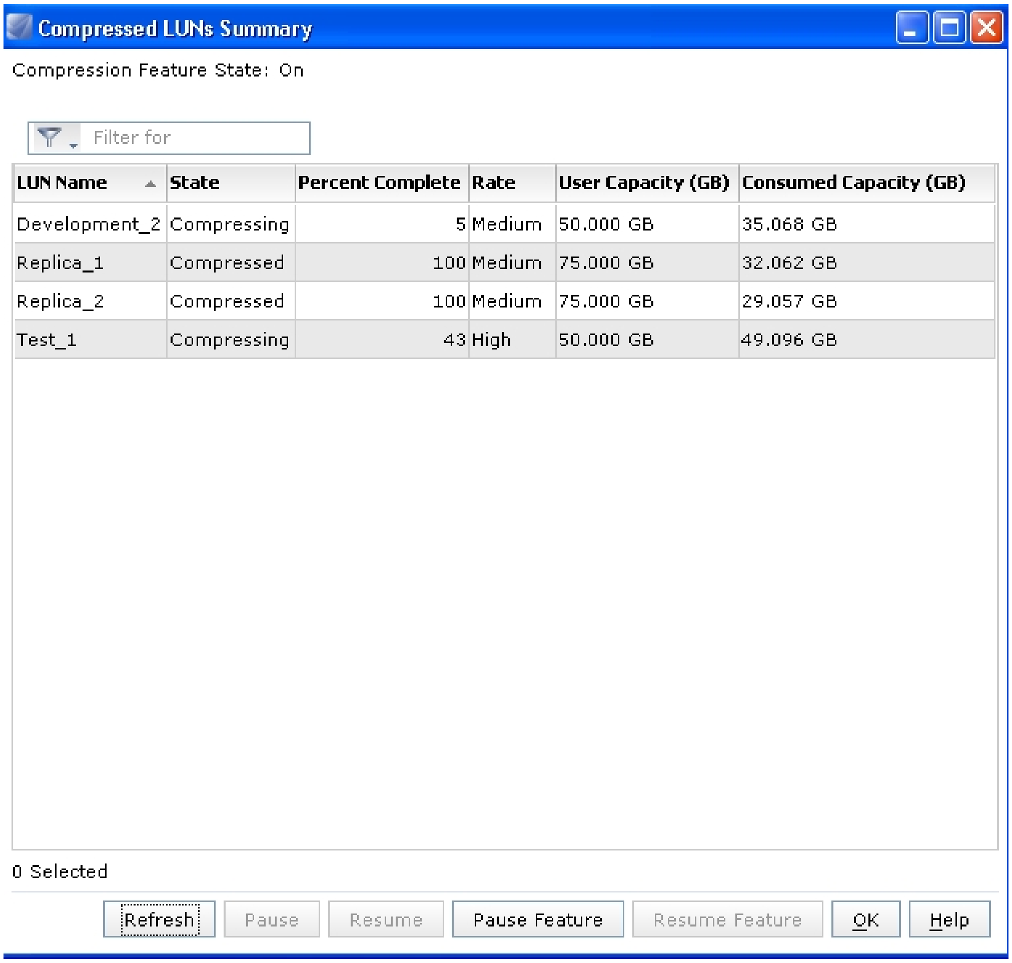


Figure 14. Compressed LUNs Summary dialog box

When using the Pause Feature option, all LUN-level compression operations are paused. Compression operations occurring via LUN Migration can be paused in the same way as other Compression operations. These operations can be cancelled in the Compression tab in the LUN Properties dialog box. EMC recommends pausing the compression feature at the system level during known periods of high system utilization if response-time-sensitive applications are running. Otherwise, the compression and subsequent space reclamation processes will use CPU and cache resources that may impact response-time-sensitive applications.

Performance

VNX File Deduplication and Compression is designed to be a noninvasive feature that can help achieve storage efficiency while avoiding high-performance usage on the Data Mover when the policy engine is active. This feature runs on a schedule. The policy engine is designed to look for old files, not newly modified or created files.

The scan performance and the deduplication and compression processes depend on the system load on the Data Mover. At the Data Mover level, deduplication:

- Scans up to 3.6 billion files per week at an average rate of 6,000 files per second
- Processes 1.8 TB (at 3 MB/s) to 14 TB (at 25 MB/s) of data per week
- Uses approximately 5 percent of CPU processing power

Because read access to a deduplicated file is a pass-through operation, users see little impact in terms of read performance. Sometimes, reading a deduplicated file is faster than reading a file that is not deduplicated. Random reads of deduplicated files may appear to users in the same way as reads of normal files appear to them. Large sequential reads of deduplicated files may take longer than reads of the files in their unprocessed form. Read performance also depends on the number of streams and compressibility of the files. Read performance for deduplicated files compared to non-deduplicated files is summarized as:

- Random read – 54 percent to 139 percent
- Sequential read – 68 percent to 107 percent

Reading a deduplicated file requires more CPU cycles than reading an unprocessed file. Hence, performance can be affected if many deduplicated files are read simultaneously. However, you can minimize the performance impact by tuning the policy to meet the needs of your environment.

Writing to a deduplicated file prompts a reduplication of the requested portion of the compressed file data. This effectively removes the limit on the maximum size of a file that can be deduplicated.

With PAX-based NDMP, it is important to note that there are performance ramifications for space-reduced restores compared to non-space-reduced restores. Because backing up a deduplication-enabled file system might reduce the amount of data written to tape, the backup time may be less (depending on bottlenecks in the backup processes). However, the restore time may be longer dataset. This is because the system deduplicates the files being restored in real time.

Migration to VNX with Deduplication and compression example

The following examples identify a customer wanting to migrate data from a Windows server to the VNX system. These real world scenarios show what happens when you migrate data and the importance of deduplication being enabled prior to migration.

Note: The fsUtils package contains the following utilities:

- fsScan: Produces data files that act as small databases of all the meta data in a file system or group of file systems.
- exScan: Provide assessment information for the current usage breakdown of existing Exchange mail characteristics such as size and last access time and to assess potential space savings from an archiving solution implementation.
- fsReports: Create reports from fsScan data files.

- fsDiff: Compare two fsScan output files

<u>In these examples we gathered the meta data with the fsscan utility which creates a .dtl file. Then we ran reports on the dtl file with the fsreports utility. This gave us information about files to dedupe and archive.</u>

Customer A

Customer has a Windows server that will be migrated to VNX:

- 2TB of MS Office data
- Highly visible project
- Existing performance issues

	File System	Deduplication Results	Savings
Initial Analysis of Windows server	2TB	Not enabled	0
FsUtil output	2,471GB	4,056,632 files can be archived 3,522,207 files can be deduped	73% of all files have gone 180+ days untouched 2,236GB combined size, or 90% of total size at 40% compression: 893GB potential savings
After Migration to VNX	1,590GB	3,461,285 files deduped	881GB space saved

Customer B

Customer has an older NAS that needs to be migrated to a VNX:

- Host based migration
- Highly visible
- Must have replication set from day one

Deduplication was enabled after IP Replication & Checkpoints & post-migration

	File System	Deduplication Results	Savings
Initial Analysis of Older NAS	760GB	Not enabled	0
FsUtil output	760GB	770,725 file can be archived 844,761 files can be deduped	69% of all files have gone 180+ days untouched 718GB combined size, or 75% of total size at 40% compression: 288GB potential savings
After Migration to VNX	? ? ?	44,862 files deduped	9GB space saved

??WHAT JUST HAPPENED??

- SavVol was at 82% of 137GB size (Checkpoints & Replication overhead)
- Dedup had only 8% of 137GB, or 10GB to operate
- As soon as the FS filled 10GB of changes, deduplication stopped for a week

After extending the SavVol by 300GB:

	File System	Deduplication Results	Savings
Initial Analysis of Older NAS	760GB	Not enabled	0
FsUtil output	760GB	770,725 file can be archived 844,761 files can be deduped	69% of all files have gone 180+ days untouched 718GB combined size, or 75% of total size at 40% compression: 288GB potential savings
After Migration to VNX	276GB	845,293 files deduped	484GB – 300GB for SavVol extension = **184GB** total savings

Summary: Do not turn on checkpoints before doing migration with deduplication and compression enabled.

When you turn on checkpoints before doing a migration with dedupe enabled. The system will dedupe files to clear space, freeing up blocks to be used. This could cause the checkpoint savvol to grow very large as writes will start to fill up those freed up blocks. This will cause the savvol to track those new blocks as needing to be saved into the savvol, this can cause the savvol to keep expanding to track all the new changes.

Limits and interoperability

Limits

The following table details the limits for the Data Compression feature.

VNX Model	5100	5300	5500	5700	7500
Total Compressed LUNs per pool	N/A	512	1024	2048	2048
Concurrent Compressions per SP	N/A	10	10	16	20
Concurrent migrations per array	8	16	16	24	24

Table 3. Limits for the Data Compression feature

Notes:

- Concurrent compression operations include initial compression of data on thin LUNs, compression of new data, and decompression.
- Concurrent migrations include any initial compression of RAID group LUNs where the LUN being compressed must be migrated to a thin LUN. (This also includes decompressions in the VNX™ Operating Environment (OE) for block release 5.32 and later).
- Compression and migration limits are not dependent on each other. For example, SPA of a VNX can have five compression operations running, and the sixth will be queued. Simultaneously eight RAID Group LUN initial compressions could be running.

The following cannot be compressed:

- Private LUNs (including Write Intent Logs, Clone Private LUNs, Reserved LUNs, MetaLUNs, and Component LUNs)
- Snapshot LUNs
- VNX snap that has a snapshot
- VNX Snapshot Mount Point (SMP)
- LUNs provisioned for VNX for File
- A LUN that is already being migrated
- A LUN that is expanding or shrinking
- A mirrored LUN replicating to a storage system running pre-29 FLARE code

Interoperability

In general, a compressed LUN has the same interoperability as a regular thin LUN. Compressed LUNs can be a source or destination LUN for VNX for Block replication applications. A LUN's compression status has no bearing on the compression status of any other LUN in a replication operation. For example, if a non-compressed LUN is the source of a SnapView clone group, the LUNs in the clone group can be compressed and/or non-compressed. Furthermore, if a compressed LUN is the source of a clone group the clones can be compressed and/or non-compressed. Compression can be enabled or disabled while in use by a replication application. Using the same example, compression can be enabled or disabled on a clone without affecting the operation of the clone group.

When a replication application such as SAN Copy writes to a compressed LUN, the I/O is treated as new data. The I/O written by the replication application can exceed the uncompressed data threshold, in the same manner host I/O would, and trigger the re-compression process. Some common replication writes that may trigger recompression are initial synchronizations, MirrorView/S syncs, MirrorView/A updates, Incremental SAN Copy updates, and clone synchronizations.

When creating a new replica that is intended to be compressed, users may want to enable compression *after* the initial synchronization. This way, compression is not triggered by the initial synchronization, which is typically a high bandwidth operation. After the initial synchronization completes, compression can be enabled to process all of the data at once, and compress new data as warranted by the system defined threshold. This is not a requirement, but merely a suggestion that may make initial synchronizations faster. To do this, you must have the capacity to accommodate the uncompressed data.

FAST VP and FAST Cache are also compatible with compressed LUNs. FAST VP can help facilitate the lowest capacity for compressed LUNs. Since compressed LUNs are likely to have low performance profiles, FAST VP would probably place these LUNs on the lowest storage tier. Users can also explicitly set the tiering policy to **Lowest AvailableTier**, thereby ensuring compressed LUNs remain on the lowest-cost drives. To learn more about FAST VP, see the *EMC FAST VP for Unified Storage* white paper on Powerlink.

FAST Cache may help performance in cases where there are intermittent bursts of activity to a compressed LUN. However, it is not expected that it will bring compressed LUN performance up to par with non-compressed LUN performance. LUNs that become more active should either be decompressed or migrated to a non-compressed LUN. To learn more about FAST Cache see the *EMC FAST Cache* white paper on Powerlink.

Special cases for LUN Migration and SAN Copy

When using LUN Migration to manually migrate data to a compressed LUN, always create a new thin LUN target and enable compression before staring the migration. LUN Migration is tightly integrated with compression; it is used in the initial compression of RAID group LUNs prior to the VNX™ Operating Environment (OE) for block release 5.32. User-initiated LUN migrations to compressed LUNs can reap the same "compress on-the-fly" benefit as system-managed migration/compression operations. The destination LUN in this case should be a newly created thin LUN with no data and with compression enabled. After creating the new thin destination LUN and enabling compression, a user-initiated LUN migration compresses data inline with the migration I/O. At the end of the migration, the resulting LUN consumes the minimum amount of capacity possible. Specifying a destination LUN that has data on it is less efficient. The data is overwritten during the migration, and space reclamation is conducted as a separate process.

If using SAN Copy to replicate from a RAID Group LUN to a compressed destination LUN, do *not* enable compression on the destination LUN. Release 30 adds a space reclamation enhancement to Virtual Provisioning. Users can take advantage of this enhancement with SAN Copy when performing remote SAN Copy pull operations, or SAN Copy push or pull operations, within the same storage system.

This enhancement allows users to copy data from CLARiiON, Symmetrix®, or third-party storage systems to thin destination LUNs in a more efficient manner than with previous software. Whitespace or strings of zeros in the data are detected by the system, and physical capacity is not allocated to the thin destination LUN for those areas. This provides immediate capacity savings. In prior releases, this would have resulted in a fully consumed thin LUN. This enhancement is for remote SAN Copy pull sessions and SAN Copy push and pull sessions within the same storage system. Remote SAN Copy push sessions result in a fully consumed thin LUN unless the source of the migration is a thin LUN on a system running Release 29 or later.

If compression is enabled on the thin destination LUN prior to running the SAN Copy session, the space reclamation enhancement is not employed. In this case, the thin LUN is written to as if it were being fully consumed. At some point, the data written by SAN Copy triggers the compression process. After compression processes the data, the end result would likely be better than the space reclamation case alone. The compression algorithm would probably remove the same whitespace and zero strings, and compress the non-zero data wherever possible. Overall, it is more efficient to allow the SAN Copy pull session to complete without compression. This way, space reclamation is realized at the outset and the amount of capacity used is minimized throughout the process. After the copy is complete, compression can be enabled. Additional capacity savings can be realized from compression and the compression feature will not have to process as much data.

Resource consumption and performance

Moving datasets from RAID Group LUNs or thick LUNs to thin LUNs provides the benefits of recapturing and repurposing unutilized capacity and provides a “pay as you go” capacity consumption model. Users make this tradeoff when they do not have stringent IOPs and response time requirements. Compression takes this a step further, providing even greater capacity savings at the cost of additional system resources and LUN performance. Compression is best suited for data that is largely inactive but requires five 9s availability.

The compression rate settings offer control over the amount of resources that are dedicated to the compression process. For the standard compression process, which includes initial thin LUN compression and recompression of new data, the following CPU ranges are observed when running the maximum allowed concurrent compression operation.

Rate	VNX series	CX4 series
Low	< 10 percent	< 15 percent
Medium	12 percent	30-50 percent
High	40-65 percent	60-80 percent

Table 2. CPU utilization when running maximum concurrent compression operations

If only running half of the allowable concurrent operations at Medium or High, the range of expected CPU utilization is approximately half of the utilization ranges given above. Similar CPU utilization ranges can be expected for compression and decompression.

EMC recommends that you pause compression at the system level when response-time critical applications are running on the storage system. The process of returning capacity to the pool that has been freed by compression can contribute to CPU and write cache usage as data is consolidated onto as few slices as possible. This process can also continue after the compression process completes. Pausing the compression feature will ensure any background compression or associated space reclamation operations do not impact server I/O.

The initial compression of RAID group LUNs more closely tracks the behavior of standard LUN Migration to a thin LUN target. Compression is performed inline with the migration, so the overall rate may be 20-30 percent lower than migrating to non-compressed thin LUNs.

The relative compression throughput at the Medium rate for different storage system models is shown in Figure 15. This represents how much data is being processed by compression operations. Data used to generate the charts was compressed at a ratio of 1.5:1, or 65 percent. Compression operations were up to the maximum number of concurrent compression operations allowed per SP.

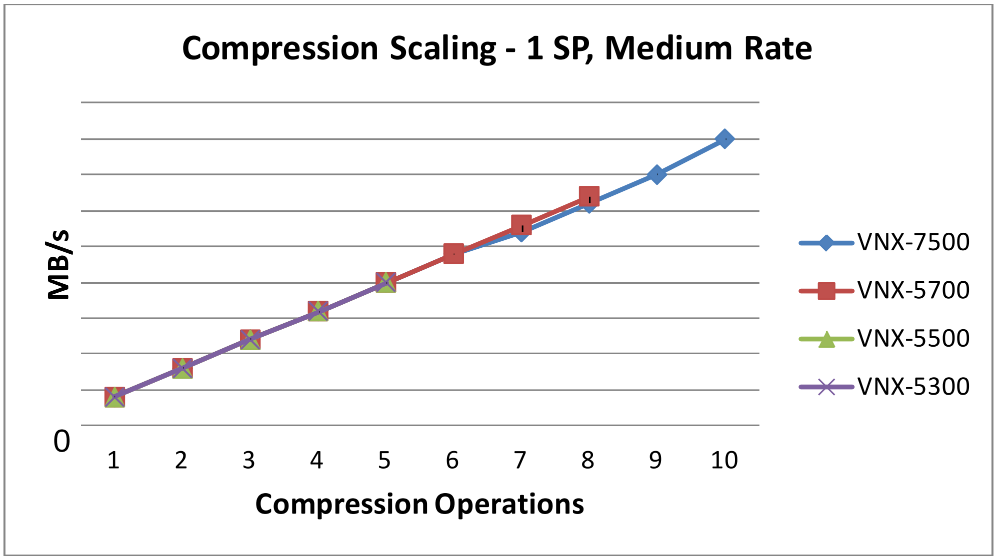


Figure 15. Compression scaling

The difference in throughput with different compression rate settings for the VNX-5700 is shown in Figure 16. There is a larger difference between the Low and Medium rates than between Medium and High. This behavior is consistent across all models.

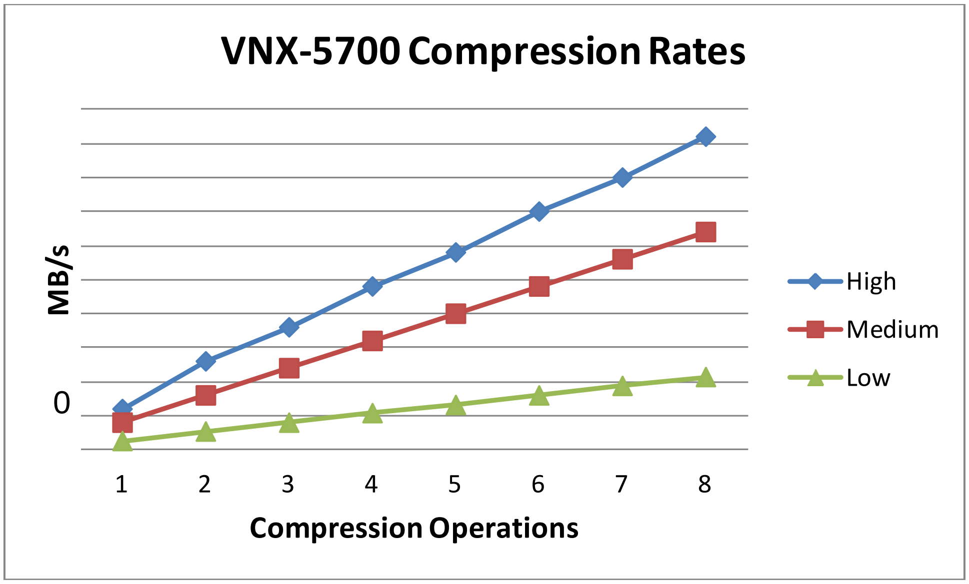


Figure 16. Compression rates

The compressibility of the data itself has little bearing on compression throughput. When data is highly compressible (8:1), compression throughput may be 10 percent lower than is the case when compressing moderately compressible data (1.5:1).Differences are only notable when approaching the max allowable compression operations with the rate set to High. The two cases are equal when the rate setting is Medium or Low.

Impact on server I/O can be moderate to high when compared to the performance of non-compressed thin LUNs. In some cases, like simulated file sharing environments, 25-50 percent lower throughput has been observed. In other cases, like large block, high bandwidth I/O, the impact can be much higher. The inline operations, inherent to reads and writes of compressed data, affect the performance of individual I/O threads; therefore, we do not recommend this for I/O-intensive or response-time-sensitive applications.

Compression's strength is improved capacity utilization. Therefore, compression is not recommended for active database or messaging systems, but it can successfully be applied to more static datasets like archives, clones of database and, messaging-system volumes. The best use cases are those where data needs to be stored most efficiently and with a high degree of availability.

Conclusion

VNX storage systems provide powerful capacity efficiency features that can improve effective capacity utilizations up to three times when compared to traditional storage devices. These capacity-optimization features are included with the VNX Operating Environment at no additional cost. The deduplication and compression features for file and block storage offer complementary capacity efficiency opportunities for all data types in the primary storage systems.

References

The following white papers are available on EMC.com:

- EMC Data Compression — A Detailed Review
- EMC VNX Virtual Provisioning – Applied Technology

The following documentation is available on Powerlink.emc.com:

- *Using VNX File Deduplication and Compression* technical module

Appendix A: NTFS file system conditioning with SDelete

Many file systems do not efficiently reuse the space associated with deleted files. When files are deleted from NTFS file systems, the deleted files' data continues to be stored in the file system until it is overwritten by new data. When files are frequently deleted, the free space in the file system may gradually become filled with deleted file data that is no longer accessible by the file system. Deleted file data reduces the effectiveness of EMC Data Compression. The retention of deleted file data is a characteristic of the file system and is relevant regardless of whether LUNs are presented directly to a Windows server or the Windows server resides in a VM.

Data Compression processes blocks associated with deleted files the same way that data is processed for valid files. Unused (never-used) blocks in the NTFS file system compress down to zero consumed capacity, but deleted file blocks are only as compressible as the deleted file data that continues to be stored on them.

The SDelete utility from Microsoft replaces deleted file blocks with zeros when it is invoked with the -c option. Note that deleted file blocks cannot be removed by defragmentation or reformatting the file system. The blocks zeroed by SDelete do not consume any capacity once they are processed by Data Compression. This can have a profound impact on compression results.

For example, let's assume a 100 GB file system has files that are 1.5:1 compressible (33 percent space savings). This file system resides on a 100 GB RAID Group LUN. Also assume that the file system reports used space to be 60 GB, but since the file system has been in use for some time, there are actually 90 GB of used blocks in the file system. This means that only 10 GB of the file system capacity is unallocated to data.

Without SDelete, the resulting compressed LUN will consume roughly 60 GB in the pool, since all 90 GB worth of blocks is compressible at a 1.5:1 ratio. Thin LUN metadata may add another 3-4 GB of consumed capacity. The 10 GB of unallocated data does not consume any capacity once it is compressed. This results in overall capacity savings of roughly 40 GB.

If SDelete is run on the file system prior to being compressed, Data Compression is more effective. There is still 60 GB of data, but the extra 30 GB of deleted file blocks and the 10 GB of unallocated capacity are overwritten by zeros by SDelete. In this case, the compressed LUN only consumes 40 GB in the pool for a total savings of 60 GB. The data itself compressed at a ratio of 1.5:1, but the zeroed capacity does not consume any space in the compressed LUN. In this example, using SDelete yielded an additional 50 percent of space savings.

SDelete writes to all space not consumed by a valid file data. If the file system resides on a thin LUN, SDelete causes the thin LUN to become fully consumed. Therefore, users must be sure they have adequate pool capacity if they choose to run SDelete on a thin LUN. On RAID Group LUNs or thick LUNs hosting NTFS file systems, run SDelete prior to enabling compression to maximize space savings. Additional information, including the download of the utility, can be found at: http://technet.microsoft.com/en-us/sysinternals/bb897443.aspx.

Appendix B: Compression states

Table 4. LUN compression states

State	Description	Valid for LUN type
Initializing	Compression is first enabled, and background setup operations are performed by the system.	All
Compressing	The thin LUN or thick LUN is being compressed.	Pool LUNs
Compressed	An initial or subsequent compression has completed.	Pool LUNs
Compression Queued	The state of compression prior to moving to Compressing. It is in this state for a short duration unless system limits for concurrent operations are exceeded.	Pool LUNs
Compression Paused	The compression operation has been paused. I/O continues to the LUN. New data is not compressed; compressed data remains compressed.	Pool LUNs
Compression Faulted	The compression operation did not complete successfully. More information may be available in State Details.	Pool LUNs
Decompressing	The LUN is being decompressed.	Thin LUNs
Decompression Queued	The LUN is waiting to be decompressed. It is usually in this state for a short duration unless system limits for concurrent operations are exceeded.	Thin LUNs
Decompression Paused	The decompression operation has been paused. I/O continues to the LUN. New data is not compressed; compressed data remains compressed.	Thin LUNs
Decompression Faulted	The decompression operation did not complete successfully. More information may be available in State Details.	Thin LUNs
System Paused	The system has paused a decompression operation due to lack of capacity in the pool. The user can either add capacity to the pool or migrate LUNs out of the pool to free up capacity.	Thin LUNs
Migrating	This is the state of a RAID group LUN during the initial compression. RAID group LUNs are migrated to a thin LUN, and compressed inline with the migration. This operation cannot be paused. It can be canceled.	RAID group LUNs
Migration Queued	This is the state of a LUN while it is waiting to be migrated. It is usually in this state for a short duration unless system limits for concurrent operations are exceeded.	RAID group LUNs

Migration Faulted	The migration operation did not complete successfully. More information may be available in State Details.	RAID group LUNs
Migration Paused	Available in the VNX™ Operating Environment (OE) for block release 5.32 and later. The migration operation for the LUN has been paused.	RAID group LUNs