Getting Started with IBM Real-time Compression on IBM Storwize V7000 Unified V1.4.0.1

Version 1.03

January 27, 2012

TABLE OF CONTENTS

Overview

Software version 1.4.0.1 of Storwize V7000 Unified introduces a new capability of Real-time Compression. This capability is made possible through the integration of the Random Access Compression Engine (RACE) technology. IBM's Real-time Compression delivers high-performance compression that supports active primary workloads. This innovative, easy-to-use compression is already publicly available in Storwize V7000 and SAN Volume Controller systems (running version 6.4) and is now fully integrated into the Storwize V7000 Unified system.

Real-time Compression typically delivers 50% or better compression for data that is not already compressed, and helps reduce:

- Storage purchase costs
- Rack space
- Cooling and energy
- Software costs for additional functions

IBM Real-time Compression increases the capacity of existing storage infrastructure, helping enterprises meet the demands of rapid data growth while enhancing storage efficiency. IBM's Real-time Compression uses IBM's patented real-time data compression techniques to primary storage, delivering optimization and savings in Storwize V7000 and SAN Volume Controller, for both internal as well as external/virtualized storage. The result is exceptional cost savings, ROI, operational and environmental efficiencies.

Introduction

IBM recently released Storwize V7000 and SAN Volume Controller version 6.4 that adds support for compressed volumes. Compressed volumes are another type of thinly provisioned volumes. Compressed volumes use IBM's Random Access Compression Engine (RACE) technology to deliver Real-time Compression capabilities for block storage environments in SAN Volume Controller and Storwize V7000 products. Version 1.4 of Storwize V7000 Unified adds support for creating file-systems that use Real-time Compression. File-system compression is delivered using the block compression capability of Storwize V7000 version 6.4 and operates at the volume level.

In implementing this unique IBM storage efficiency in unified storage environments, users will get a good understanding of how this technology can help improve their unified storage systems efficiency as well as operational efficiencies. Specifically, key areas to be observed are: compression efficiency, transparency in the environment, and application performance.

This document discusses the attributes of IBM's Real-time Compression for file-sharing environments in Storwize V7000 Unified. The document also provides technical information and guidelines on what should be considered when deploying compression in the unified storage environment.

Compression Efficiency and Data Types

The best candidates for data compression are data types that are not already compressed by nature. These data types range between many workloads and applications, such as databases, character/ASCII based data, server virtualization, Computer Aided Design (CAD) data, software development, engineering data, vector data, and other general-purpose enterprise applications and workloads.

In cases where the data is compressed by nature, little additional savings can be achieved by using storage system compression. It is recommended not to spend system resources on data that can't be compressed or achieves low compression rates. Workloads that create compressed data should not be stored on compressed filesystems. The following examples represent workloads and data types that are already compressed by nature:

- Compressed audio, video, and image file formats – files such as jpeg, png, mp3, medical imaging (DICOM), mpeg2, etc.
- Compressed user productivity file formats – MS-Office 2007 newer formats (pptx, docx, xlsx, etc.), pdf files, executable files (exe), etc.
- Compressed file formats – such as zip, gzip, rar, cab, tgz, etc.

In cases where data is encrypted by the client or application used, no savings can be achieved by using any compression method. Due to the high entropy found in encrypted data, compression algorithms cannot find repetitions or similarities within them, and are completely ineffective in such cases. Workloads that create encrypted data should not store their data in compressed filesystems. If, however, encryption in the storage system/environment operates at the back-end disk level and is performed post compression in the data path – the data can be compressed and will have the same encryption capabilities in the back-end.

Following is a high-level short list focused on workloads that are good candidates and list of workloads that should be avoided for compression:

- Common workloads suitable for compression
    - General-purpose – collaboration, home-directories, engineering, etc.
    - Databases – DB2, Oracle, MS-SQL, etc.
    - Server Virtualization – KVM, VMware, Hyper-V, etc.
    - Others – CAD, animation rendering, seismic, scientific, etc.
- Common workloads NOT suitable for compression
    - Workloads using pre-compressed data types such as video, images, audio, etc.
    - Workloads using application/network-based encryption
    - Other workloads using incompressible data or data with low compression rate

Choosing which application workloads are good candidates for compression can be determined by the nature of the data that is targeted to be stored in a compressed filesystem and the compression savings expected for this data. As a general rule of thumb, IBM recommends to compress data that gets more than 45% compression savings.

Compression Granularity

When using filesystem compression in Storwize V7000 Unified systems, the compression is performed in the storage pool level and more specifically in the volume (block-device) level – the underlying Storwize V7000 volumes that are used to store the filesystem are compressed. This means that compression is not performed for individual files, but rather in a filesystem storage pool level. In order to make the most efficient use of the architecture and optimize system resource usage, a selective compression method has to be used. This will prevent cases where incompressible files get compressed. This is achieved using a **Selectively Compressed Filesystem**. This type of file system is created from multiple filesystem pools. Some of the pools are compressed and others are not compressed. Files are placed in a compressed or non-compressed pool based on their file-name extensions. It is recommended to always use this kind of filesystem in order to avoid compression of file-types that are already compressed by their nature.

When creating a selectively compressed filesystem, the system will use file placement rules that indicate in which pool files should be stored. Storwize V7000 Unified placement rules are very unique and useful for placing data in the right pool upon the file creation. Placement policies are flexible and powerful in the sense that any attribute can be used to classify where data is stored. This can be used for many purposes to optimize data placement in the environment. When used for compression purposes, it is recommended to use placement rules to store non-compressible file types in non-compressed pools based on filename extension. The system is loaded with a pre-defined list of filename extensions that that will be excluded from compression and will be placed in a non-compressed storage pool. The default list contains file-name extensions that typically achieve poor compression rates. It is highly recommended to review this list and add filename extensions for incompressible or encrypted files types that are used in your environment and may not be included with the default list.

Application/Filesystem Usage Patterns

Understanding the nature of the data that is stored in a filesystem is very helpful in evaluating how suitable it is for compression. There are 2 types of filesystems to consider:

- Heterogeneous – filesystems used to store a variety of data types from different applications
- Homogeneous – filesystems used to store one (or very few) data types that belong to one application

Heterogeneous filesystems contain diverse content (data that was created by several different application workloads). Heterogeneous filesystems contain data of different types. Since different data types populate such filesystems, there are situations where incompressible or encrypted data will be stored on such filesystems. In such cases, system resources can end up being spent on data that can’t compress.

Heterogeneous filesystems are typically found in user productivity environments, such as user home-directories, where users typically store diverse types of files. In many situations compressed content can be found in such environments as mentioned in ‘Compression Efficiency and Data Types’ section above. In order to avoid attempting compression of data that is already compressed or encrypted it is recommended to use selectively compressed filesystems with heterogeneous filesystems, unless the heterogeneous filesystem contain only compressible, unencrypted data types.

Homogeneous filesystems contain data that was created by a single application, and store data of the same kind. The following examples represent application workloads that typically store their data on homogeneous filesystems:

- Databases
- Server virtualization

Homogenous filesystems are good candidates for compression since the majority of the data stored in such filesystems typically achieves similar compression rate across the filesystem. This is true as long as the homogeneous filesystem contains compressible data. If it contains incompressible data (already compressed by nature or encrypted) it will not compress well, even though the filesystem is homogeneous.

Although the data in the filesystem is homogenous and compressible it is still recommended to use a selectively compressed filesystem using placement policy. This is because in reality sometimes users or administrators may use such filesystems to store other data types (temporarily or persistently) that may be incompressible. In order to avoid storing such non-compressible data in compressed storage pools – it is always better to use a selectively compressed filesystem which will store such incompressible files (based on their filename extension) in a non-compressed storage pool within selectively compressed filesystems.

Review the following guidelines in order to achieve the highest compression savings in your environment:

- Avoid using application compression when using compressed filesystems. In most cases, data that is already compressed will not get significant additional savings from compressing it again.
- Avoid using application encryption when using compressed filesystems. Encrypted data is not compressible.

Storwize V7000 management interfaces can be used to get rich compression savings information that can be used to measure compression effectiveness in the environment. This information can be used to measure the savings achieved in your environment for workloads that use compressed storage pools and filesystems.

Data Compressibility

Choosing which application workloads are good candidates for compression is determined by the nature of the data that is targeted to be stored in a compressed file-system and the compression savings expected for this data. As a general rule of thumb, IBM recommends to compress data that gets more than 45% compression savings.

Data Compression Rate	Recommendation
Higher than 45% compression savings	Use Compression
25-45% compression savings	Assess workload with compression
Lower than 25% compression savings	Avoid Compression

If the data you want to store on compressed filesystems already exists in a different environment, you can use the following methods to estimate the expected compression rate of the data types generated by the application.

If the data currently exists on a filesystem running on a host (Linux, AIX/VIOS, HP-UX, Solaris, Windows, VMware-ESXi) you can use the **Comprestimator** utility to analyze the data by efficiently and quickly sampling the block-device used to store the filesystem on the host running it – this will provide you with the estimated compression rate for the mixture of data types stored in this filesystem. IBM provides the Comprestimator utility that analyzes the expected compression savings for an existing volume containing data considered for compression. Comprestimator is a command line host-based utility that can be used to estimate expected compression rates for existing block devices in their native environment. The utility uses advanced mathematical and statistical algorithms to perform the device sampling and analysis process efficiently and in a very short time period. The utility displays its accuracy level by showing the maximum error range of the expected compression results achieved, based on the formulas it uses. The utility runs on a host that has access to the devices that will be analyzed, and performs only read operations so it has no effect whatsoever on the data stored on the device. It is important to emphasize that since Comprestimator is a host-based utility that analyzes any block device accessible from the host it is running on, it can be used on any device, regardless of the storage system it is stored on. Please note that it is highly recommended to use Comprestimator to analyze volumes that contain as much data as possible rather than volumes that are mostly empty of data. This increases accuracy and reduces the risk of analyzing old data that was already deleted, but may still have traces on the device.

Depending on the environment configuration, in some cases Comprestimator will be used on more than one host, in order to analyze additional data types that are stored in volumes accessible from other hosts. A copy of Comprestimator can be downloaded from the Comprestimator Utility web page.

In some other cases the existing data may be stored on a proprietary NAS system (e.g. IBM N series). In these cases you typically do not have access to the block-device used to store the proprietary filesystem. In such cases you can use any compression utility (e.g. gzip, winzip) to compress sample application files based on their extensions and to get the expected compression savings for this file type. Comprestimator can also be used to analyze large files in some cases. This way large application files (such as database or server virtualization files) can be quickly analyzed for expected compression rate.

Additional Considerations

In addition to compression rate achieved for compressed workloads, there are other factors and metrics to be taken into consideration when choosing which data to compress on a system. These considerations include the following:

- Real-time Compression is supported for both internal storage and externally virtualized storage. Using the external virtualization capability of Storwize V7000, compression can be used with all storage systems supported for external virtualization.
- Compression is supported for a maximum of 200 compressed volumes per I/O group in this release – this limit applies only to compressed volumes; there is no restriction for the number of non-compressed volumes in the I/O group. A cluster with 4 I/O groups can support as many as 800 compressed volumes.
- System resources for using compression – before using compression on existing Storwize V7000 systems it is highly recommended to measure the current system utilization, in order to ensure it has the CPU resources required for compression.
  - If CPU utilization on the nodes in the I/O group is below 25%, then this I/O group is suitable for using compression.
  - If the CPU utilization of a node is sustained above 25% most of the time, then this I/O group may not be suitable for compression, as it is too busy. If your system is currently supported for the Non Disruptive Volume Move feature, then it is possible to move this volume to another I/O group that has the resources required for compression.
  - Note that system resource allocation for compression is performed upon the first compressed filesystem or volume is created, before it is used.
- Compressed volumes cannot be used with Easy Tier. Specifically, Easy Tier and compression can both work on the same Storwize V7000 system, but Easy Tier will not handle activity of compressed volumes.
- Application I/O patterns –
  - Real-time Compression and its RACE foundations were designed and purpose built for enterprise-level random-access application workloads. While exceling in such application workloads, IBM does not recommend using Real-time Compression for heavy sequential oriented workloads. The most common use-case for such workload is using Real-time Compression to compress volumes that are used as disk-based backup targets.
  - Workloads generating random-access I/O patterns are highly recommended for use with Real-time Compression.

Free Space Management

IBM chose the Random Access Compression Engine (RACE) technology for integrating compression into the highly efficient Storwize V7000 product line. The groundbreaking Real-time Compression capability was introduced in Storwize V7000 and SAN Volume Controller enterprise product lines with release 6.4. Integrating this groundbreaking block-device compression technology with the advanced file sharing and data protection capabilities of the Storwize V7000 Unified was a natural choice.

Compression in Storwize V7000 Unified is configured in the filesystem storage pool level. A filesystem can be composed of one or more storage pools. Each of these storage pools can be created as compressed or not compressed. The system supports a wide range of flexible configurations. It is important to mention that the filesystem's logical use of space is separate from the physical space allocated to store the data in the storage pool. When data is stored in a compressed storage pool within a filesystem – the original (uncompressed) data size is presented to users and applications accessing the data. The uncompressed data size is also used and presented in the filesystem utilization metrics. So, if a filesystem is configured to be 10TB size total and already contains 7TB of user data it is 70% used and has 3TB, which are 30%, of free file-space capacity in the filesystem. This information will be presented for filesystems regardless of whether they are using compressed storage pools or not. This contributes to complete compression transparency for users and applications accessing the compressed data.

Compressed storage pools are also thin-provisioned by nature. This means that the system will allocate physical capacity only when it is actually used to create and update files, rather than allocating all required capacity in advance. It is important to understand that physical space that was allocated for a file remains allocated even when the file is deleted. Upon file deletion – the filesystem marks the space free internally, but the volume is not aware of this file-level deletion – so the physical allocated space from the pool remains the same. This means that the system will have different utilization levels for the filesystem and the storage pools it uses. In addition to that, due to the unique architecture of GPFS it will not immediately re-use the space that became free when the file was deleted in order to store new data. This will reduce thin-provisioning savings over time, and the system will eventually save capacity from compression only.

Filesystem and Storage Pool Utilization

Space management on the Storwize V7000 Unified is very similar to standard filesystem space management policies and guidelines. It is important to maintain a filesystem size that provides the required capacity for applications to store their data. Managing this file-space capacity means you set the filesystem size according to the size of the data that will be stored in the filesystem by users and applications. This is the standard common practice regardless of the type of volumes or storage pools used. After sizing the filesystem according to user and application requirements, the system will configure the optimal size of compressed volumes out of the compressed filesystem pool. Monitoring filesystem free space metrics should be performed just like any other filesystem. The administrator of the system maintains the filesystem size. Each filesystem uses a threshold for displaying warning messages in situations where filesystem space is about to run out.

When creating a filesystem that uses compressed filesystem pools, the system will automatically create the compressed volumes out of the storage pool. The volumes will be provisioned using a virtual capacity size that is set by the system to support the filesystem size that was configured. The total virtual size of all volumes used by a filesystem should equal to the total size of the filesystem.

Compressed volumes are provisioned using extents from the storage pool that was used to create them. The storage pool free space has to be monitored to ensure that there is enough free space available in the pool to sustain the growth of actual allocated size of the compressed volumes. Compressed volumes will allocate extents from the storage pool based on their consumption by the filesystem. The Storwize V7000 Unified monitoring system displays and continuously monitors the percentage of the physical space that is allocated from a storage pool. Once the used percentage is greater than the threshold set for the pool, a warning alert will appear in the system. In order to reduce the percentage of space allocated from the volume and reduce the risk of running out of physical space the storage pool physical size has to be increased. Adding more disks to this pool reduces the percentage of utilization in this pool and brings it back below the threshold.

Planning and Provisioning Guidelines

Provisioning and physical space allocation of filesystems using compressed storage pools has to be understood in order to achieve the best results of compression in the environment. When a filesystem is created, its maximum size is set based on the user and application requirements, and following the typical provisioning methodologies for filesystems in the environment. Filesystems using compressed storage pools allocate the physical capacity only when the filesystem actually uses it (in a thin-provisioning manner). The size of allocated space is further reduced by the Real-time Compression technology.

After evaluating the expected compression rate for the data to be stored in a filesystem, planning of the virtual/physical storage pool configuration can begin. This procedure will assume a filesystem with a single compressed pool for storing data to simplify the description. Decide what’s the desired filesystem size for applications; this is the uncompressed size of the user and application data that is planned to be stored in this filesystem. This size is determined based on application space requirements and filesystem sizing practices in your environment regardless of compression.

This paragraph describes what you should consider when initially creating physical space for a compressed filesystem. It provides recommendations for initial sizing of physical storage for a filesystem depending on its size and the compression rate for the data it will store. Expected maximum physical allocation for this filesystem from the storage pool once data is stored on it can be calculated using the expected compression rate for the data and the filesystem size. For example, a 10TB filesystem storing data that is expected to get 60% compression savings will theoretically need only a 4TB storage pool. This can increase probability of thin-provisioned storage pools becoming utilized above recommended thresholds. This is not a recommended practice, because of the risk in running out of physical space due to the tight planning. This may seem very efficient for initial configuration of the system, but increases the maintenance levels that will be required later, since every time the filesystem size will have to grow for application purposes, new physical storage will have to be added to the compressed pool. This is why the recommendation is to use physical storage pools that are larger than the expected maximum physical allocation size (calculated based on filesystem size and expected compression rate). It is recommended to use the physical pool size that ranges between the following:

1. **Minimum** – increase the expected maximum physical allocation by 25%. This will keep the pool below 80% allocation as long as compression rate is same or higher than what was initially expected.
2. **Maximum** – create physical storage in the size of the file system, which is actually the size of original data before compression.

The optimal configuration depends on provisioning methodologies and actual compression rates in each environment. IBM recommends using the 2nd approach (maximum) for new users deploying Real-time Compression for the first time. It is also recommended to take this approach when the nature of the data to be stored with compression can’t be fully characterized/analyzed in advance, and expected compression rate may be inaccurate. Although choosing the 2nd approach may seem inefficient and counter-intuitive to compression and the savings it introduces – it greatly simplifies management and is very easy for new users experiencing with the technology. In addition, the physical space that may seem as unused can be eventually provisioned for other purposes or for future growth/expansion of the filesystem needed for application data growth. Savvy compression users that already gained experience with the technology and have more accurate compression rate expectations for their data, may configure less physical storage in advance. Such experience and knowledge is acquired over a short time through the use of compression with multiple application workloads and data types in a specific environment.

Proactive Monitoring

As mentioned before, free space is managed in the filesystem space and in the storage pool space independently. The file and storage pool utilization metrics are integrated into a single coherent view in the GUI. This view provides a consolidated view containing all information needed to monitor all space related information in the system. In addition a monitoring and notification system turns alerts on whenever thresholds of file space or storage pool space utilization thresholds are breached.

The most important utilization metric to monitor is the utilization of the physical space already used in a storage pool. This metric relates to the actual physical storage capacity already used for storing compressed data written to the pool. It is important to make sure physical allocation does not go over the recommended threshold (default is set to 80%). In order to reduce the utilization of the used space, the storage pool size has to increase. Adding more physical capacity to the pool reduces its utilization. Whenever a threshold is passed and an alert is turned on – the system will suggest corrective action for the specific scenario. If a corrective action to reduce utilization is not performed before the storage pool reaches 100% utilization, NSDs can go offline, which can cause the filesystem to go offline.

File space utilization is also very important to monitor in order to void situations where filesystems run out of space and reach 100% utilization. This metric relates to the file space original uncompressed file size as observed by users and applications. Either deleting files from the filesystem, or extending the size of the filesystem can address this. This will create new NSDs from the unused physical space in the compressed pool. The system uses the same threshold and alerting system and suggests corrective actions when thresholds are breached. If a corrective action to reduce utilization is not performed before the filesystem reaches 100% utilization, the filesystem will work in 'read-only' mode. Read requests will be serviced, but any write/update activity will fail.

Over time, after data has been fully populated into filesystems and natural data growth starts to occur, increasing filesystems sizes and other provisioning activities will benefit from Real-time Compression. In order to take full advantage of compression benefits in the system when a system is already at a high utilization of physical space used. In these situations physical storage pools need to be over provisioned. This means that the total virtual size of all volumes in a pool can be greater than the actual physical pool size. The compression rate achieved for the data stored on these volumes is reduced by compression and physical space is allocated accordingly. Please review the following example – a 10TB filesystem is created from a storage pool that has 10TB of space. In this example a volume in the size of 10TB is used to store this filesystem. The data stored in this filesystem gets an average of 60% compression savings. Once the filesystem is full with 10TB of data that gets 60% compression savings, it has actually allocated only 4TB of physical space from the pool to store the compressed data, although the virtual volume size is 10TB. In this case in order to utilize the remaining 6TB that are not allocated yet, additional virtual capacity (beyond 10TB) should be configured for volumes in this pool.

The example is used to explain that in order to be able to use all the physical space in the pool eventually, the virtual capacity provisioned from this 10TB physical pool for data that achieved 60% compression savings will be 25TB. Once these virtual 25TB are full with data that compresses at 60% the entire physical capacity of 10TB will be used from the volume. This scenario is used to explain how over provisioning should be used with compression in order to use all the physical space in a pool. In reality you will never want to use up the entire pool and you will always have to keep a certain amount of unallocated contingency capacity in the pool. In general, using up to 80% of a pool is a good practice that utilizes the physical capacity and takes advantage of compression benefits. Also, in reality, data will not always achieve the exact same compression rate. Data is dynamic and may change during its lifespan. It is recommended to have the contingency capacity always unallocated and available for such cases where compression rate may change due to application data changes. This will protect the environment from cases where compression rate may deteriorate (due to incompressible or lower-compression data) and there is no free space left in the pool to support it. So following these guidelines, if your data achieves 60% savings, you have 10TB of physical space which you want to leave at 80% physical allocation maximum, you will only plan to use 8TB out of the physical 10TB available in the pool. With 60% compression savings 20TB of uncompressed user data can be stored on 8TB of physical space. Calculating the recommended over provisioned virtual size for a pool can be done in the following way:

- Compression savings = *CR*; e.g.: 0.6 for 60% as in the example above
- Physical capacity (in TB) = *PC*; e.g.: 10 as in the example above
- Contingency threshold = *CT*; e.g.: 0.8 for 80% as in the example above
- Recommended maximum virtual capacity (in TB) = *(CT * PC) * (1 / (1 – CR))*

Test Recommendations

The Storwize V7000 Unified software release 1.4 beta program enables users to test their specific application workloads and data for compression efficiency and performance. In order to get results that can be later used to model the design of the deployment into production environments, it is highly recommended to test the system with workloads and data sets that are identical (or at least as similar as possible) to the real production environment, in terms of data and workload.

In order to test compression and performance effect on any data set or workload, a baseline has to be created. It is recommended to create a baseline with a non-compressed filesystem. Since this beta program is running on a test system, the workload data set has to be copied from the production system that is used to currently store it. Once the production data set is copied to the V7000 non-compressed filesystem, the baseline environment is ready. The total capacity required to store the data when it is not compressed can be collected now, and the workload that exercises this data set can now be generated. Each production workload has its own specific I/O patterns and loads. So it is highly recommended to use a workload that is identical to the production environment.

It is important to capture application performance metrics during these tests. Application performance results represent the user-experience, or application run-time, which takes into account all environment components. These results can later be used to estimate which workloads are appropriate for the system.

After the baseline tests are completed and results are collected, the data set now has to be copied to a compressed filesystem so it can be used for generating the same production workload again. IBM recommends testing **selectively compressed filesystems** to estimate the savings and performance of the compressed filesystem for the environment used. Now that the compressed filesystem contains the data set, compression savings can be reviewed and analyzed using the V7000 Unified management interface. After collecting compression savings information, the workload that exercises the compressed data set can now be generated.

Testing Filesystem and Storage Pool Utilization

In order to assess the user experience in terms of compression savings and managing space for compressed filesystems, IBM recommends generating the following scenarios for both types of compressed filesystems (in addition to other scenarios a customer may want to assess). It is also strongly recommended to evaluate different storage pool size configurations as described in the "Planning and Provisioning Guidelines" section (minimum/maximum/other):

1. Fill a compressed filesystem up and make sure you are able to reduce the utilization level.

   a. Test deletion of files to reduce file-space utilization
   b. Test extending the filesystem size by using more storage from the pool
2. Have a compressed filesystem partially used (e.g.: 30% used). Then delete all files in it and partially populate it by copying data to the filesystem again. Repeat the deletion and copying of new data into the filesystem a few times:
   a. Track compression rate achieved for the filesystem throughout this process and confirm this meets the compression savings expected for this data
3. Generate a scenario where the storage pool goes beyond recommend utilization threshold. When the system is in this situation, make the corrective actions suggested by the system and confirm it reduces the utilization level below threshold.

In general, it is highly recommended to reproduce space management related situations that were previously experienced in the environment with the V7000 Unified system or other NAS/file-sharing systems. Please track the information provided in the GUI while experiencing these situations. This will help gauge its efficiency in alerting the user about the situation as well as determine how helpful it is in recommending corrective actions.

A Note about Synthetic Benchmark Tools

Using real-life workloads when testing a system’s performance provides the most useful and valuable information. This is especially true when testing compression and other storage efficiency technologies. Tested workload generates I/O patterns that characterize the real application behavior, and the data stored and accessed looks like the real data that the system is highly likely to handle when it is used in production environments later.

Another way to test performance is to use synthetic test tools. These tools can be used for testing the limits of traditional storage systems or volumes. When using such tools it is important to remember that these tools typically generate synthetic I/O patterns using synthetic data sets. Since compression and other storage efficiency technologies are influenced by the nature of the data being optimized and how it is being accessed, synthetic benchmark results can easily provide the wrong impression for expected performance. This can make such results inaccurate, and in some cases even practically useless when trying to measure real-life application environment behavior. If you plan to use such tools, please consider their following characteristics:

- The data used in the benchmark – benchmarks typically use patterns that vary between being “all zeros” to being “completely random”. Production workloads generate content with varying data patterns that represent the information contained in the data. Using synthetic data will impact compression rates that can subsequently cause performance change in many cases.
- I/O access patterns generated – benchmarks typically generate a mix of synthetic I/O patterns that uses sequential/random reads/writes. Production workloads generate I/O patterns that are related to the data they store and access. In many cases the order in which data is read is identical or similar to the way it was written to storage. IBM’s RACE technology uses time-based compression, which takes advantage of the similar read/write order. This creates both compression and performance efficiencies, that can’t

be tested when using synthetic I/O patterns. This cannot be achieved with application simulators as well, since they just generate synthetic I/Os that are completely unrelated to the data stored in the volumes they access.

- Hot spots – benchmarks sometimes generate workloads that are not realistically spread across application volumes. So a benchmark can create synthetic hotspots on small range of data. On the other hand, if there are no hot spots at all and the test accesses the entire range of offsets across a volume, it will not represent a real environment as well. Using production workloads create real hot spots rather than synthetic ones, allowing the system cache to be more efficient.

Step-by-step Instructions to Get-Started with Compression

The following procedure guides you through the process of how to configure your Storwize V7000 Unified system to examine compression in your storage environment. The procedure describes the configuration and evaluation of a single compressed filesystem.

Step 1 – Estimate expected compression rates for existing data

As described earlier, it is important to create the file system so that it is large enough to contain all the data that will be written to it. For existing data that will be copied to a new compressed file system, it is necessary to review the compressibility of this data and use this information to choose the best file system size.

Use the Comprestimator tool to analyze the estimated compression rate for volumes you plan to compress. A copy of Comprestimator can be downloaded from IBM's technical support website 'Fix Central' – StorageDisk-2076-Comprestimator_Utility_v1.2.01 .

Comprestimator is a command line host-based utility that can be used to estimate expected compression rate for block-devices. The utility uses advanced mathematical and statistical algorithms to perform the sampling and analysis process in a very short and efficient way. The utility runs on a host that has access to the devices that will be analyzed, and performs only read operations so it has no effect whatsoever on the data stored on the device.

After installing Comprestimator on a host that has access to the block-devices/LUNs/volumes you plan to compress. Following that, use Comprestimator to analyze the device for expected compression. It is highly recommended to use Comprestimator to analyze volumes that contain as much data as possible rather than volumes that are mostly empty of data. This increases accuracy and reduces risks to analyze old data that was already deleted, but may still have traces on the device.

The following example shows how to run the tool and analyze a device ('/dev/sdb'). Compresstimator can be used with the '-h' flag for all other available options.

```
comprestimator -d /dev/sdb
```

Review the output generated by Comprestimator for the volume. Depending on the volume type you are analyzing choose one of the following values from the output as expected compression rate for the volume:

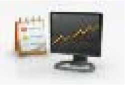
- When analyzing an existing thinly provisioned volume/block-device, use the *‘Compression Savings (%)’* value in Comprestimator output.
- When analyzing an existing fully allocated volume/block-device that was zeroed before it was used to store the current data on the volume, use the *‘Compression Savings (%)’* value in Comprestimator output.
- When analyzing an existing fully allocated volume/block-device that was **not** zeroed before it was used to store the current data on the volume, use the *‘Total Savings (%)’* value in Comprestimator output.

Verify the expected compression rates for this volume with the chart in Data Compressibility section to decide whether this volume is a good candidate for compression.

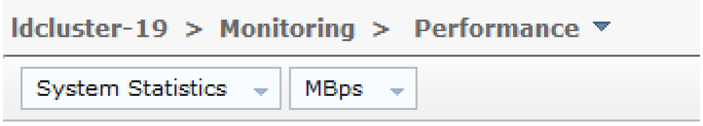
Step 2 –Assess the Suitability of I/O Group for Compression

Confirm that the CPU utilization in I/O group X is at an appropriate level.

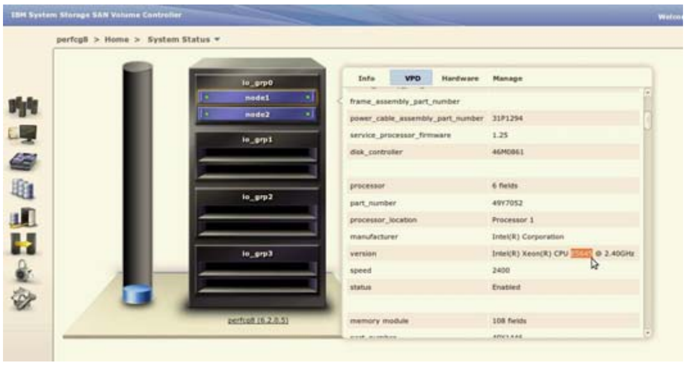
1. Log in to the GUI.
2. Open Monitoring -> Performance.

   Note: Monitoring Icon looks like this 

3. Switch between selecting node S and node T from the dropdown box in the upper left corner of the GUI. This dropdown box will select “System Statistics” by default.



4. Monitor the CPU utilization over busy periods of the day or the week.
5. Only if you are using SAN Volume Controller 2145-CG8 nodes, you should first find out what is the number of CPU cores in your system. To determine if your system uses 4 or 6 CPU cores:
   a. Select Monitoring > System to view VPD information related to the processor. The version entry for 2145-CG8 nodes contains one of the following two values:
      i. Intel® Xeon® CPU E5630 - 4 cores
      ii. Intel® Xeon® CPU E5645 - 6 cores



frame_assembly_part_number	
power_cable_assembly_part_number	31P1294
service_processor_firmware	1.25
disk_controller	46M0861
processor	6 fields
part_number	49Y7052
processor_location	Processor 1
manufacturer	Intel(R) Corporation
version	Intel(R) Xeon(R) CPU [illegible] @ 2.40GHz
speed	2400
status	Enabled
memory module	108 fields

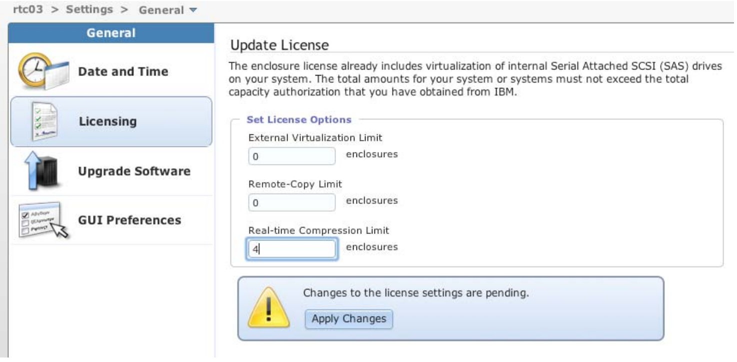
6. Verify if your system is suitable for using compression based on your system's CPU utilization:

   - If CPU utilization on the nodes in the I/O group is below 25%, then this I/O group is suitable for using compression.
   - If the CPU utilization of a node is sustained above 25% most of the time, then this I/O group may not be suitable for compression, as it is too busy.

   Note: It is also possible to perform this analysis using performance metrics in TPC. Simply draw the CPU % graph in the StorageSubsystemPerformance ByNode view from TPC and observing the CPU utilization over a longer period of time.

Step 3 – Enable Real-time Compression license

In order to enable the license, enter the total number of enclosures in your system. Select "Apply Changes" to complete the process.



Update License

The enclosure license already includes virtualization of internal Serial Attached SCSI (SAS) drives on your system. The total amounts for your system or systems must not exceed the total capacity authorization that you have obtained from IBM.

Set License Options

External Virtualization Limit: 0 enclosures

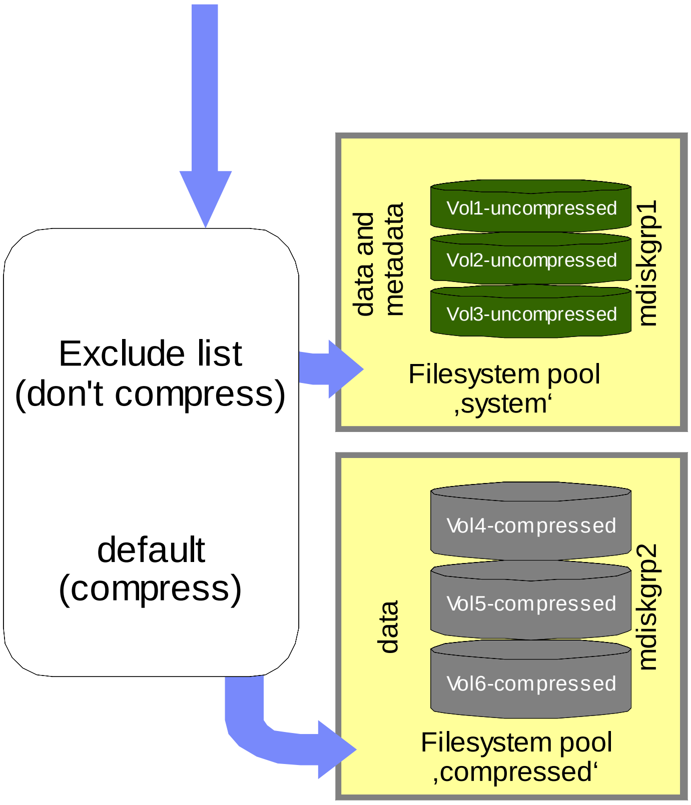
Remote-Copy Limit: 0 enclosures

Real-time Compression Limit: 4 enclosures

Changes to the license settings are pending.

Step 4 – Creating a selectively compressed file system

As mentioned in the Compression Granularity section above, a compressed file system is created with two filesystem pools. One for compressible files – configured with compression enabled, and one for compressible files – configured with compression disabled.
A filesystem placement policy is configured to place the compressible files in the compressed filesystem pool. The policy is based on a list of file extensions of compressed file types, that are defined as an exclude list.



Note that there capacity has to be managed separately for each pool, in other words, if many compressed files are copied to a compressed filesystem pool, it may reach a full capacity state. New file creations are not be automatically routed to the non-compressed filesystem pool.

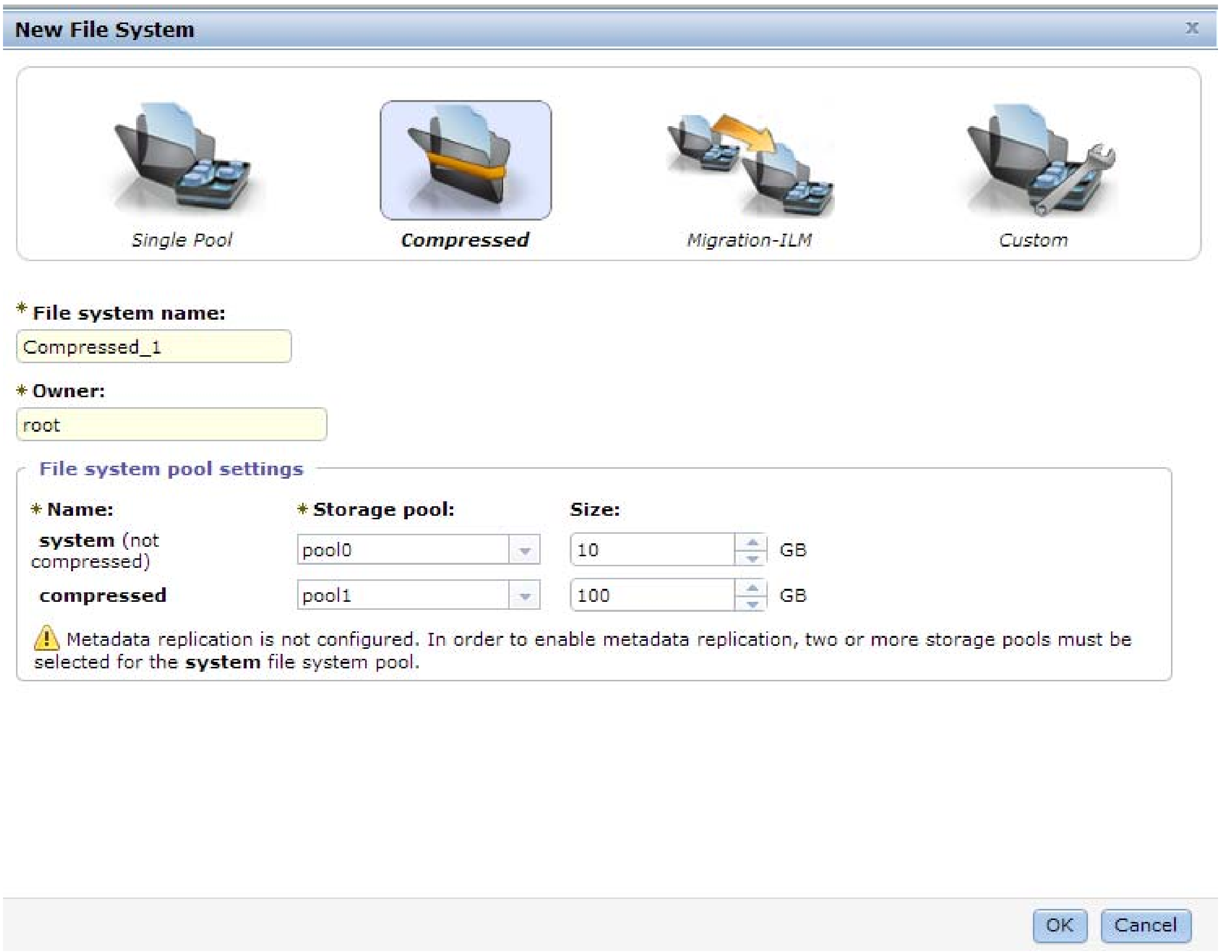
Note: We recommend that you create or identify storage pools for compressed volume data and uncompressed metadata. The storage pool for GPFS metadata needs to be 5-10% of the needed file system capacity.

1. Open the files panel: **Files-→File Systems**

Note – The Files icon looks like this:

2. Click on New File System at the top of the table and select the Compressed option:

3. Fill out as shown here:



* File system name: Compressed_1

* Owner: root

File system pool settings

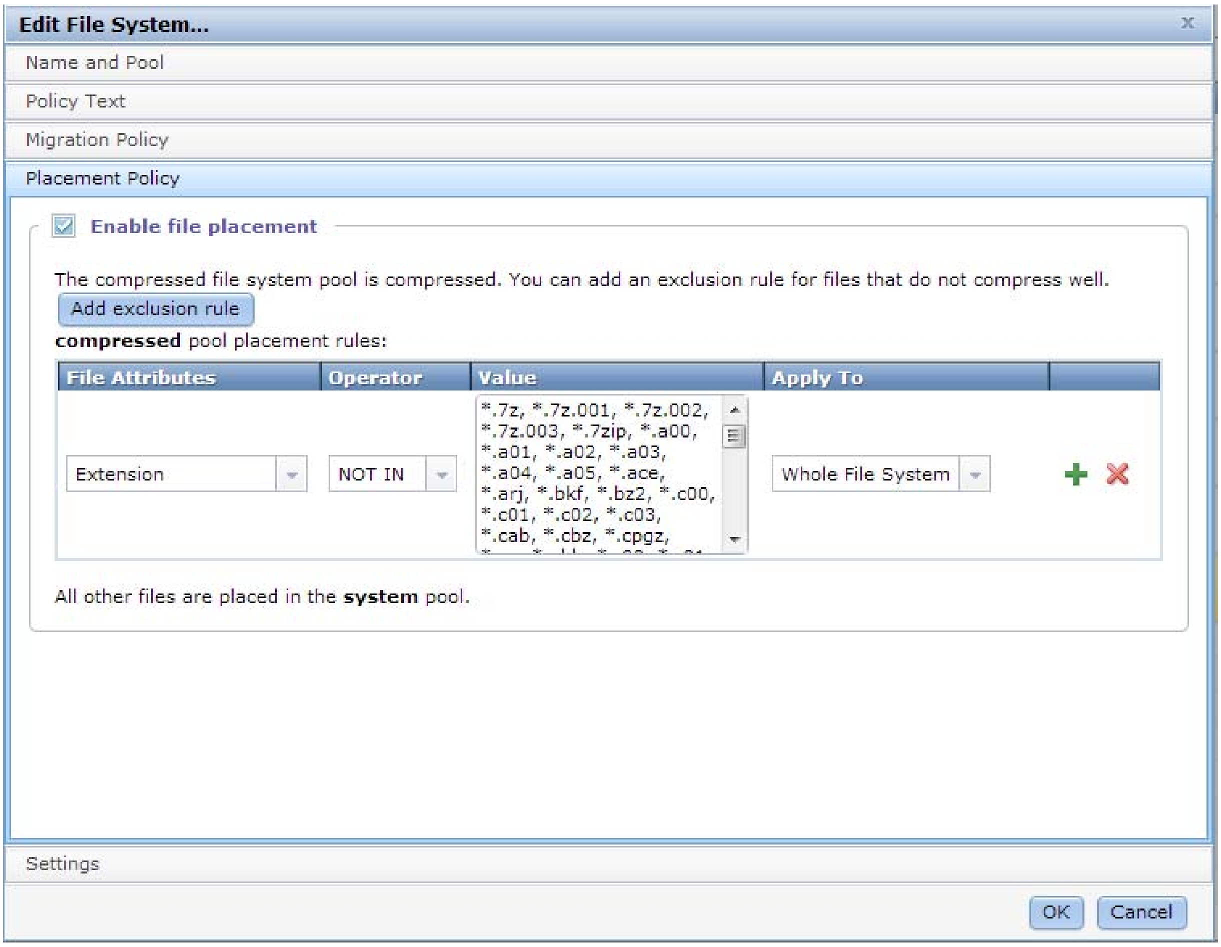
* Name:	* Storage pool:	Size:
system (not compressed)	pool0	10 GB
compressed	pool1	100 GB

⚠ Metadata replication is not configured. In order to enable metadata replication, two or more storage pools must be selected for the **system** file system pool.

1. Filesystem name: <name>
2. Owner: root
3. Set the pool sizes.
4. The ‘system’ pool is used for storing the filesystem’s metadata and the non-compressed files; the ‘compressed’ pool is used for storing the compressed files.

You need to set a different storage pool for each filesystem pool.

Reviewing and modifying the filesystem placement policy



☑ Enable file placement

The compressed file system pool is compressed. You can add an exclusion rule for files that do not compress well.

compressed pool placement rules:

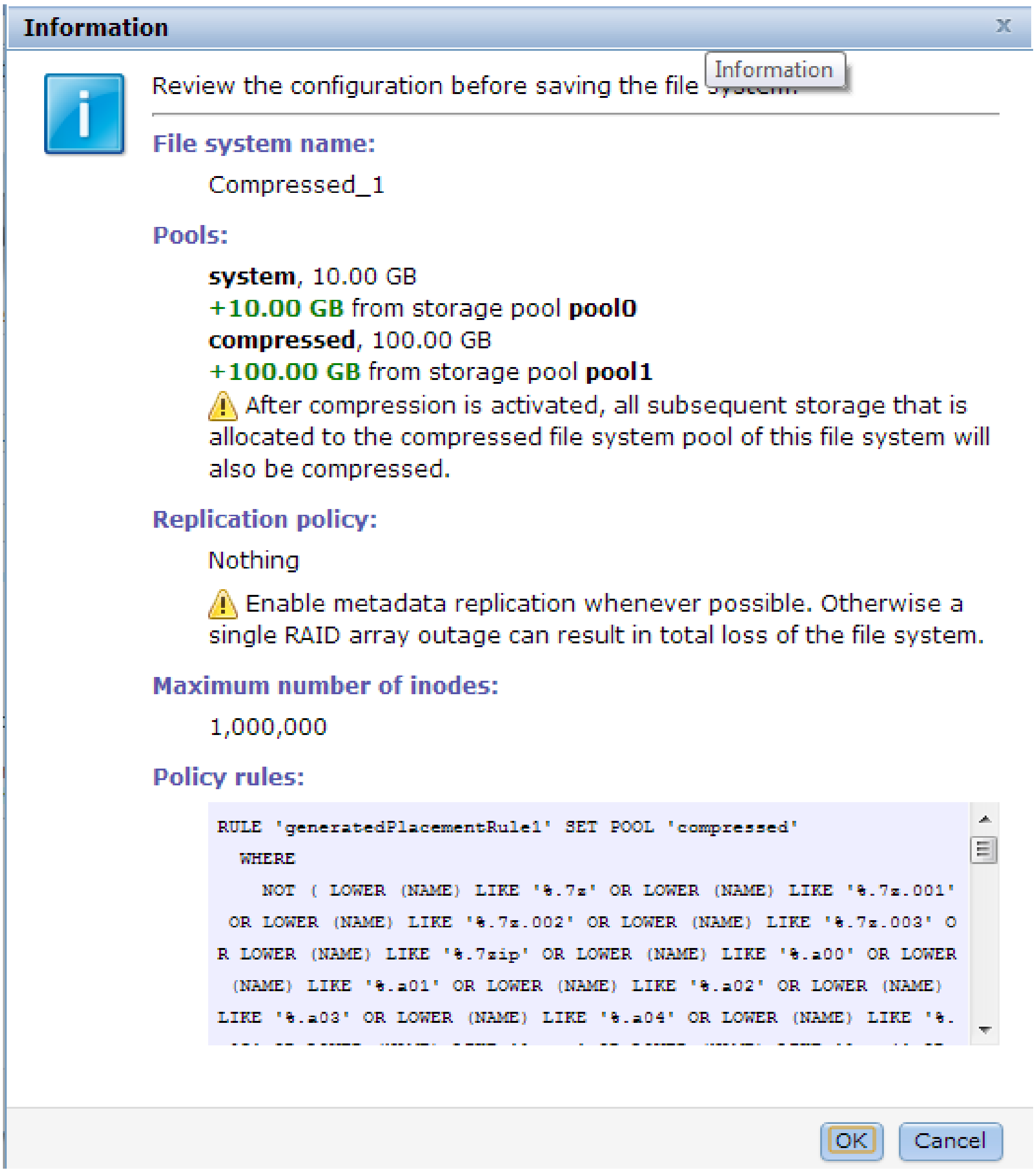
File Attributes	Operator	Value	Apply To	
Extension	NOT IN	*.7z, *.7z.001, *.7z.002, *.7z.003, *.7zip, *.a00, *.a01, *.a02, *.a03, *.a04, *.a05, *.ace, *.arj, *.bkf, *.bz2, *.c00, *.c01, *.c02, *.c03, *.cab, *.cbz, *.cpgz,	Whole File System	✚ ✖

All other files are placed in the **system** pool.

1. Select the new file system→properties→edit
2. Select the placement policy section
3. Review all filename extensions and add other file types if needed.
   Following is the default list of filename extensions that will be stored in the non-compressed pool:

   1st, 3dr, 3g2, 3gp, 7z.001, 7z.002, 7z.003, 7zip, a00, a01, a02, a03, a04, a05, aac, abr, abw, ace, acsm, ai, air, apk, application, arj, asx, atom, avi, azw, bbb, bc!, bik, bin, bkf, bz2, c00, c01, c02, c03, cab, cbr, cbz, cda, contact, cpgz, cpt, crw, crx, cso, daa, dao, dcr, dic, djvu, dlc, dmg, docm, docx, dot, dotm, dotx, dt2, dvf, eml, emz, epub, exe, fla, flac, flv, fnt, fpx, gif, gp5, gpx, gz, hqx, htm, icon, idx, img, ipa, iso, isz, jad, jar, jpeg, jpg, keynote, lng, logic, m2t, m2ts, m3u8, m4a, m4b, m4p, m4v, mdi, menc, mkv, mobi, mov, mp3, mp4, mpg, mrw, msi, mswmm, mts, nba, nbh, nbu, nco, ncs, nvram, odg, odt, ofx, ogg, one, onepkg, opml, otf, pages, pak, part1.exe, pcd, pcm, pdf, pmd, png, potx, pps, ppsx, ppt, pptm, pptx, prg, prproj, pub, pwi, pzl, r00, r01, r02, r03, r04, r05, r06, r07, r08, r09, r10, rar, rem, rmvb, rpm, rss, safariextz, sc2replay, scr, sisx, sit, sitx, sng, svp, swf, swp, sxw, tao, tar, tar.gz, tga, tgz, theme, themepack, thm, thmx, tif, tiff, toast, tod, torrent, ts, u3d, uax, uif, upg, vmsd, vmxf, vob, vswp, wav, wba, wbcat, wbmp, wma, wmv, wv, xhtml, xlsm, xlsx, xpi, z01, z02, z03, z04, z05
4.

.



Review the configuration before saving the file system.

File system name:

Compressed_1

Pools:

system, 10.00 GB
+10.00 GB from storage pool **pool0**
compressed, 100.00 GB
+100.00 GB from storage pool **pool1**
⚠ After compression is activated, all subsequent storage that is allocated to the compressed file system pool of this file system will also be compressed.

Replication policy:

Nothing

⚠ Enable metadata replication whenever possible. Otherwise a single RAID array outage can result in total loss of the file system.

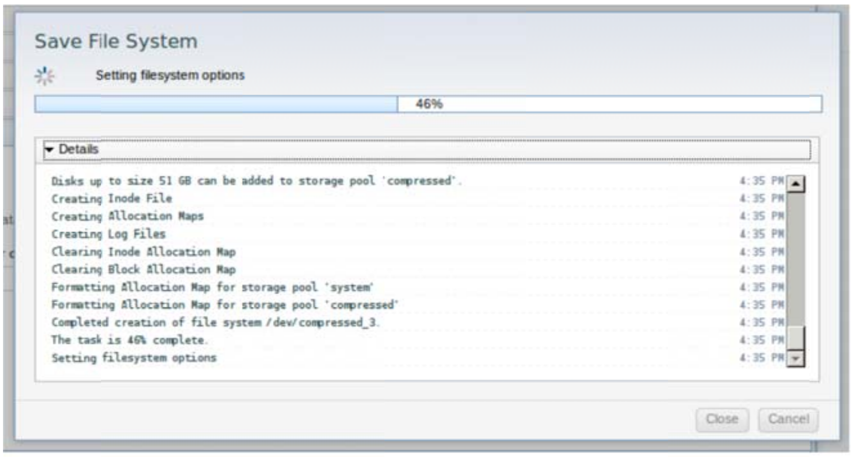
Maximum number of inodes:

1,000,000

Policy rules:

```
RULE 'generatedPlacementRule1' SET POOL 'compressed'
  WHERE
    NOT ( LOWER (NAME) LIKE '%.7z' OR LOWER (NAME) LIKE '%.7z.001'
 OR LOWER (NAME) LIKE '%.7z.002' OR LOWER (NAME) LIKE '%.7z.003' O
R LOWER (NAME) LIKE '%.7zip' OR LOWER (NAME) LIKE '%.a00' OR LOWER
 (NAME) LIKE '%.a01' OR LOWER (NAME) LIKE '%.a02' OR LOWER (NAME)
LIKE '%.a03' OR LOWER (NAME) LIKE '%.a04' OR LOWER (NAME) LIKE '%.
```

5. Click OK to create the file system. The progress dialog will display which CLI commands are executed.



Save File System

Setting filesystem options

46%

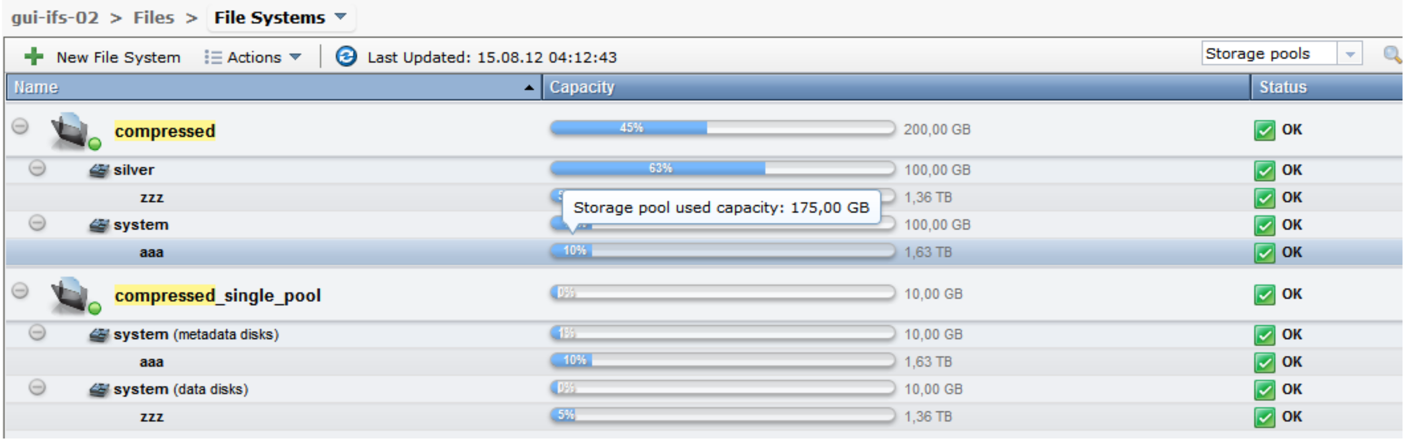
▾ Details

```
Disks up to size 51 GB can be added to storage pool 'compressed'.    4:35 PM
Creating Inode File                                                  4:35 PM
Creating Allocation Maps                                             4:35 PM
Creating Log Files                                                   4:35 PM
Clearing Inode Allocation Map                                        4:35 PM
Clearing Block Allocation Map                                        4:35 PM
Formatting Allocation Map for storage pool 'system'                  4:35 PM
Formatting Allocation Map for storage pool 'compressed'              4:35 PM
Completed creation of file system /dev/compressed_3.                 4:35 PM
The task is 46% complete.                                            4:35 PM
Setting filesystem options                                           4:35 PM
```

Step 5 – Monitoring and Resizing a File System

How to monitor file system use of space?

1. Open the files panel: **Files-→File Systems**



Last Updated: 15.08.12 04:12:43

Name	Capacity		Status
compressed	45%	200,00 GB	✓ OK
silver	63%	100,00 GB	✓ OK
zzz	5	1,36 TB	✓ OK
system	[illegible]	100,00 GB	✓ OK
aaa	10%	1,63 TB	✓ OK
compressed_single_pool	0%	10,00 GB	✓ OK
system (metadata disks)	1%	10,00 GB	✓ OK
aaa	10%	1,63 TB	✓ OK
system (data disks)	0%	10,00 GB	✓ OK
zzz	5%	1,36 TB	✓ OK

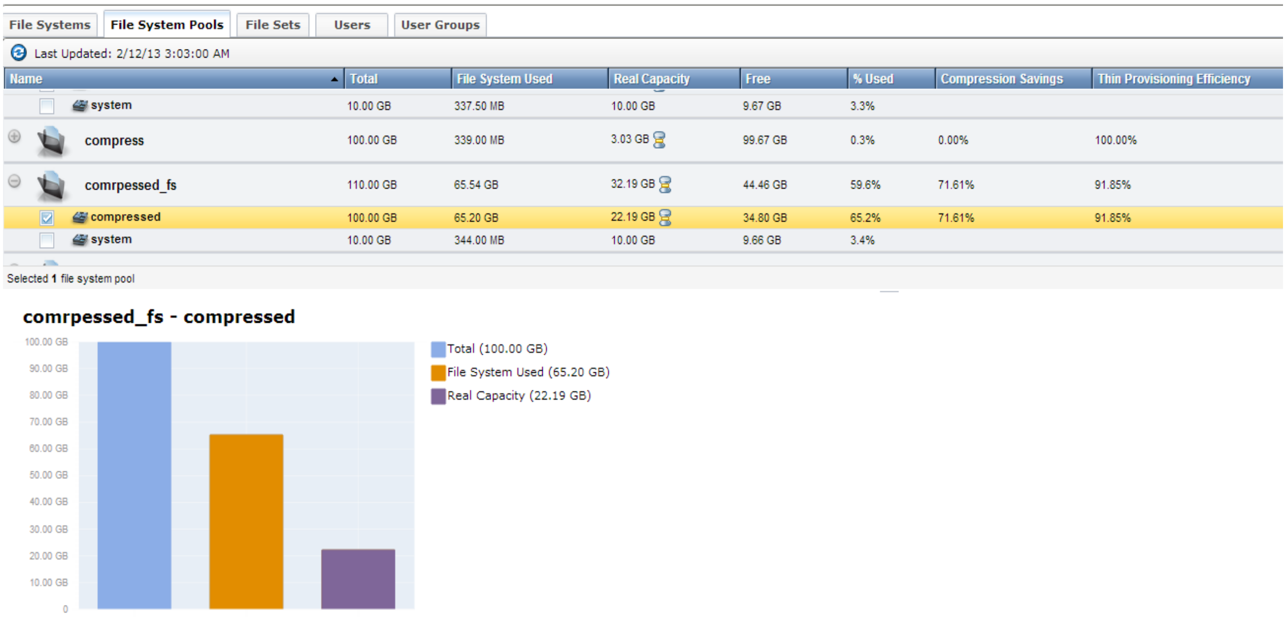
Storage pool used capacity: 175,00 GB

This panel provides the following information:

- File system capacity (the upper bar) – the capacity of the file system itself – i.e. the user's view.
  In the example above, the "compressed" file system is 45% full out of its 200GB total size.
- File system pool capacity (the bar next to the storage pool name) – the allocated and used capacity of the file system pool.
  In the example above, the "compressed" file system consists of the compressed data file system pool named "silver", total size 100GB, currently allocated 63%, and the metadata file system pool named "system"

File and block warning levels are shown in this panel as well.

2. Open the Capacity panel: **Monitoring-→Capacity**



Last Updated: 2/12/13 3:03:00 AM

Name	Total	File System Used	Real Capacity	Free	% Used	Compression Savings	Thin Provisioning Efficiency
□ system	10.00 GB	337.50 MB	10.00 GB	9.67 GB	3.3%		
compress	100.00 GB	339.00 MB	3.03 GB	99.67 GB	0.3%	0.00%	100.00%
comrpessed_fs	110.00 GB	65.54 GB	32.19 GB	44.46 GB	59.6%	71.61%	91.85%
☑ compressed	100.00 GB	65.20 GB	22.19 GB	34.80 GB	65.2%	71.61%	91.85%
□ system	10.00 GB	344.00 MB	10.00 GB	9.66 GB	3.4%		

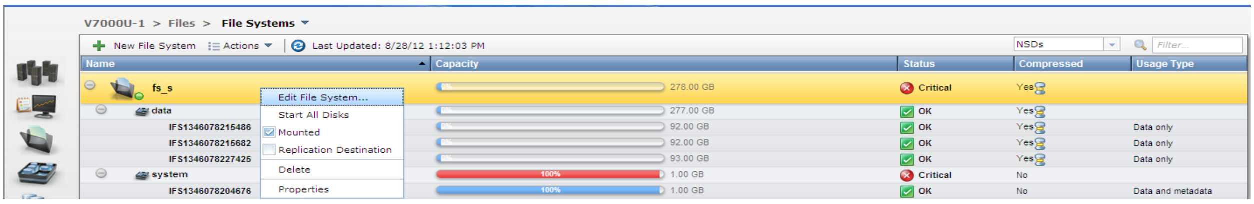
Selected 1 file system pool

comrpessed_fs - compressed

This panel provides a deeper insight into how well compression works.

How to resize a filesystem and therefore automatically add volumes?

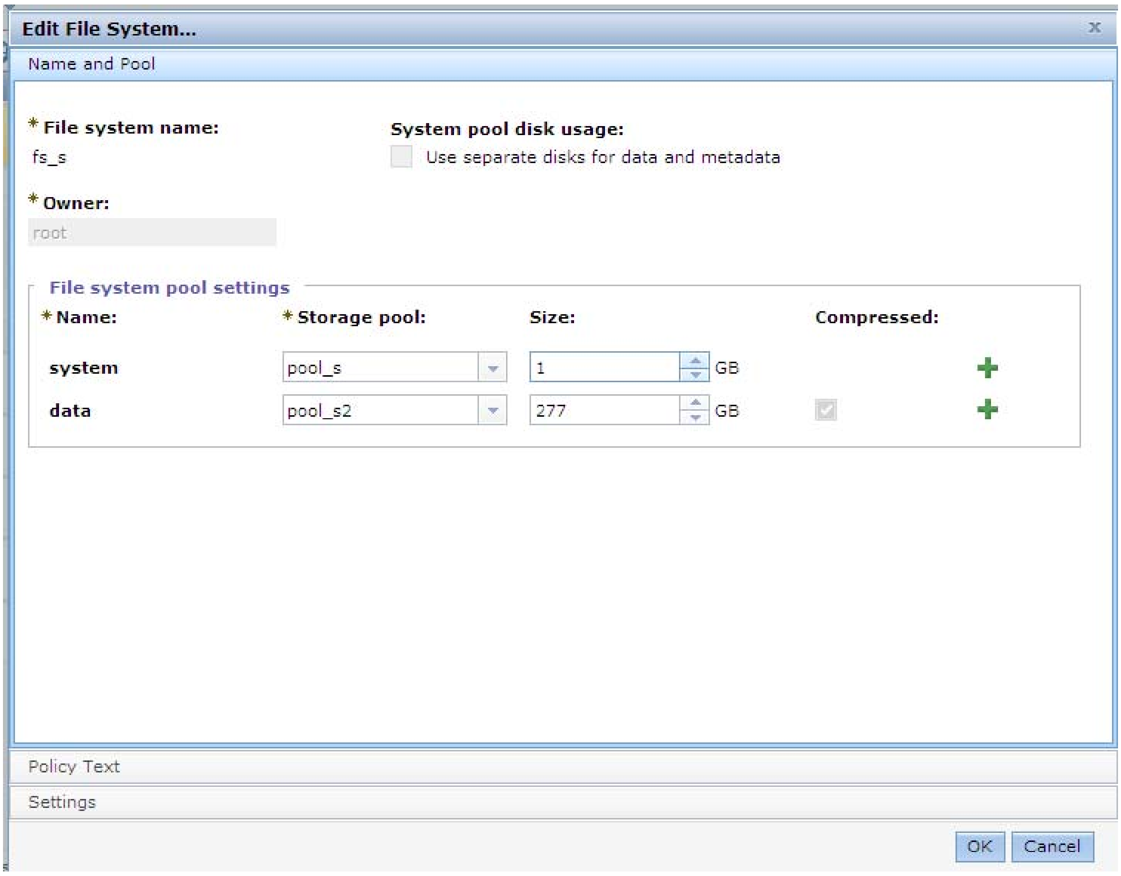
1. Right-click on the desired filesystem and select “Edit File System”. Note that currently the system pool is 100% full and the status is listed as “Critical” for this particular filesystem..



Last Updated: 8/28/12 1:12:03 PM

Name	Capacity	Status	Compressed	Usage Type
fs_s	0% 278.00 GB	Critical	Yes	
data	0% 277.00 GB	OK	Yes	
IFS1346078215486	0% 92.00 GB	OK	Yes	Data only
IFS1346078215682	0% 92.00 GB	OK	Yes	Data only
IFS1346078227425	0% 93.00 GB	OK	Yes	Data only
system	100% 1.00 GB	Critical	No	
IFS1346078204676	100% 1.00 GB	OK	No	Data and metadata

2. The following Edit screen shows the current settings:



Edit File System...

Name and Pool

* File system name:
fs_s

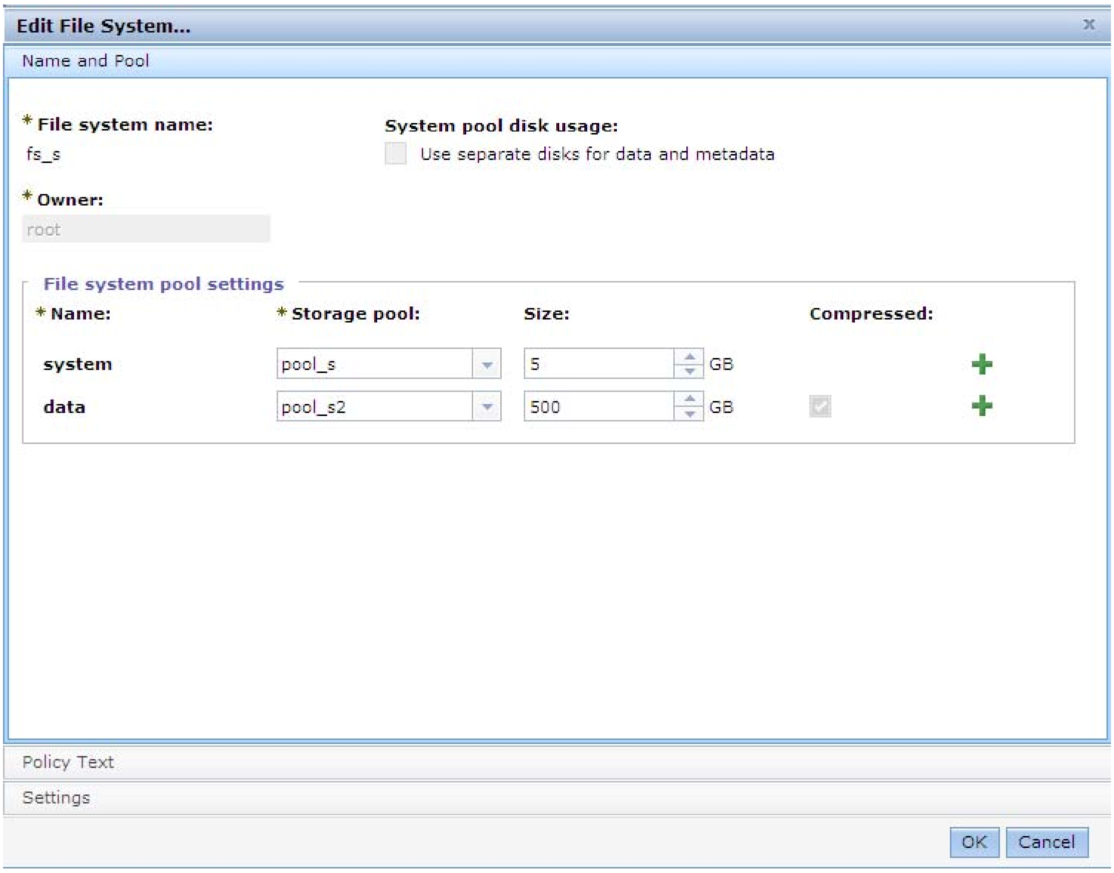
System pool disk usage:
□ Use separate disks for data and metadata

* Owner:
root

File system pool settings

* Name:	* Storage pool:	Size:	Compressed:
system	pool_s	1 GB	
data	pool_s2	277 GB	☑

3. The size of the system pool will be increased from 1GB to 5GB and the size of the data pool will be increased from 277GB to 500GB.



Edit File System...

Name and Pool

* File system name:
fs_s

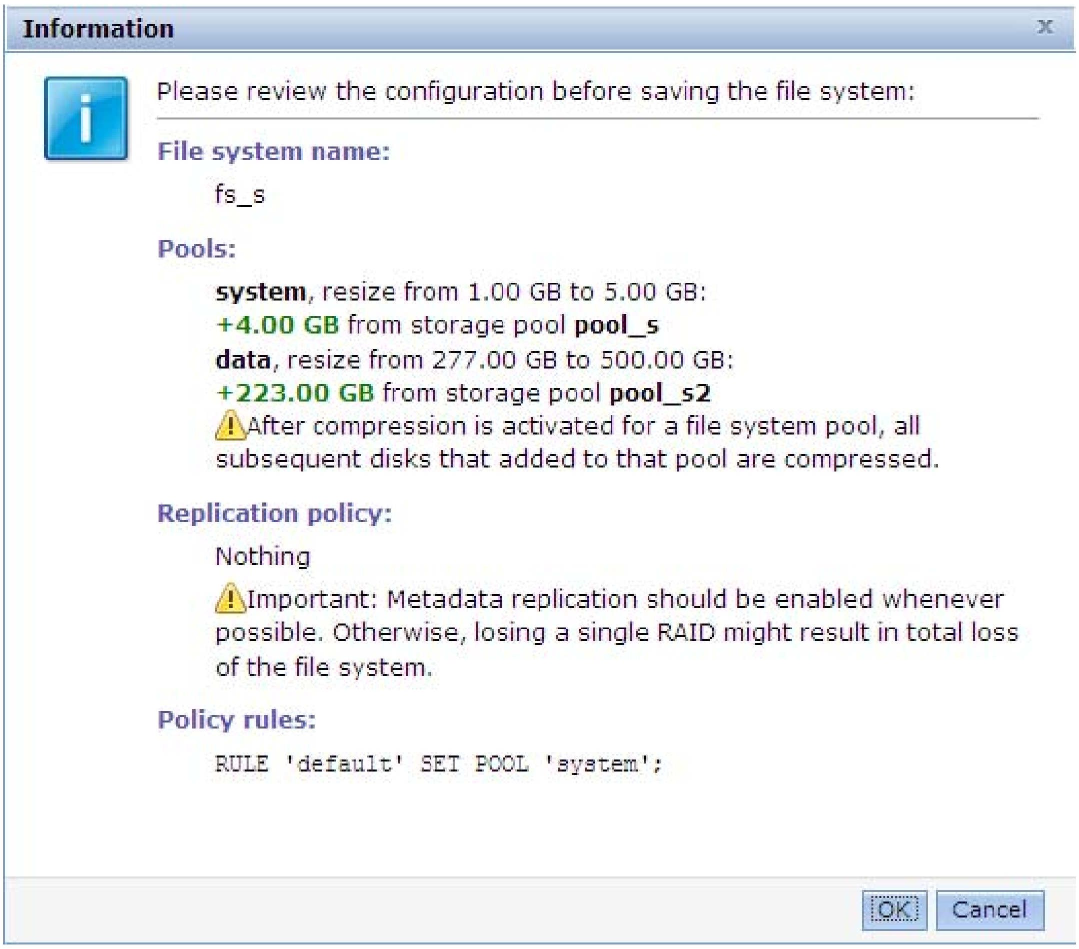
* Owner:
root

System pool disk usage:
☐ Use separate disks for data and metadata

File system pool settings

* Name:	* Storage pool:	Size:	Compressed:
system	pool_s	5 GB	
data	pool_s2	500 GB	☑

4. The preview window is used to review the configuration. Read the warnings carefully.



Information

Please review the configuration before saving the file system:

File system name:

fs_s

Pools:

system, resize from 1.00 GB to 5.00 GB:
+4.00 GB from storage pool **pool_s**
data, resize from 277.00 GB to 500.00 GB:
+223.00 GB from storage pool **pool_s2**
⚠After compression is activated for a file system pool, all subsequent disks that added to that pool are compressed.

Replication policy:

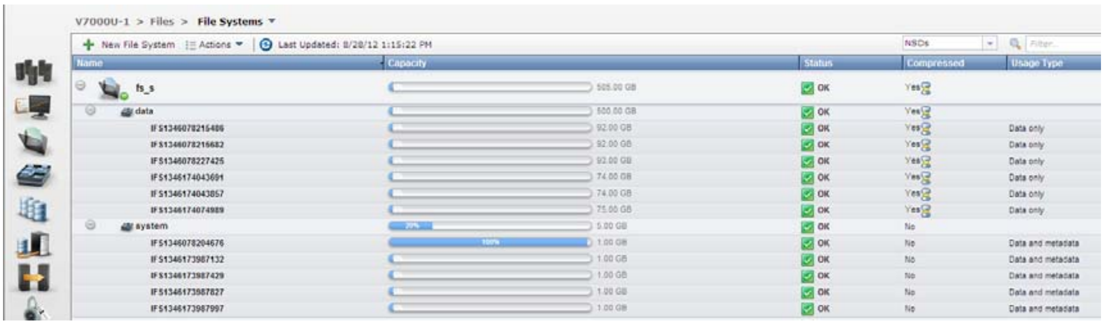
Nothing

⚠Important: Metadata replication should be enabled whenever possible. Otherwise, losing a single RAID might result in total loss of the file system.

Policy rules:

```
RULE 'default' SET POOL 'system';
```

5. The result is that the resizing of the data pool from 277GB to 500GB (+223GB) automatically caused the system to add 3 additional compressed volumes of 74-75GB each. The resizing of the system pool from 1GB to 5GB (+4GB) automatically caused the system to add 4 additional uncompressed volumes of 1GB each.

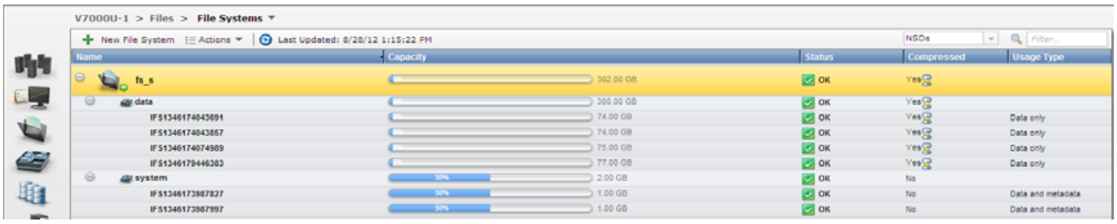


V7000U-1 > Files > File Systems

Name	Capacity	Status	Compressed	Usage Type
fs_s	505.00 GB	OK	Yes	
data	500.00 GB	OK	Yes	
IFS1346078215486	92.00 GB	OK	Yes	Data only
IFS1346078216682	92.00 GB	OK	Yes	Data only
IFS1346078227425	92.00 GB	OK	Yes	Data only
IFS1346174043691	74.00 GB	OK	Yes	Data only
IFS1346174043857	74.00 GB	OK	Yes	Data only
IFS1346174074989	75.00 GB	OK	Yes	Data only
system	20% 5.00 GB	OK	No	
IFS1346078204676	100% 1.00 GB	OK	No	Data and metadata
IFS1346173987132	1.00 GB	OK	No	Data and metadata
IFS1346173987429	1.00 GB	OK	No	Data and metadata
IFS1346173987827	1.00 GB	OK	No	Data and metadata
IFS1346173987997	1.00 GB	OK	No	Data and metadata

6. Using the same process to reduce the filesystem size by resizing the data pool from 500GB to 300GB automatically caused the system to reduce a number of compressed volumes. The resizing

of the system pool from 5GB to 2GB also caused the removal of some of the uncompressed volumes assigned to this pool.



V7000U-1 > Files > File Systems

New File System | Actions | Last Updated: 8/28/12 1:15:22 PM

Name	Capacity	Status	Compressed	Usage Type
fs_s	302.00 GB	OK	Yes	
data	300.00 GB	OK	Yes	
IFS1346174043691	74.00 GB	OK	Yes	Data only
IFS1346174043857	74.00 GB	OK	Yes	Data only
IFS1346174074989	75.00 GB	OK	Yes	Data only
IFS1346179446383	77.00 GB	OK	Yes	Data only
system	50% 2.00 GB	OK	No	
IFS1346173987827	50% 1.00 GB	OK	No	Data and metadata
IFS1346173987997	50% 1.00 GB	OK	No	Data and metadata

Additional Resources

IBM Redpaper: Real-time Compression in SAN Volume Controller and Storwize V7000.

IBM Storwize V7000 Information Center - Version 6.4 / Product Overview / What’s New / Compressed Volumes.