Mindteck Whitepaper

Data Deduplication and its Benefits

Presented by:

Storage Engineering Team,
Mindteck Inc,
16/3 Cambridge Rd, Bangalore, India

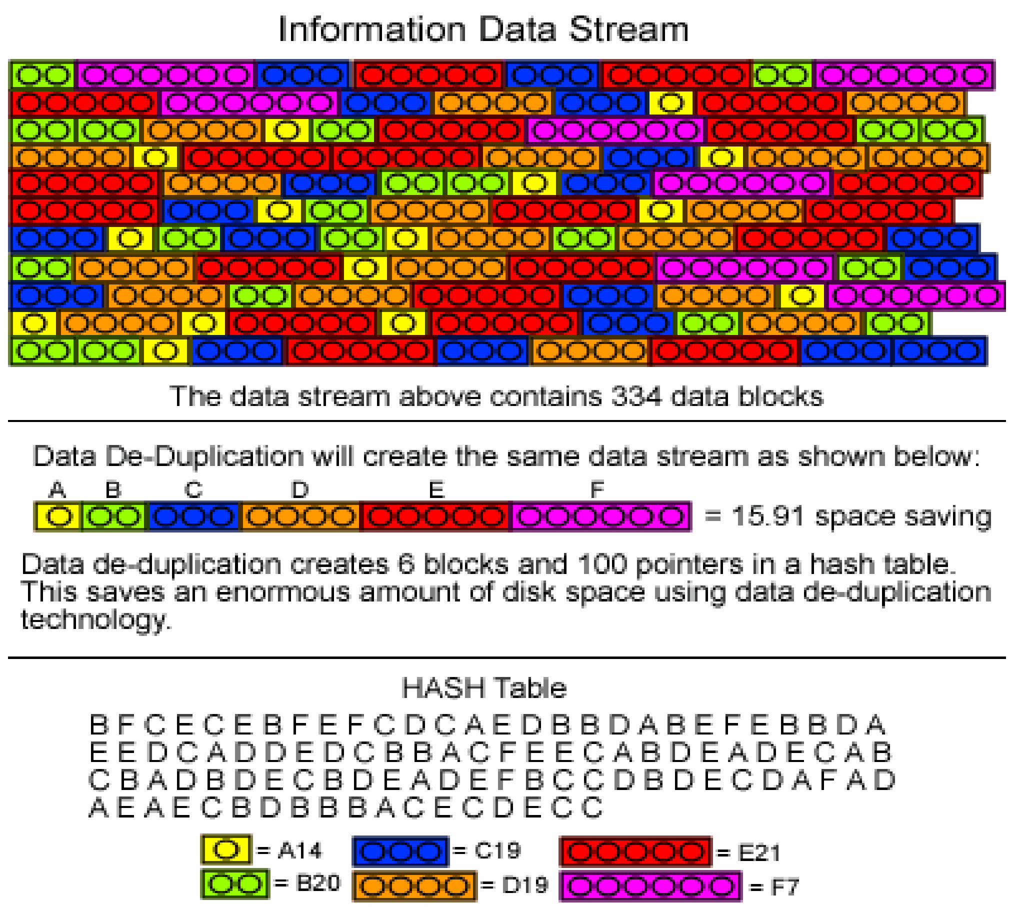
Date: 9th Sept 2009

TABLE OF CONTENTS

WHAT IS DEDUPLICATION

Deduplication is a task of identifying record replicas in a data repository that refer to the same real world entity or object and systematically substitutes the reference pointers for the redundant blocks; also known as storage capacity optimization.

As the volume of data in an organization grows, the amount of repeated data takes a toll on storage availability. For example, a 10 MB PowerPoint presentation copied to 100 users will require 1 GB of storage for the attachments on an Exchange server. The problem gets worse when that 1 GB of duplicated storage is backed up every week. After a year, that 1 GB of wasted space can ultimately demand 52 GB on tape or other backup storage.



LEVELS:

- File level deduplication(Single instance storage)
- Block level deduplication
- Byte level deduplication

PURPOSE:

Despite of less expensive ATA/SATA disk drives, one of the biggest challenges for enterprise storage systems today continues to be the storage cost. The purpose of deduplication is

- To increase the amount of information that can be stored on disk arrays
- To increase the effective amount of data that can be transmitted over Networks

APPROACHES:

The data de-duplication operation inevitably introduces some amount of overhead and often involves multiple processes at the solution level, including compression. This means that the choice of where and how de-duplication is carried out can affect the speed of a backup process.

The de-duplication process can be applied to data in a stream (during ingest) or to data at rest on disk (post processing). The approach is

- In-band: deduplicate the data as they're writing it to the array or VTL
- Out-of-band: deduplication is a secondary process that may run asynchronously

There are advantages and disadvantages to each method.

An important differentiator among deduplication products is whether they work in-band or out-of-band

- The advantage to the in-band method is that it works with the data only one time. The drawback is that, depending on the implementation, it could slow down the incoming backup.
- The out-of-band method has to write the original data, read it, identify its redundancies, and then write one or more pointers if it's redundant. The advantage to this is that you can apply more parallel processes (and processors) to the problem, whereas the in-band method can apply only one process per backup stream. The disadvantage is that the data is written and read more than once, multiple reads and writes.

If the question is how fast we can get our data onto the box then post-processing deduplication is preferred.

IMPLEMENTATIONS:

Since deduplication is a time consuming task even for small collections, the implementation aim is to foster a method to find the weights, identification bounds and the best evidence that should be used for a given digital library.

Each piece of deduplicated data is processed using a "hash algorithm" such as MD5 or SHA-1, or sometimes a combination of the two. This hash algorithm returns a designation that is unique to each piece of data, and the hash is stored in an index. When another piece of data is processed, its hash result is compared to other indexed results. If the current result already exists in the index, that piece of data is a duplicate, so the new data is not saved. Instead, only a "stub" to the existing data is inserted.

- Genetic Programming (GP) [14] is a Machine Learning (ML) technique that helps finding good answers to a given problem where the search space is very large and when there is more than one objective to be accomplished.
- For in-band deduplication file system level implementations are preferred for e.g. WAFL
- For out-of-band deduplication user level implementations are preferred and in some cases integrated with back up software.

BENEFITS:

Data deduplication can achieve data reduction levels ranging from 10 to 1 to 50 to 1. With less storage needed, storage costs are reduced, because this means fewer disks and less frequent disk purchases. Less data also means smaller backups, which translates into smaller backup windows and faster recovery time objectives (RTO). The smaller backups also allow for longer retention times on virtual tape libraries (VTL) or archives.

When the data is actually moved to a backup, archive or replication platform, only the first instance of that data is committed to disk. Subsequent instances are simply denoted with a small stub that references the saved iteration

- Deduplication is ideal for organizations wishing to backup, consolidate and improve performance during backups.
- In cases where the data is being backed up or archived over and over again, the realized storage savings get better and better, achieving 20:1 (95%) in many instances.
- Disk backup systems extend the benefits of data de-duplication across the Enterprise, and integrate it with tape, replication, and encryption into a complete backup solution for multi-site environments
- Data de-duplication reduces disk requirements by 90% or more and makes WAN-based replication a practical disaster recovery tool. The result is fast, reliable backup and restore, reduced media usage, reduced power and cooling requirements (kind of a Green solution), and lower overall data protection and retention costs. But for deduplication to be effective, data must be held long enough so that a comprehensive index of data develops to deduplicate against.

Typical deduplication storage savings for various environments.

Data Types	Typical Space Savings
Backup data	95%
VMware	65%
Email Archival	35%
Document Archival	30%
Audio/Video Files	10%

Note that no repeating archival data such as image files and encrypted data are generally not considered a good candidate for deduplication.

Amount of data at the start of the dedupe process Deduplication Ratio
= ---
Amount of data at the end

40:1, 60:1 and 80:1 are common.

- Operating system VMDKs deduplicate extremely well because the binary files, patches, and drivers are highly redundant between virtual machines (VMs). Maximum savings can be achieved by keeping these in the same volume
- Application binary VMDKs deduplicate to varying degrees. Duplicate applications deduplicate very well; applications from the same vendor commonly have similar libraries installed and deduplicate somewhat successfully; and applications written by different vendors don't deduplicate at all.

OVERHEADS:

The total storage used by the deduplication metadata files is approximately 5% of the total data in the volume. So for 1TB of total data, the metadata overhead would be approximately 10GB to 50GB. The breakdown of the overhead associated with the deduplication metadata is as follows:

- There is a fingerprint record for every 1KB data block, and the fingerprint records for all of the data blocks in the volume are stored in the database file. There is an overhead of ~ 2% associated with this database file.
- The size of the deduplication log files depends on the rate of change of the data and on how frequently deduplication is run. This accounts for ~ 2% overhead in the volume.
- Finally, when deduplication is running, it creates some temporary files that could account for up to 1% of the size of the volume. These temporary metadata files are deleted when the deduplication process has finished running.
- When using the deduplication in a file-based (NFS/CIFS) environment, deduplication is straightforward; as duplicate blocks are freed, they are marked as available, and the system recognizes these free blocks and makes them available to the volume. Deduplication in a block-based (FCP/iSCSI) LUN environment is slightly more complicated. This is because of the space guarantees and fractional reservations often used by LUNs. With space guarantees, for instance, a 500GB LUN that is created consumes exactly 500GB of physical disk space. If the data in the LUN is reduced through deduplication, the LUN still reserves the same physical space capacity of 500GB, and the space savings are not apparent to the user.
- When one deduplication process is running, there is 0% to 15% performance degradation on other applications.
- With eight deduplication processes running, there may be as much as a 15% to 50% performance penalty on other applications running on the system.
- Since deduplication alters the data layout on the disk, it can affect the performance of sequential read applications
- The performance impact is also more significant on sequential reads from SATA drives as compared to FC drives.
- When considering deploying deduplication, scalability would be a concern. You should understand how a storage performance change as the data deduplication system grows. For example, very large hash index tables may hurt performance. All deduplication vendors are taking steps to address scaling performance issues.
- While dedupe requires processing, which takes time and resources, the issue is where to spend the time: at the start of the backup process or the end; and which CPU you want to absorb the processing overhead
- Not much benefits with deduplication of compressed data

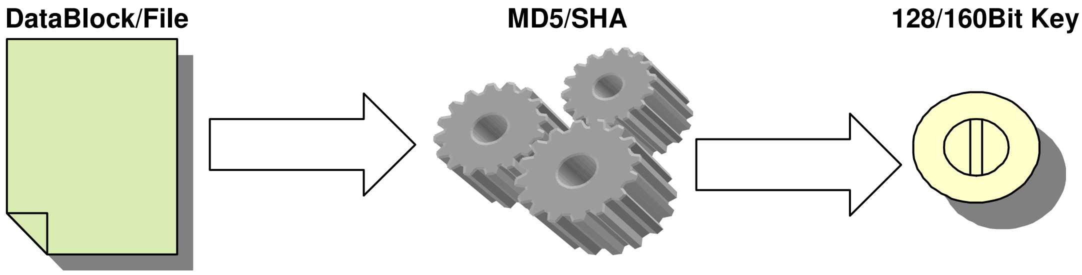
DATASHEET:

Hash algorithms: MD5 and SHA

Both MD5 and SHA will generate fixed size unique key for a particular data set.

MD5:128-bit/16-byte digests. Faster than SHA.

SHA: 160-bit/20-byte digests. More secure and stronger against brute force attacks.



Parameters for deduplication implementation:

1. Storing keys(finger prints) in Primary memory (In Hash table)
2. Storing keys(finger prints) in Secondary memory (In File)
3. Operating data chunk size

- Storing finger prints in primary memory:

  Storing finger prints in Hash table data structure makes faster access but scalability is an issue as explained below.

- RAM size needed for Hash-based index:

  For 1TB of data with 4K chunks needs 250 million 20-byte hash values or 5GB of primary storage. As the data grows, dedicated appliance with huge RAM will be needed. Hence only with lower-end and mid-range data storage can prefer hash-based chunking. With the smaller chunk size 512 bytes, we can obtain more efficient deduplication but required to have huge primary memory.

- Storing finger prints in secondary memory:

  Storing finger prints in files. Three files hash.txt, ref.txt and data.txt are maintained to store hash values, data block pointers and unique data respectively. During the data retrieval, redundant blocks data will be reconstructed with the ref.txt and data.txt files

  Data obtained with deduplication pilot project: Application based deduplication software

  Results of deduplication software run on a file data: Structured data set.

  src.txt: size 4608 bytes: Data set stored = Second block to fifth blocks are duplicate of first block, 7th block is duplicates of 6th, and 8th & 9th blocks were unique.

  Deduplication software creates three files: hash.txt: 132 bytes, ref.txt: 99 bytes, data.txt: 2048 bytes, Total: 2279 bytes. Storage savings = 39%

  Results of deduplication software run on portion of a data disk: Random data set.

  Data size: 100000 blocks with block size=512bytes

  Deduplication software creates three files: hash.txt: 2631255 bytes, ref.txt 1097292 bytes, data.txt: 40824320 bytes, Total: 44552867. Storage savings = 12%