White paper
ETERNUS CS800 - Data Deduplication Background

This paper describes the process of Data Deduplication inside of ETERNUS CS800 in detail. The target group consists of presales, administrators, system engineers and other interested parties.

Contents

Introduction

The term "data deduplication", as it is used and implemented by Fujitsu in this white paper, refers to a specific approach to data reduction built on a methodology that systematically substitutes reference pointers for redundant variable-length blocks (or data segments) in a specific dataset. The purpose of data deduplication is to increase the amount of information that can be stored on disk arrays and to increase the effective amount of data that can be transmitted over networks. When it is based on variable-length data segments, data deduplication has the capability of providing greater granularity than single-instance store technologies that identify and eliminate the need to store repeated instances of identical whole files. In fact, variable-length block data deduplication can be combined with file-based data reduction systems to increase their effectiveness. It is also compatible with established compression systems used to compact data being written to tape or to disk, and may be combined with compression at a solution level.

Note: The data reduction field is one in which standardization of terminology is still emerging. The term data deduplication can also appropriately be applied to data reduction approaches that do not use variable length segments. Some vendors may also use the term to refer to approaches that are primarily file based or that may use fixed-length data segments.

Before we talk about data deduplication it may be helpful to remind readers more familiar with storage at an applications level about how files and data sets are represented in conventional disk- based storage systems. The data in a single file or in a single dataset is rarely stored in sequential or contiguous blocks even on a single disk system, and in the case of RAID storage, data is almost always written to multiple blocks that are striped across multiple disk systems. In the operating system's file system, the file or the dataset is represented by a set of metadata that includes reference pointers to the locations on the disk where the blocks that make up the data set physically reside. In Windows systems the File Allocation Table maps these links; in UNIX/Linux systems the inodes hold the mapping information. Several block-based data storage utilities, including differential snapshots and data deduplication, use a technique in which a single segment or block of data may be referenced simultaneously by multiple pointers in different sets of metadata. The technique for data deduplication also makes use of the idea of using multiple pointers to reference common blocks.

Data Deduplication–Multiple Datasets from a Common Storage Pool

At a summary level, data deduplication operates by segmenting a dataset in a backup environment this is normally a stream of backup data into blocks and writing those blocks to a disk target. To identify blocks in a transmitted stream, the data deduplication engine creates a digital signature like a fingerprint for each data segment and an index of the signatures for a given repository. The index, which can be recreated from the stored data segments, provides the reference list to determine whether blocks already exist in a repository. The index is used to determine which data segments need to be stored and also which need to be copied during a replication operation. When data deduplication software sees a block it has processed before, instead of storing the block again, it inserts a pointer to the original block in the dataset's metadata. If the same block shows up multiple times, multiple pointers to it are generated. Variable-length data deduplication technology stores multiple sets of discrete metadata images, each of which represents a different dataset but all of which reference blocks contained in a common storage pool.

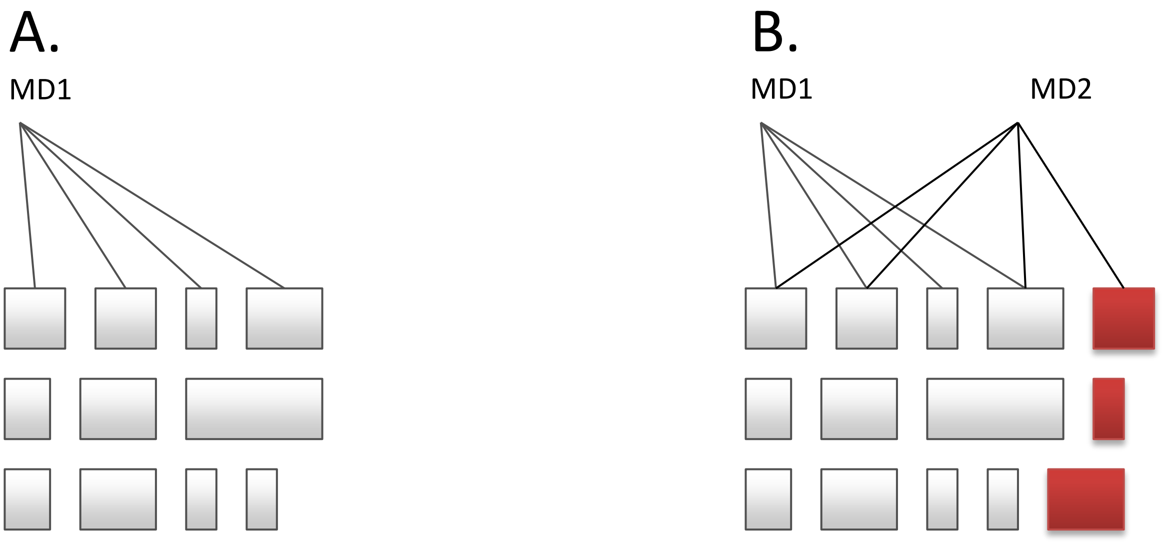


Figure 1 - Data Deduplication Methodology

When a deduplicated storage pool is first created (A), there is one set of metadata with pointers to the stored blocks. As new datasets are added (B), a separate metadata image (MD2) is added for each, along with new blocks. Here, MD1 continues to point to the original blocks; MD2 points both to some of the original blocks and to new blocks. For each backup event, the system stores a complete metadata image of the dataset, but only new data segments are added to the blockpool

Since the leverage of the data deduplication technology is highest when there are repeated data segments, the technology is most frequently used today to store backup data. The methodology allows disk to support retention of backup data sets over an extended length of time, and it can be used to recover files or whole data sets from any of multiple backup events. Since it often operates on streams of data created during the backup process, data deduplication was designed to be able to identify recurring data blocks at different locations within a transmitted data set. Because fixed- size blocks do not support these requirements well, the Fujitsu deduplication methodology is built around a system of variable-length data segments.

Fixed-Length Blocks vs. Variable-Length Data Segments

It is possible to look for repeated blocks in transmitted data using fixed-length block divisions, and that approach is currently being used by several backup software suppliers to include deduplication as a feature of the software, and in at least one backup appliance on the market. Fixed block systems are used most often when general purpose hardware is carrying out deduplication because less compute power is required. The tradeoff, however, is that the fixed block approach achieves substantially less effective reduction than a variable-block approach. The reason is that the primary opportunity for data reduction in a backup environment is in finding duplicate blocks in two transmitted data sets that are made up mostly but not completely of the same segments. If we divide a backup data stream into fixed-length blocks, any change in size to one part of the dataset creates changes in all the downstream blocks the next time the data set is transmitted. Therefore, two data sets with a small amount of difference are likely to have very few identical blocks (see figure 2).

Instead of fixed blocks, Fujitsu's deduplication technology divides the data stream into variable- length data segments using a methodology that can find the same block boundaries in different locations and contexts. This block-creation process allows the boundaries to "float" within the data stream so that changes in one part of the dataset have little or no impact on the boundaries in other locations of the dataset. Through this method, duplicate data segments can be found at different locations inside a file, inside different files, inside files created by different applications, and inside files created at different times.

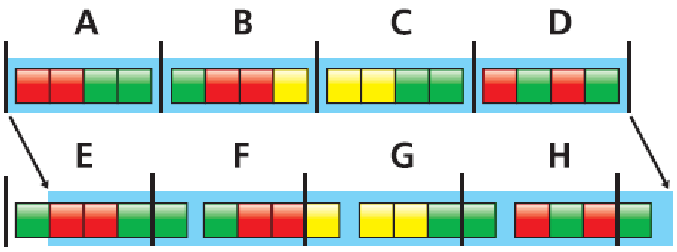
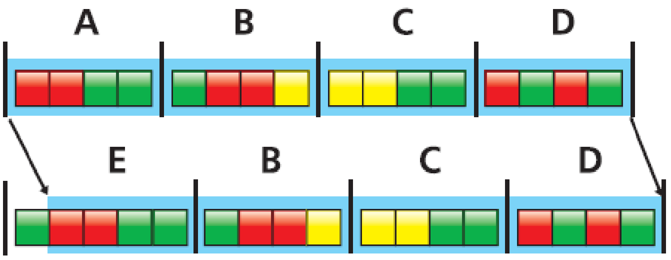


Figure 2 - Dividing Data Sequences into Fixed or Variable Sized Blocks

Applying fixed block lengths to a data sequence:

The upper line shows the original block division—the lower line shows the blocks after making a single change to Block A (an insertion). In spite of the fact that the shaded sequence of information is identical in the upper and lower lines, all of the blocks have changed content and no duplication is detected. If we stored both sequences, we would have 8 unique blocks



Applying variable-length segmentation to a data sequence:

Data deduplication utilizes variable-length blocks or data segments when looking at a data sequence. In this case, Block A changes when the new data is added (it is now E), but none of the other blocks is affected. Blocks B, C, and D are all recognized as identical to the same blocks in the first line. If we stored both sequences, we would have only 5 unique blocks.

Effect of Change in Deduplicated Storage Pools

When a dataset is processed for the first time by a data deduplication system, the number of repeated data segments within it varies widely depending on the nature of the data (this includes both the types of files and the applications used to create them). The effect can range from negligible benefit to a gain of 50% or more in storage efficiency. However when multiple similar datasets are written to a common deduplication pool—such as a sequence of backup images from a specific disk volume—the benefit is typically very significant because each new write operation only increases the size of the total pool by the number of new data segments that it introduces. In data sets representing conventional business operations, it is common to have a data segment-level difference between two backup events of only 1% or 2% although higher change rates are also seen frequently.

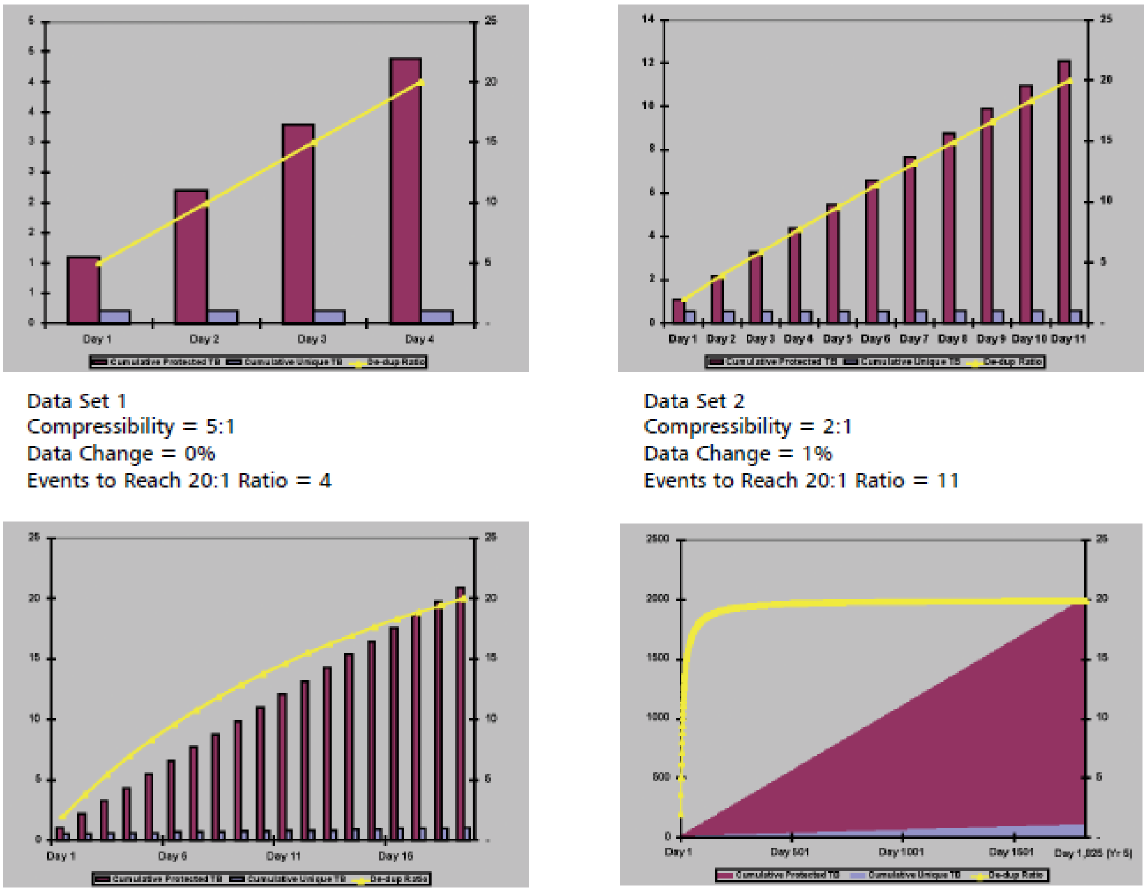
The number of new data segments introduced in any given backup event will depend on the data type, the rate of change between backups, whether a fixed-block or a variable-block approach is used, and the amount of data growth from one backup job to the next. The total number of data segments stored over multiple backup events also depends to a very great extent on the retention policies set by the user—the number of backup jobs and length of time they are held on disk. The difference between the amount of space that would be required to store the total number of backup datasets in a conventional disk storage system and the capacity used by the deduplication system is referred to as the deduplication ratio.

Figure 3 shows the formula used to derive the data deduplication ratio, and Figure 4 shows the ratio for four different backup datasets with different overall compressibility and different change rates. Figure 5 also shows the number of backup events required to reach the 20:1 deduplication ratio widely used in the industry as a working average for a variable-length data segment-based data reduction system. In each case, for simplicity we are assuming a full backup of all the primary data for each backup event. With either a daily full model or a weekly full/daily incremental model, the size of the deduplicated storage pool would be identical since only new data segments are added during each backup event under either model. The deduplication ratio would differ, however, since the space that would have been required for a non deduplicated disk storage system would have been much greater in a daily full model—in other words the storage advantage is greater in a full backup methodology even though the amount of data stored remains essentially the same.

$$\textbf{Deduplication Ratio} = \frac{\textbf{Total Data Before Reduction}}{\textbf{Total Data After Reduction}}$$

Figure 3 - Deduplication Ratio Formula

What is clear from the examples is that deduplication has the most powerful effects when it is used for backing up data sets with low or modest change rates between backup events, but even for data sets with high rates of change the advantage can be significant.



Data Set 1
Compressibility = 5:1
Data Change = 0%
Events to Reach 20:1 Ratio = 4

Data Set 2
Compressibility = 2:1
Data Change = 1%
Events to Reach 20:1 Ratio = 11

Data Set 3
Compressibility = 2:1
Data Change = 5%
Events to Reach 20:1 Ratio = 19

Data Set 4
Compressibility = 2:1
Data Change = 10%
Events to Reach 20:1 Ratio = 1825

Figure 4 - Effects of Data Change on Deduplication Ratios

In order to help end users select the right deduplication appliance, Fujitsu has developed a sizing calculator that models the growth of backup datasets based on the amount of data to be protected, the backup methodology, type of data, overall compressibility, rates of growth and change, and the length of time the data will be retained. The sizing estimator helps users understand where data deduplication will have the most advantage and where more conventional disk or tape backup systems may provide more appropriate functionality.

Note: Contact your Fujitsu representative to participate in a deduplication sizing exercise.

Sharing a Common Deduplication Block Pool

Data deduplication systems gain the most leverage when they allow multiple sources and multiple system presentations to write data to a common, deduplicated storage pool. Fujitsu's ETERNUS CS800 appliances are an example. Each configuration model provides access to a common deduplication storage pool (also known as "blockpool") through multiple presentations that may include a combination of NAS volumes (CIFS or NFS) and virtual tape libraries as well as the Symantec specific OpenStorage (OST) API which writes data to Logical Storage Units (LSUs). Because all the presentations access a common storage pool, redundant data segments are eliminated across all the datasets being written to the appliance. In practical terms, this means that an ETERNUS CS800 appliance will recognize and deduplicate blocks that come from different sources and through different interfaces for example, the same data segments on a print and file server backed up via NAS and on an email server backed up via a VTL.

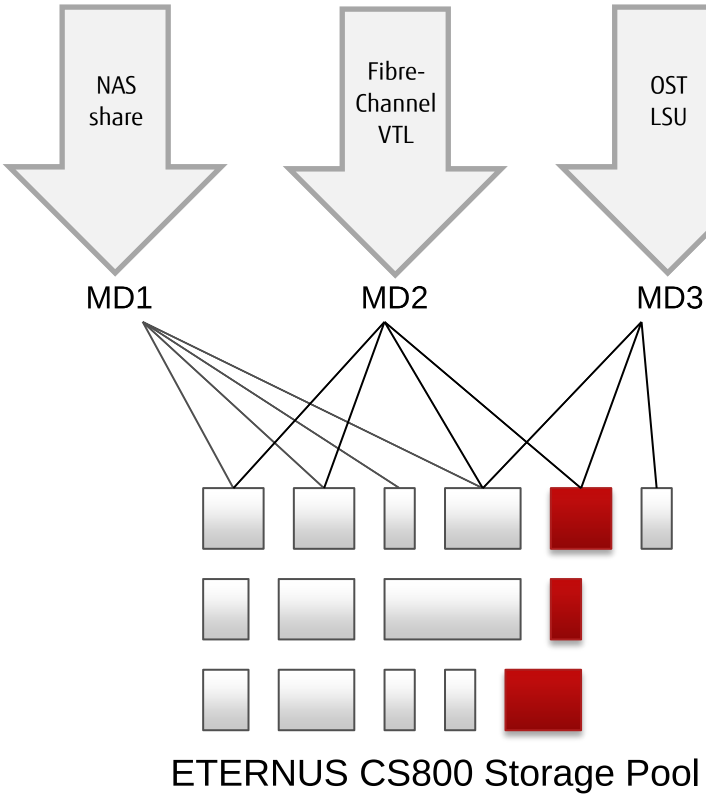


Figure 5 - Sharing a Deduplication Storage Pool

All the datasets written to ETERNUS CS800 share a common, deduplicated storage pool irrespective of what presentation, interface, or application is used during ingest. One ETERNUS CS800 appliance can support multiple presentations and interfaces simultaneously.

Data Deduplication Architectures

The data deduplication operation inevitably introduces some amount of overhead and often involves multiple processes at the solution level, including compression (data is normally compressed after it is deduplicated, most "deduplication" processes also include compression when the total data reduction is considered). This means that the choice of where and how deduplication is carried out can affect the speed of a backup process. The deduplication process can be applied to data in a stream (during ingest) or to data at rest on disk (post-processing). It can also occur at the destination end of a backup operation or at the source (i.e., at the application server where the backup data is initially processed).

Wherever the data deduplication is carried out, just as in the case of processes like compression or encryption, the fastest performance will in most cases be obtained from purpose-built systems optimized for the specific process. An alternative approach that uses software agents running on general purpose operating platforms can also carry out deduplication but it has some disadvantages: all operations are software based, all protected servers must run the agents, application servers carrying out the process are not designed for the specific data deduplication task, and the server resources will be shared with other operations. For these reasons, the functionality of the software agent approach today generally limits it to very small data sets where system performance is not a priority, and to environments with few servers (since on-going server management overhead is relatively high). Systems that divide deduplication steps between different platforms in the same system (part of the process carried out on a client and part on a backup server), can help in situations where bandwidth limitations are the primary concern, but these systems introduce more complexity than an appliance- based approach.

The data deduplication approach with the highest overall performance and easiest implementation will generally be one that carries out the process on specialized hardware systems at the destination end of backup data transmission. It will also tend to keep the overall backup most efficient since the backup process itself is separate from the deduplication effort and can operate at high efficiencies with any backup software package.

Evolution of Deduplication Data Flow

In addition to deciding how to deduplicate data and where the process will take place, there are also different approaches taken to when the operation is carried out. When the first deduplication appliances were developed, most hardware platforms could not bring data into the system and deduplicate it inline fast enough to keep up with the requirements for midrange and Enterprise data protection. The early pure inline systems were limited to relatively small systems, and higher performance was achieved by devices that allowed users to move deduplication out of the backup window. Some systems were designed to use a fully deferred post processing data flow that brings all of a user's data into the appliance during the ingest window, and then carries out deduplication at a later time. These systems were designed to provide faster ingest capability, with the tradeoff of requiring extra disk space for a landing area and delaying replication operations. Some post process operations could also extend the total data protection time—in other words, the time to ingest, replicate, and dedupe might take longer than a more concurrent process.

Fujitsu's first generation appliances combined the inline and post processing approaches, using a methodology that we call "adaptive". In this data flow, data ingested was buffered on disk, but was also deduplicated immediately, as soon as a few MB were written; replication was also carried out as a parallel operation. In addition to providing similar performance profiles to inline deduplication, the adaptive system gave users the option of using the system in a full post processing mode in which deduplication and replication were deferred until a later time. With first-generation hardware, the adaptive approach provided the highest end to end performance and gave users the largest number of options for applying different deduplication policies to different data sets.

The equation changed with the advent of a new generation of higher performance multi-core processors and mass market versions of high performance storage media, including solid state and high rpm SAS drives. That combination made it possible to design deduplication systems that could provide post processing levels of performance using a more traditional inline data flow. Fujitsu went through a year-long process of converting its deduplication software, using an approach designed specifically to take advantage of the new hardware platforms.

Applying Data Deduplication to Replication

Up to now, our discussion has focused primarily on the storage benefits of deduplication, but the technology provides similar benefits to remote replication by dramatically reducing the bandwidth needed to copy data over networks. The result gives disk backup a practical way to provide WAN-based disaster recovery (DR) protection and to reduce requirements for removable media.

The minimum Disaster Recovery (DR) protection required from every IT organization is ensuring that backup data is safe from site loss or damage. Equipment and applications can be replaced eventually, but digital assets are often irreplaceable. No matter how resilient or redundant a given storage or backup system may be or how many layers of redundancy it might have, when all copies of data are located at a single site and in a single hardware system, they are vulnerable to site-specific damage, including natural disasters, fire, theft, and malicious or accidental equipment damage.

Data deduplication technology gives IT departments a new DR option by making inter-site replication over WANs a practical alternative that can enhance DR preparedness, reduce operating expenses, and decrease the usage of removable media.

Background on Replication Approaches

There are two generally accepted models for replication: synchronous and asynchronous. Synchronous replication, often referred to as mirroring, continuously maintains two primary, active datasets in the same state by transferring blocks between two storage systems at each I/O cycle. Synchronous replication is normally designed to provide very rapid failover to the replica if the primary dataset is compromised, and it usually involves two separate storage systems, often in different locations. Because synchronous replication systems delay I/O-complete status signals to the host until both the local and remote writes are complete, they require high speed links, always reduce performance, and are complex to manage. For these reasons, the technique is typically reserved for very high value primary data used in transaction-oriented applications that must remain continuously available.

Asynchronous replication can be applied to mirroring of primary data as well. In this operation, a second data set is maintained dynamically as a duplicate of the primary set, but it is allowed to lag behind the primary by some period of time. The delay may only be one or two I/O cycles—so the mirror is a near replica—but it may be longer. Asynchronous mirroring requires less bandwidth and normally minimizes negative impact on operation of the primary data, although if the mirrored image falls too far behind the primary, the primary system may have to periodically suspend writes to allow the mirror to catch up.

Asynchronous replication can also be applied to non-dynamic, point-in-time images, including backup images, to provide site loss and disaster recovery protection. The technique is much less complex to implement than mirroring techniques, it can provide protection from other classes of faults, can reduce an organization's use of removable media, and it has less impact on primary applications. Backup data is a good replication candidate for DR purposes–it is a point-in-time copy of the primary data, and it is isolated from the primary applications by the backup process. What has kept replication of backup data from being deployed widely is the fact that the large data volume typical of backup has made it difficult to replicate over typical Wide Area Networks.

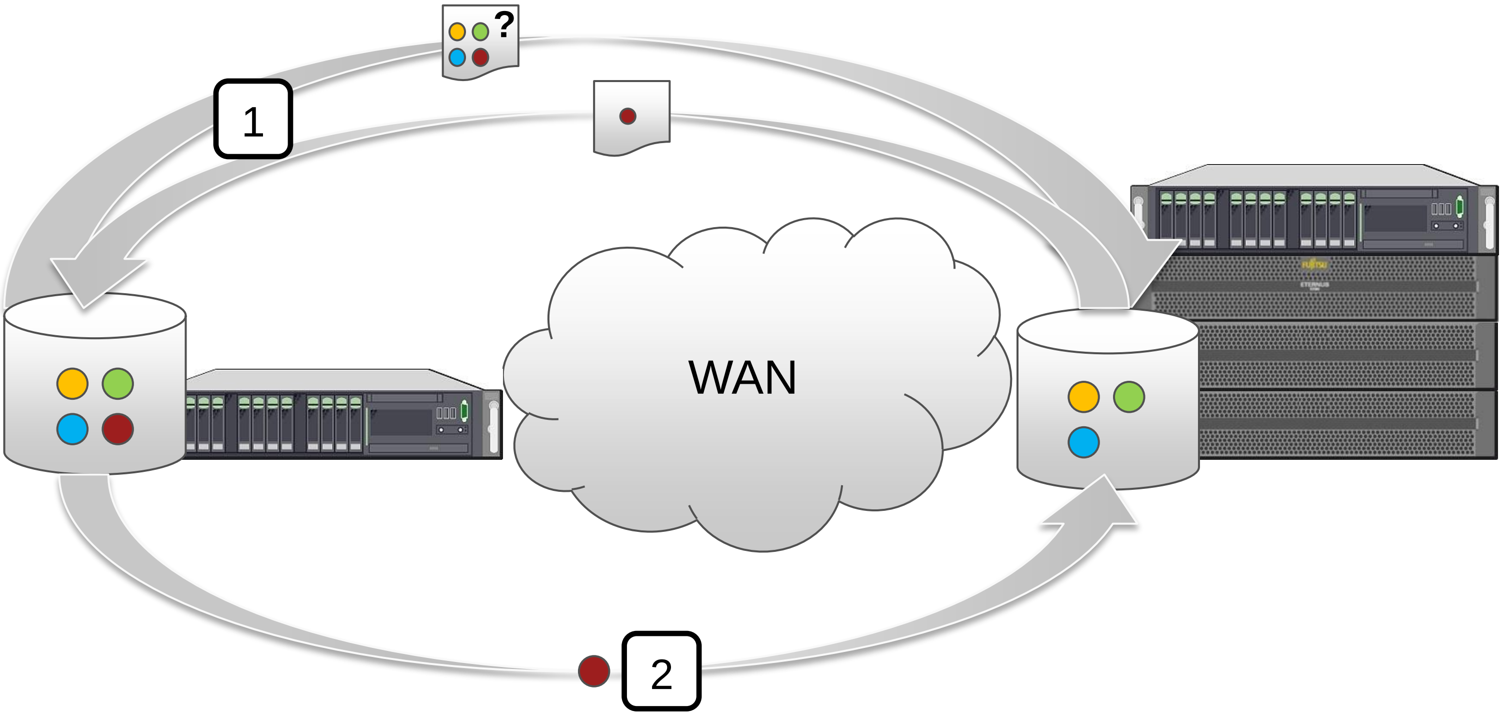
Data Deduplication-Enabled Replication

Data deduplication makes the process of replicating backup data practical by reducing the bandwidth and cost needed to create and maintain duplicate datasets over networks. At a basic level, deduplication-enabled replication is similar to deduplication-enabled data stores. Once two images of a backup data store are created, all that is required to keep the replica or target identical to the source is the periodic copying and movement of the new data segments added during each backup event, along with its metadata image, or namespace.

As a note, the following discussion describes the asynchronous replication method employed by Fujitsu in its ETERNUS CS800 replication solutions. The process used by other vendors and different data reduction systems may differ significantly. ETERNUS CS800 uses replication to create and maintain duplicate images of backup datasets on different devices using transmission over WAN connections.

The replication process begins by copying all the data segments in one division of a source appliance to an equivalent division in a second, target appliance. Although this initial transfer can occur over a network, data volumes often make it more practical to temporarily co-locate the source and target devices to synchronize the datasets, or to transfer the initial datasets using tape.

After the source and target are synchronized, for each new backup event written to the source, the replication process only sends the new data segments. If the new backup event has changed by 1%, the expected bandwidth requirement to create the replica will be 1/100 of the bandwidth that would have been needed to replicate the entire backup dataset written to the source. The bandwidth requirement might be reduced further because of Fujitsu's use of a two-stage, pre-transmission process as part of its replication software.



Step 1:
Source sends a list of elements to replicate to the target.
Target returns list of blocks not already stored there.

Step 2:
Only the missing data blocks are
replicated and moved over the WAN.

Figure 6 - Replication–Verifying Data Segments Prior to Transmission

In this system, before any data is sent to the target device, the ETERNUS CS800 replication software sends a list of the blocks available for replication to the target device (the list is much smaller than the actual data). The target device checks the list of data segments against the index of data segments it has already stored, and it returns a list of elements that are not already locally available and that need to be sent from the source appliance. Then, the source sends copies of only the new data segments over the network. The data segments are sent in the background, the process begins as soon as the backup job has begun being written to the source, and the replication is completed when the metadata for the new backup image is transmitted. At that point the backup image is available for recovery at the target.

The ETERNUS CS800 replication software allows multiple source appliances to point to the same target device and replication normally takes place on a partition-to-partition basis (i.e., each source will consist of a specific appliance partition that replicates data to a similar image on a source device either a NAS share or a virtual library partition). All the replication images at the target are supported by a common deduplication pool which deduplicates data segments across all the backup images sent. That means that deduplication will take place between different source sites—so if the same blocks are backed up at source sites A and B, they only have to be stored once on a common site C when both A and B are replicating data to the same target appliance.

The pre-transmission process that checks to see what data segments are already present at the target site is an important feature of the ETERNUS CS800 replication process. It means that if data segments were backed up yesterday from source site A and they are backed up again today at source site B, they not only will not be stored again at the target, they will not be sent over the network. Only the metadata needs to be sent and stored. This pre-transmission deduplication of the data segments can significantly reduce the bandwidth needed for replication in environments where users in distributed sites work on similar file sets.

Encryption Applied to Replication

Because many organizations use public data exchanges to supply WAN services between distributed sites and data transmitted between sites can take multiple paths from source to target, deduplication appliances need to offer encryption capabilities to ensure the security of data transmissions. In the case of ETERNUS CS800, all replicated data both metadata and actual blocks of data is encrypted at the source level using SHA-AES 128-bit encryption and decrypted at the target appliance.

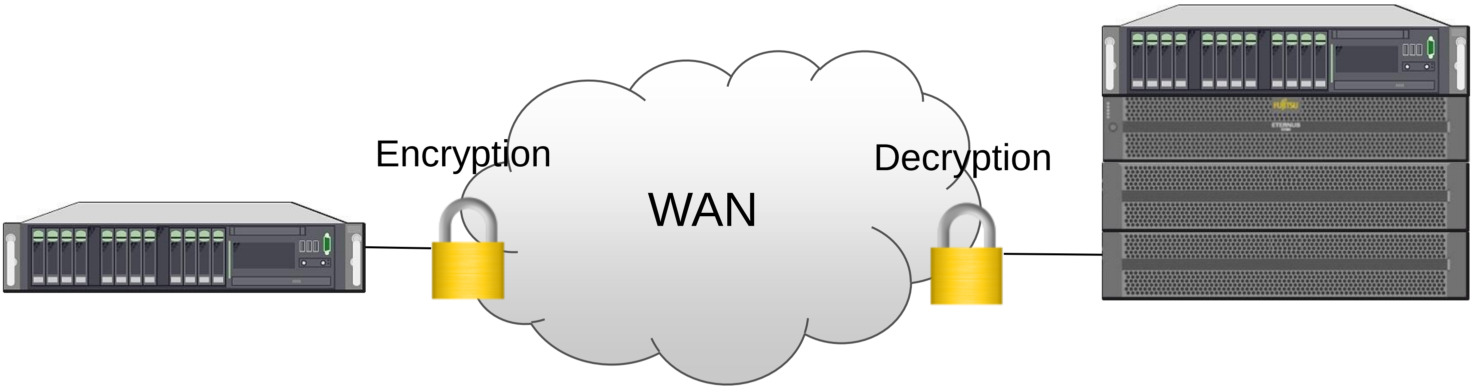


Figure 7 - Applying Encryption to Data During Replication

In ETERNUS CS800 appliances, replicated data is encrypted prior to transmission using 128-bit Advanced Encryption Standard (SHA-AES) and decrypted at the target site. Keys are automatically managed with the units.

Contact
FUJITSU Technology Solutions
Address: Mies-van-der-Rohe-Straße 8, 80807 Munich, Germany
E-mail: storage-pm@ts.fujitsu.com
Website: www.fujitsu.com
2017-01-31 WW EN