WHITE PAPER

Data De-duplication Methodologies:

Comparing ExaGrid's Byte-level Data De-duplication To Block Level Data De-duplication

EXAGRID SYSTEMS

Table of Contents

Introduction

Though data de-duplication 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 de-duplication 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 de-duplication found in disk-based backup systems:

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

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

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

- Shortest possible backup window due to a post-processing model
- Instant restore capability by keeping most recent backup intact
- Optimizations for your specific backup application and content awareness
- Scales to larger amounts of data and avoids hash table limitations

Shortest Possible Backup Window

A key difference between ExaGrid's byte-level data de-duplication and most block level data de-duplication implementations is the timing of the actual data de-duplication process. ExaGrid's byte-level data de-duplication performs its de-duplication process after the backup has been 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 de-duplication is then performed invisibly, after the data is backed up.

Most block level data de-duplication implementations process the data "in-line" as it is written to disk. As the data is received, the in-line process must perform a series of operations on the data to determine which data is unique and which data is redundant. Though implementations vary, most will refer to the need to cut the data into 8k chunks or blocks, calculate a "hash" that represents the data, and compare that hash with previous entries in an ever growing hash table. Based on the result, the process either stores a pointer or writes yet another small 8 kb chunk to disk. While attempts can be made to optimize these algorithms, there is no substitute for the speed achieved by doing nothing as data flows into a disk-based backup system.

A primary driver behind organizations moving from tape to disk is to reduce the size of their backup window. De-duplication's role is simply to make the use of disk cost-effective. However, to provide you with the space and cost-savings, in-line block level de-duplication methods sacrifice backup performance and therefore do not achieve the best possible backup window. ExaGrid's byte-level data de-duplication provides you with the maximum space efficiency without forcing an organization to sacrifice backup performance.

Instant Restore Capabilities

Another key reason organizations have turned from tape to disk is the speed of restores. Slow or failed restores from tape have a significant business impact. Here again, ExaGrid's byte level data de-duplication delivers superior results when it comes to restore speed versus in-line block level methods.

More than 90% of restores involve the data contained in the most recent backup. Further, restores from the most recent backup tend to be larger restores as the organization may have had a server failure or a disaster and needs the most recent copy of the data versus a historical copy. Generally, restores from historical data involve a single file deleted by a user or a single corrupt database that needs to be recovered. Given that 90% of restores come from the most recent backup, it only makes sense to optimize for that case.

Because ExaGrid's byte level data de-duplication allows the entire backup to land without processing, it can and does maintain a full copy of the most recent backup for instant restore. This means that a restore of the most recent backup is coming from a complete copy on disk versus a copy that has been de-duplicated and requires re-assembly. Restoring from de-duplicated data is naturally slower than from an intact copy on disk. When is restore speed more critical than after a lost server or a disaster?

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 de-duplication versus one using the in-line block level data de-duplication methodology? With byte-level data de-duplication, the most recent copy of the database backup is stored in its entirety. To restore that database, the backup application simply pulls the full copy from disk. There is no reassembly or reversal of de-duplication required.

Remember, the in-line block level based system would have fragmented that 100 GB database backup into 8k chunks as the data was written. There is no complete copy of that database, or any of the backup jobs, in the system. Therefore, the system would need to re-assemble over 12 million separate 8 kb chunks fragmented throughout the disk system during the restore of that database. Common sense would guide as to which restore would be faster.

Content Aware and Optimized for Your Backup Application

ExaGrid Systems has worked with all of the major backup application vendors to enable its disk-based backup system to understand the backup applications format and operations. Therefore, ExaGrid's byte-level data de-duplication 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.

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, when it determines it is time to age out a backup job, one backup application reads in the entire backup job file before over-writing it with new data. This causes a performance hit to the backup window as it basically puts a restore in the middle of the backup. Because it is content and backup application aware, ExaGrid Systems disk-based backup system avoids this penalty by telling the application the read has completed without the need for it to read all of the data. ExaGrid's byte level data de-duplication 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 check-sum 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.

By contrast, in-line block level data de-duplication views a stream of data as a series of 8k chunks (or blocks). These implementations are not content aware and do not interact differently with each backup application. Therefore, disk-based backup systems based on block level data de-duplication are not capable of handling the special requirements of each backup application and cannot offer content aware features.

Scales to Larger Amounts of Data and Avoids Hash Table Limitations

An important aspect of the scalability of a data de-duplication method is the mechanics by which it actually performs its work. ExaGrid's byte-level data de-duplication 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 data de-duplication 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 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 data de-duplication on all previous copies, the resulting requirement is just 880GB (500GB compressed last version and 19 x 20GB byte-level delta changes). 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).

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 de-duplication. The block level data de-duplication 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 block level data de-duplication, byte-level data de-duplication 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 de-duplication, which creates very large numbers of small chunks spread over the disk. One of the many benefits of byte-level data de-duplication 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 30 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. Block level 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 block level 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 block level 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 block level 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 block level data-de-duplication 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.

Conclusion: Advantage - Byte-level Data De-duplication

There are 2 primary methods of data de-duplication in the disk-based backup market. ExaGrid’s byte-level data de-duplication provides several advantages, including shorter backup windows, faster restore times, content awareness and backup application integration, and better scalability.

Advantage / Disadvantage	ExaGrid’s Byte-level Data de-duplication	Block Level Data De-duplication
Shortest possible backup window	Data is written at speed of disk and then post-processed	Data is processed in-line, slowing down the backup
Instant restore	Most recent backup is kept intact on disk providing restores at full speed of disk	8 KB chunks spread over disk (100 GB restore would require re-assembly of 12 million chunks)
Content and Backup application aware	Optimized for specific backup application operations	Treats each backup application the same.
Scales to larger data amounts and avoids hash table limits	Stores data in large 100 MB segments with minimal meta-data	Creates 125 million hash entries per terabyte of data

EXAGRID SYSTEMS

Cost Effective and Scalable Disk-based Backup

http://www.exagrid.com
1-800-868-6985