WHITE PAPER

Data Reduction Methodologies: Comparing ExaGrid's Byte-Level-Delta Data Reduction to Data De-duplication

February 2007

EXAGRID
SYSTEMS

Though data reduction technologies have been around for years, there is a renewed focus on them recently as they are being utilized by products in the disk-based backup market. Data reduction enables disk to be a feasible long-term retention backup media—making it the same or lower cost than tape-based systems.

There are 2 primary methods of data reduction found in disk-based backup systems:

- Byte-level delta data reduction – compares versions of data over time and stores only the differences at the byte level
- Data de-duplication – reads blocks of data as it is written and stores only the unique blocks

As with any technology, each data reduction method has uses for which it is the best choice. In disk-based backup, ExaGrid's byte level delta data reduction is the better method. It has several important advantages over data de-duplication methods.

This paper discusses the 4 key reasons why ExaGrid's byte level delta data reduction outperforms data de-duplication in a disk-based backup system. ExaGrid's approach achieves the following:

- Scales to larger amounts of data
- Avoids hash table and restore fragmentation issues
- Processes the backup data after it has been written to disk
- Is content aware and optimized for your specific backup application

Scaling to Larger Amounts of Data

An important aspect of the scalability of a data reduction method is the mechanics by which it actually performs its work. ExaGrid's byte-level delta data reduction achieves its goal by comparing the current version of data to past versions. For example, let's say you backup 1TB of data in a weekend full backup. The following week, when you back it up in full again, only 20GB has changed at the byte level. ExaGrid's byte-level delta data reduction compares the two versions and sees the small byte level changes to the data. It then stores the most recent copy using simple 2 to 1 compression, and using byte level delta comparison, retains the prior version by only storing the changes required to reconstruct it when a restore is called for. If you were to keep 20 copies of 1TB, you would typically require 20TB of standard disk. However, if you use 2 to 1 compression on the last backup and byte level delta data reduction on all previous copies, the resulting requirement is just 880GB (500GB compressed last version and 19 x 20GB byte level deltas). In this example the result is a 23 to 1 reduction in the disk required (880GB with the ExaGrid system versus 20TB of standard disk storage).

Hash Table Constraints

Because the byte-level delta method reduces the data by comparing it to past versions of the data and storing only the changes, there is very little data to track versus the alternate method of block level data deduplication. The data deduplication approach uses a very large number of small 8 kb blocks in order to perform data reduction. This results in hundreds of millions of entries to store and track, and greatly increases the likelihood of fragmentation problems. Unlike data deduplication, byte-level delta data reduction stores the backups in large, easier to manage 100 MB segments, avoiding the fragmentation issues that can be associated with the alternate method of block level data deduplication, which creates very large numbers of small chunks spread over the disk. One of the many benefits of byte level delta data reduction and larger 100 MB comparisons, is that it allows you to inter-connect a large number of servers so they can work together as a unified virtual pool of processor, memory and storage, supporting up to 100 TB of primary data in a single virtualized system. This type of architecture can be self-managing and provide load balancing capabilities across servers in the same virtual system.

By contrast, block level data de-duplication works by identifying repeating patterns in data and storing only unique ones. It identifies each pattern by calculating a number based on the data called a hash. When it sees a repeated pattern (i.e., hash), it stores a pointer to the previous occurrence in a table (called a hash table) rather than the actual data, resulting in less space utilization. In other words, when two patterns produce the same hash, they are considered to be the same. Data de-duplication has to break data into very small 8 kb “chunk sizes” so that it can get a maximum number of like segments. In fact, the smaller the chunk size, the more likely data de-duplication can find repeating patterns and gain better data reduction.

The reason for this is very simple. Think of a list of social security numbers in a file. If you were to divide the numbers into segments of three, you would see a lot of repeating patterns as many people share the same first 3 digits. You would also find some accidental similarities. However, if you were to divide the numbers into segments of 9 (whole numbers), you would find no repeating patterns.

The drawback is that the smaller the chunk size, the harder it is to manage the data and the tables required to re-assemble the data when you need it. So, in an attempt to find a balance, most data de-duplication implementations set the average chunk size around 8 kb. This means that for every 1 TB of backup data there would be 125 million hash table entries.

As you process more and more data with data de-duplication, some problems develop:

- The hash tables get too large creating a limit on how much data can be handled
- Hash collisions can occur where two different data chunks result in the same hash even though they are not identical at the byte level
- You wind up with a number of small 8 kb segments fragmented all over the disk

As a result, most implementations of data-deduplication are limited in how much data a single system can manage. While some vendors will claim scalability into the petabytes of data, they accomplish this by selling individual systems that actually behave as isolated blocks of storage, as opposed to a single, cooperating virtualized system.

Restore Fragmentation Issues

Let's consider a 100 GB database backup as an example. If you were to need to restore the most recent backup of that entire database, what would be the difference between a disk-based backup system based on byte-level data reduction versus one using the data de-duplication methodology? With byte-level delta data reduction, the most recent copy of your backup is stored in 100 MB segments and compressed using simple compression technology. To restore that database, no more than 1,000 related segments would be uncompressed, combined, and quickly restored. However, with data de-duplication, the system would need to re-assemble over 12 million separate 8 kb chunks fragmented throughout the disk system to restore that database. Common sense would guide as to which restore would be faster.

In-line versus Post-Processing

Another key difference between ExaGrid's byte-level delta data reduction and most data de-duplication implementations is the timing of the actual data reduction. ExaGrid's byte level delta data reduction performs its data reduction after the backup is already written to disk (post processing). This post processing approach ensures the highest possible performance and the smallest possible backup window. To accomplish this, the ExaGrid system includes enough disk space to ensure there is room for each nightly backup. Data reduction is then performed invisibly, after the data is backed up.

Most data de-duplication implementations process the data "in-line" as it is written to disk. As the data is received, they must create the small 8k chunks, calculate the appropriate hash, and compare it to the ever growing hash table. Based on the result they either store a pointer or write yet another small 8 kb chunk to disk.

Content Aware and Optimized for your Backup Application

ExaGrid Systems has worked with all of the major backup application vendors to enable our disk-based backup system to understand their backup applications. Therefore, ExaGrid's byte-level delta data reduction is content aware and optimized for each backup application. This means that the ExaGrid system knows how each backup application operates, and understands file content and boundaries.

Data de-duplication is done generally at the block level. Therefore, most implementations are not content aware and do not interact differently with each backup application. Therefore, disk-based backup systems based on data de-duplication see data as an unrelated series of blocks. They are not capable of handling the special requirements of each backup application and cannot offer content aware features.

There are several key benefits to being backup application and content aware. Below are two examples:

- Every backup application has nuances in how it writes data to disk and manages things like media rotation (i.e., aging out backups). For example, one backup application deletes an entire file before over-writing it with new data. Another reads just the beginning and end of the former file before overwriting. Because it is content and backup application aware, ExaGrid Systems disk-based backup system is optimized to handle these operations by the backup application, yielding leading performance.

- Most backup applications store a check sum in the backup data to ensure integrity. Again, because the ExaGrid System understands the backup application formats, we can calculate an independent checksum on the data and compare it to the one found in the backup application. If they do not match, then we know there is a problem with the data and can alert the administrator. This gives you another layer of certainty that your backup data is stored correctly.

Conclusion: Advantage - Byte Level Delta Data Reduction

There are 2 primary methods of data reduction in the disk-based backup market. ExaGrid's byte level delta data reduction provides several advantages including better backup and restore performance, scalability, and backup application integration.

Advantage / Disadvantage	**ExaGrid's Byte Level Delta Data Reduction**	**Data De-duplication**
Scales to more data	No hash tables, easy to manage data segments	Requires growing hash tables
Restore fragmentation	Does not have problem (100 GB restore would be no more than 1000 segments)	8 KB chunks spread over disk (100 GB restore would require re-assembly of 12 million chunks)
Backup data processing	Done after data is already written to disk	Done in-line as data is written
Content and Backup application aware	Yes. Optimized for each backup application.	No. Treats each backup application the same.

About Exagrid

ExaGrid® offers a turnkey appliance that works in conjunction with your existing backup applications and is 25-30 percent the cost of standard SATA disk. ExaGrid provides data reduction through byte-level delta technology, which stores deltas for each version instead of storing full file copies. This unique approach reduces the amount of disk space needed to at least 20 to 1, resulting in a significant cost savings over standard SATA drives.

ExaGrid Systems, Inc.
2000 West Park Drive
Westboro, MA 01581
1 800.868.6985
www.exagrid.com