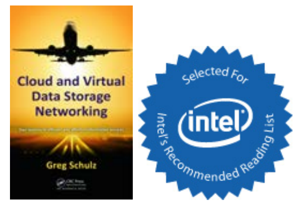
Complements of StorageIO

This chapter download from the book “Cloud and Virtual Data Storage Networking” (CRC Press) by noted IT industry veteran and Server StorageIO founder Greg Schulz is complements of The Server and StorageIO Group (StorageIO). Learn more about the techniques, trends, technologies and products covered in this book by visiting storageio.com and storageioblog.com and register for events and other promotions. Follow us on twitter @storageio or on Google+ among other social media venues.



Visit storageio.com/events to see upcoming seminars and activities

Cloud and Virtual Data Storage Networking has been added to the Intel Recommended Reading List (IRRL) for Developers. Click on the image below to learn more about the IRRL.

StorageIOblog.com site sponsor

Learn more about Cloud and Virtual Data Storage Networking (CRC Press) by visiting storageio.com/books

To become a chapter download sponsor or to discuss your other project opportunities contact us at info@storageio.com .

Chapter 8

Enabling Data Footprint Reduction: Storage Capacity Optimization

The good news is that networks are faster, storage has more space capacity; the bad news is that there is more information to move, process, and store in a given amount of time and within a certain budget.

– Greg Schulz

In This Chapter

- Why to use different technologies for different applications and service needs
- DFR technologies: archive, compression, de-duplication, and thin provisioning
- DFR techniques, including data management

This chapter takes a closer look at different data footprint reduction (DFR) techniques, their characteristics, caveats, and benefits to address various needs. Key themes, buzzwords, and trends that will be addressed include active and inactive data, archiving, compression, data footprint reduction, data management, RAID, thin provisioning, and de-duplication (dedupe).

8.1. DFR Techniques

The importance of data footprint reduction is that it can help drive efficiencies so that more can be done with available resources and technology improvements while supporting demand. By reducing your data footprint, more data can be moved in the same or less time to meet service requirements. Additionally, in order to leverage cloud environments, data needs to be able to be moved effectively in a timely manner. By reducing your data footprint using various technologies, your environment can become more efficient and more effective.

As noted in Chapter 7, there are many different DFR technologies to address various storage capacity optimization needs, some of which are time (performance) centric whereas others are space (capacity) focused. Different approaches use different metrics to gauge efficiency and effectiveness.

In general, common DFR technologies and techniques include:

- Archiving (structured database, semistructured email, unstructured file data)
- Compression and compaction including real time or time deferred
- Consolidation of storage and data
- Data management, including cleanup and deletion of unnecessary data
- Data de-duplication (dedupe), also known as single instancing or normalization
- Masking or moving issues elsewhere
- Network optimization
- Spacing-saving snapshots
- Thin provisioning and dynamic allocation

Table 8.1 shows different DFR technologies and techniques that can be applied across different applications, types of data, and storage.

Which DFR technique is the best? That depends on what you are trying to accomplish in terms of business and IT objectives. For example, are you looking for maximum storage capacity at the lowest cost, with performance not being a concern? Or do you need a mix of performance and capacity optimization? Are you looking to apply DFR to primary on-line active data or applications or for secondary, near-line, inactive, or off-line data? Some forms of storage optimization reduce the amount of data or maximize available storage capacity. Other forms of storage optimization are focused on boosting performance or increasing productivity.

8.2. Archiving

The goal of archiving is to maximize the effective use of on-line or expensive resources by keeping those for active services delivery while preserving information that needs to be retained on lower-cost media. Backup, though similar, focuses on protecting data with a shorter retention period using tools for rapid restoration of a single file, folder, or file system. Archive has longer retention time and a focus on preserving the state of a collection of data as of a point in time for future or possible future access. Archiving can have one of the greatest impacts on reducing data footprint for storage in general, but particularly for on-line and primary storage. For example, if it can be identified in a timely manner what data can be removed after a project is completed, what data can be purged from a primary database, or which older data can be migrated out of active email databases, a net improvement in application performance as well as available storage capacity can be realized.

Table 8.1 Various DFR Techniques and Technologies with Different Characteristics

Technique	Performance	Space or capacity	Comment
Relocation—move problem elsewhere	If off-site, then network issues need to be considered	Keep QoS, SLOs, and SLAs in perspective vs. cost of service	Off-loading to someone else buys time while or until you can optimize
Archive	Maintain or improve general QoS for normal activities in addition to faster data protection or other IRM tasks	Recover space to be used for growth, new applications, or other enhancements. Move archive data to another tier, including cloud or MSP	Applies to regulatory and compliance data or applications, including databases, email, and file shares. Archive, then delete stale or dead data
Backup modernization	Reduce backup/restore or data protection time	Free up space for more backups/faster restore	Reduce overhead of data protection
Bandwidth optimization	Application or protocol bandwidth vs. latency	More data moved in the same or less time	May boost bandwidth vs. latency
Compression	Minimal to no impact on performance, depending on where and how implemented	Some capacity benefit over broad spectrum of applications with various algorithms	Application, database, operating, file system, network, storage system, or device based
Consolidation	Consolidate IOPS to fast 2.5-in. 15.5K SAS or FC HDDs and SSD devices	Consolidate space to large-capacity SAS and SATA HDDs	Avoid causing bottlenecks as a result of aggregation
Data dedupe	Some impact possible on performance in exchange for data reduction benefit	Good data reduction benefit over some types of data or applications	Verify ingest or dedupe rate in addition to restore or re-inflate rate
RAID	May be better option for some data mirroring	Parity-based has less overhead vs. mirroring	Look at number of HDDs per RAID group
Space-saving snapshots	Make copies faster to enhance service delivery	Reduce overhead or space needed for copies	Data distribution, development/testing, analytics
Storage tiering	SSD, 15.5K SAS/FC	2-TB HDDs and tape	Storage may be in cloud
Thin provisioning	Improve SSD utilization	Improve capacity usage	Avoid overbooking

Applying archiving as a form of DFR for cloud or virtualized environments enables more data to be retained in a denser, more cost-effective footprint and reduces the amount of data and associated resources. Archiving can be used to clean up and either discard data that is no longer needed or move it to another medium or resource where the associated management costs are lower. Dedupe and/or compression simply reduce the data's impact; while they may be applicable for some data, they are also similar to treating the symptom instead of the disease.

Archive applies to:

- Transactional structured and unstructured data
- Medical and healthcare environments, including Picture Archiving and Communication System (PACS), as well other Digital Imaging and Communications in Medicine (DICOM) accessible data
- Email, instant messaging (IM), and messaging including voice mails
- Energy and mineral exploration, along with simulation modeling
- Databases such as IBM DB2, Oracle, SAP, MySQL, and SQLserver
- Collaboration and document management, including Microsoft SharePoint
- Engineering drawings, diagrams, and other records retention
- Home directories and file shares or project repositories
- Digital Asset or Archive Management (DAMs) for video and audio
- Security and gaming video surveillance

Data archiving is often perceived as a solution for compliance, but archiving can also be used for many noncompliance purposes, including general DFR, performance boosting, and enhancing routine data maintenance and data protection. The reality is that while regulatory compliance data, including HIPPA, Hitech, PCI, SarBox, and CFR financial or HIPAA medical, require long-term retention, other common application data for almost every business, including those that do not fall under regulatory requirements, can benefit from—if not require—long–term data retention. There are opportunities and DFR benefits of leveraging archive as a storage optimization and green IT enabling technology across all types of data or applications.

Archiving can be applied to structured databases data, semistructured email data and attachments, and unstructured file data. Archiving is evolving and important for preserving information, with clouds as a synergistic opportunity for parking or preserving data. For example, data archived from an on-line database, email, SharePoint, or file system can be migrated to lower-cost, high-capacity disk or tape, or to a cloud MSP service. Key to deploying an archiving solution is having insight into what data exists along with applicable rules and policies to determine what can be archived, for how long, in how many copies, and how data may ultimately be retired or deleted. Archiving requires a combination of hardware, software, and people to implement business rules.

A challenge with archiving is having the time and tools available to identify what data should be archived and what data can be securely destroyed. Also complicating matters is that knowledge of the data value is needed, which may well involve legal issues about who is responsible for making decisions on what data to keep or discard. If a business can invest in the time and software tools, as well as identify which data to archive, the return on investment can be very positive toward reducing the data footprint without limiting the amount of information available for use.

8.2.1. Tools and Targets

While I was writing this chapter, I had a conversation with someone in the storage industry who commented that he thought clouds were going to be the magic bullet for enabling archive and that archiving was the application to drive mass cloud adoption. After further discussion, we came to the conclusion that clouds will be another target medium or virtual medium for archiving. In that role, assuming that cloud and MSP services are cost-effective, trusted, and safe and secure, they can help stimulate archiving deployment where targets or devices are the barriers. However, where the storage medium or target is not the primary barrier to archive adoption, discussion about clouds and optimization may stimulate management interest to get a project underway.

Some common discussion points concerning archiving include what applications and functions you need to support. For example, healthcare may involve Picture Archiving and Communication System (PACS) for medical images as well as Electronic Medical Records (EMR) such as those from Caché-based databases or lab and radiology systems using DICOM-based access protocols. Will you be archiving an entire project, including all files in folders or a file system, or indicial items? Do you also need to archive applications necessary for using the data? If you are archiving a database, are you simply removing rows of data after they have been copied to another table or destination, or do you also have to preserve the context of the data, including business rules with XML wrappers? From a regulation or compliance perspective, what are the requirements of where data can be placed, and how many copies are needed? For example, if your objective is to move archive data to a cloud or MSP service, first verify that there are no regulations stipulating in what geographic are your particular applications data can reside. Some local or national governments regulate what types of data can leave or cross borders into other states or countries.

Indexing or data classification can occur at the application layer via native or optional plug-in capability, via archiving software, or, in some solutions, in the target device, or even all of the above. For legal management systems support, redaction (blanking or marking out of certain data) may be needed along with litigation hold of data to prevent accidental deletion or digital shredding. Other archiving features include encryption for security, write once/read many (WORM), and access audit trails along with reporting.

Archive target devices include disk-based systems that support various interfaces and protocols including NAS, DICOM, or VTL, or object-based access using various APIs including XAM. Various types of media can be found in archive targets ranging from SSD to high-capacity HDDs and tape. Some storage targets support path to tape as well as gateway or bridge functions to remote storage systems or cloud service providers. There are various approaches to energy efficiency, including intelligent power management (IPM), ranging from disk drive spindown to varying performance and power consumption to the amount of work being done along with removable media. Other features often include replication, compression, and dedupe.

An effective archiving strategy or deployment includes:

- Policies such as what to archive where and for how long, how to dispose of data
- Management buy-in and support for implementing policies and procedures
- Organizational involvement from different interests across the business
- Server, storage, or system resource management and discovery tools to determine what you have and its usage profiles
- Application plug-ins to interface with data movers and policy managers
- Archiving tools to interface with applications and target devices or cloud services
- Compliance and security (physical as well as logical) of data and processes
- Storage target devices or cloud and MSP services

Leveraging the right technology, tool, and best practice techniques is important for an optimized data storage environment. To obtain maximum reliability, routine maintenance should be performed on all magnetic media including disk and tape. Routine maintenance includes regular proactive data or media integrity checks to detect potential errors before they become a problem. For disk-based on-line primary as well as secondary and disk-to-disk solutions, media maintenance involves drive integrity checks or powering up spun-down disks along with background RAID parity checks. Media verification can be accomplished using software, appliances, as well as functionality found in some tape libraries.

In addition to media management, another import best practice is securing data during transmission and transportation as well as at rest. This means leveraging encryption to provide data security and information protection compliance, which for some geographic locations is a regulatory requirement. As part of a long-term data retention strategy and data protection, verify that encryption keys are also safely secured as well as available when needed.

General tips and comments:

- Factor in total cost of ownership (TCO) and return on investment (ROI).
- Include time and cost for safe, secure digital destruction of data (tape and disk).
- Archiving is useful for managing compliance and noncompliance data.
- Long-term data retention applies to all types of data that has business value.
- Implement tape and media tracking along with data protection management.
- Adhere to vendor-recommended media management and handling techniques.
- Align the applicable technology, for example, storage tier, to the task at hand.

Keep in mind that you cannot go forward if you cannot go back: As a business, to provide sustainably, being able to go back in time and access preserved and protected data insures business sustainability.

8.3. Compression and Compaction

Compression is a proven technology that provides immediate and transparent relief to move or store more data effectively, not only for backup and archiving, but also for primary storage. Data compression is widely used in IT and in consumer electronic environments. It is implemented in hardware and software to reduce the size of data to create a corresponding reduction in network bandwidth or storage capacity.

If you have used a traditional or TCP/IP-based telephone or cell phone, watched a DVD or HDTV, listened to an MP3, transferred data over the Internet or used email, you have likely relied on some form of compression technology that is transparent to you. Some forms of compression are time-delayed, such as using PKZIP to zip files, while others are real-time or on the fly, such as when using a network, cell phone, or listening to an MP3.

Compression technology is very complementary to archive, backup, and other functions, including supporting on-line primary storage and data applications. Compression is commonly implemented in several locations, including databases, email, operating systems, tape drives, network routers, and compression appliances, to help reduce your data footprint.

8.3.1. Compression Implementation

Approaches to data compression vary in time delay or impact on application performance as well as in the amount of compression and loss of data. Two approaches that focus on data loss are lossless (no data loss) and lossy (some data loss for higher compression ratio). Additionally, some implementations make performance a main consideration, including real-time for no performance impact to applications and time-delayed where there is a performance impact.

Data compression or compaction can be timed to occur:

- On-the-fly, for sequential or random data, where applications are not delayed
- Time-delayed, where access is paused while data is compressed or uncompressed
- Postprocessing, time-deferred, or batch-based compression

Data compression or compaction occurs in the following locations:

- Add-on compression or compaction software on servers or in storage systems
- Applications including databases and email as well as file systems
- Data protection tools such as backup/restore, archive, and replication
- Networking components including routers and bandwidth optimization
- Cloud point of presences (cPOPs), gateways, and appliances
- Storage systems, including primary storage, tape drives, disk libraries, or VTLs

8.3.2. Real-Time and On-the-Fly Compression

With active data, including databases, unstructured files, and other documents, caution needs to be exercised not to cause performance bottlenecks and to maintain data integrity when introducing data footprint reduction techniques. In contrast to traditional ZIP or off-line, time-delayed compression approaches that require complete decompression of data prior to modification, on-line compression allows for reading from or writing to any location in a compressed file without full file decompression and the resulting application or time delay. Real-time appliance or target-based compression capabilities are well suited for supporting on-line applications including databases, On Line Transaction Processing (OLTP), email, home directories, websites, and video streaming without consuming host server CPU or memory resources, or degrading storage system performance.

With lossless compression, compressed data is preserved and uncompressed exactly as it was originally saved, with no loss of data. Generally, lossless data compression is needed for digital data requiring an exact match or 100% data integrity of stored data. Some audio and video data can tolerate distortion in order to reduce the data footprint of the stored information, but digital data, particularly, documents, files, and databases, have zero tolerance for lost or missing data.

Real-time compression techniques using time-proven algorithms, such as Lempel-Ziv (LZ) as opposed to MD5 or other compute, "heavy-thinking" hashing techniques, provide a scalable balance of uncompromised performance and effective data footprint reduction. This means that changed data is compressed on the fly with no performance penalty while maintaining data integrity and equally for read operations. Note that with the increase of CPU server processing performance along with multiple cores, server-based compression running in applications such as database, email, file systems, or operating systems can be a viable option for some environments.

LZ is of variable length for a wide range of uses and thus is a popular lossless compression algorithm. LZ for compression generally involves a dictionary or map of how a file is compressed, which is used for restoring a file to its original form. The size of the dictionary can vary depending on the specific LZ-based algorithm implementation. The larger the file or data stream, combined with the amount of recurring data, including white spaces or blanks, results in a larger effective compression ratio and subsequent reduced data footprint benefit.

Real-time data compression allows the benefits associated with reducing footprints for backup data to be realized across a broader range of applications and storage scenarios. As an example, real-time compression of active and changing data for file serving as well as other high-performance applications allows more data to be read from or written to a storage system in a given amount of time. The net result is that storage systems combined with real-time compression can maximize the amount of data stored and processed (read or write) without performance penalties.

Another example of using real-time compression is to combine a NAS file server configured with high-performance 15.5K SAS and Fibre Channel HDDs with FLASH-based SSDs to boost the effective storage capacity of active data without introducing the performance bottleneck associated with using larger-capacity HDDs. Of course, compression will vary with the type of solution being deployed and the type of data being stored.

Benefits of real-time compression for on-line active and high-performance DFR include:

- Single solution for different applications
- Improved effective storage performance
- Increased capacity for fast disk drives
- Enhanced data protection capabilities
- Extended useful life of existing resources

In some DFR implementations, a performance boost can occur as a result of the compression, because less data is being transferred or processed by the storage system and offsetting any latency in the compression solution. The storage system is able to react faster during both operations and take up less CPU utilization without causing the host application server to incur any performance penalties associated with host software-based compression.

Another scenario for using real-time data compression is for time-sensitive applications that require large amounts of data, including on-line databases, video and audio media servers, Web and analytic tools. For example, some databases such as Oracle support NFS3 direct I/O (DIO) and concurrent I/O (CIO) capabilities to enable random and direct addressing of data within a Network File System (NFS)–based file. This differs from traditional NFS operations, where a file is sequentially read or written. To boost storage system performance while increasing capacity utilizations, real-time data compression that supports NFS DIO and CIO operations expedites retrieval of data by accessing and uncompressing only the requested data. Additionally, applications do not see any degradation in performance, because CPU overhead off-loaded from host or client servers to act as storage systems does not have to move as much data.

One of the many approaches to addressing storage power, cooling, and floor space challenges is to consolidate the contents of multiple disk drives onto a single larger-capacity but slower disk drive—for example, moving the contents of three 600-GB 15,000-RPM SAS or Fibre Channel disks drives to a single 7200-RPM 2-TB SAS or SATA disk drive to avoid power consumption, at the expense of performance and cost for data movement.

An alternative approach is to use real-time compression to boost the effective capacity of each of the fast 600-GB 15.5K-RPM disk drives to approximately the same as the single 7200-RPM 1-TB to 2-TB SAS or SATA disk drive. The benefit is that real-time compression boosts the effective storage capacity by several times that of a single 1-TB or 2-TB HDD without the corresponding 3–4× drop in performance to achieve energy efficiency. This approach is well suited to environments and applications that require processing large amounts of unstructured data, improving their energy efficiency without sacrificing performance access to data. Some applicable usage examples include seismic and energy exploration, medical PACS images, simulation, entertainment and video processing of MP3 or MP4 as well as JPEG and WAV files, collection and processing of telemetry or surveillance data, data mining, and targeted marketing.

Examples of real-time-enabled compression DFR solutions are VTLs for backup and primary enabled on-line storage; EMC CLARiiON is an early adopter, with IBM real-time compression technology acquired in 2010 via their acquisition of Storwize and NetApp, which supports both real-time compression and dedupe in its FAS storage systems and V-series gateways. Other examples include databases such as Oracle, Microsoft Exchange Email, and various file systems as well as storage system–based compression.

8.3.3. Postprocessing and Deferred Compression

Postprocessing and deferred compression are often misunderstood as real-time data compression. These processes dynamically decompress data when read, with modified data being recompressed at a later time. The benefit to this method is that static data that seldom changes can be reduced, freeing storage space, while allowing applications to read more data in a given timeframe.

The downside to this approach is that changing data, including fileservers, email, office documents, design, and development, as well as database files, is written without the benefit of compression. The impact is that more space on disk is required to write the data, with no performance improvement benefit during write operations plus the overhead of a subsequent read and rewrite operation when data is eventually recompressed. An example of this is the Ocarina technology that was acquired by Dell in 2010.

8.4. Consolidation and Storage Tiering

Another form of DFR is consolidating underutilized storage capacity onto fewer larger-capacity devices for lower-performance or inactive data. For higher-performance applications and data, underutilized data can be consolidated onto fewer yet faster devices such as SSD and 15K-RPM SAS or Fibre Channel devices. Consolidating and retiering data onto different storage tiers may seem a little like rearranging the deck chairs on a sinking ship, particularly if that is the only thing being done or if it does not line up with the preferences of some other DFR technique. However, if you have a limited (or no) budget and need some quick relief, combining consolidation along with with techniques such as data management, data deletion, or other DFR techniques mentioned in this chapter can provide near-term relief and serve as a foundation for additional DFR tools.

A caveat for consolidating and retiering is the potential to cause bottlenecks or aggravation as a result of aggregation. Reduce this possibility by consolidating active data onto fewer slower high-capacity disk drives instead of smaller, faster devices. Remember that the objective is to reduce costs and maximize resources to support growth without introducing barriers to business productivity and efficiency.

Retiering means realigning data to the right class or category of technology while balancing needs for performance, availability, capacity, energy, and economics to a given service level objective (SLO). For example, using fewer yet faster devices, combined with some larger-capacity 2-TB 3.5–in. SAS or SATA devices for inactive or less frequently accessed data, to consolidate activity, or IOPS, onto SSD enables fast 15.5K SAS or FC drives to be more effective in terms of space utilization. Similarly, combining consolidation and retiering with archiving enables less frequently accessed data to move off primary storage—including out of structured databases, email, SharePoint or file systems—onto lower-cost media and helps stretch resources further.

8.5. Data De-duplication

Data de-duplication (dedupe) is a technique or technology for eliminating duplicate or recurring data. It is a more intelligent form of data compression.

Some common alternate names or references to dedupe include:

- Intelligent compression
- Normalization by database professionals
- Differencing or elimination of recurring data
- Commonalty factoring to reduce duplicate data
- Single-instance storage (SIS)

Dedupe facilitates:

- Multiple versions of documents without the overhead
- Improving economics of disk- and tape-based backup
- Facilitating faster local backups, and restores
- Boosting storage capacity and utilization
- Improved network bandwidth utilization
- Supporting ROBO or satellite office data protection

Dedupe has become a function found in various products from hardware to software. For example, dedupe can be found or implemented in:

- Operating systems or file systems
- Layered software utilities
- Backup or other data protection software
- WAFS/WAAS/WADM and other bandwidth optimizers
- Agents, appliances, and gateways
- VTL or VxLs and storage systems, including NAS

Deduplication normalizes the data being processed by eliminating recurring or duplicate data that has already been seen and stored. Implementations vary, with some working on a file basis while others work in a fixed block, chunk, or byte boundary; still others are able to adjust to variable-size byte streams. For example, in a backup usage scenario, data that is being backed up is analyzed to see if it has already been seen and stored. If the data has been previously stored, then an entry is made indicating where the data is stored and the new copy is discarded (deduped).

If new data is seen, it is stored and a reference pointer is made along with an entry in the dedupe database (otherwise known as a dictionary, index, repository, or knowledge base). How the incoming data stream is analyzed and what size or amount of data is compared varies by implementations, but most use some form of computed hash value of data being analyzed to enable rapid lookup of known data. This is where the intelligence comes in with regard to dedupe vs. traditional algorithms such as LZ, because the more data that can be seen over time and in different contexts, the more reduction can occur, as is the case with global dedupe (discussed a bit later in this chapter).

While data compression essentially performs a coarse elimination of recurring data patterns, data de-duplication works on a more granular level and requires more processing power and intelligence. Data dedupe builds on traditional coarse compression by adding intelligence, leveraging processing power and awareness of what data has been seen in the past to reduce the data footprint. Essentially, dedupe, regardless of where it is implemented, trades time (thinking and looking at data access patterns or history) for space (capacity) reduction. For example, by being application-aware, data dedupe can look at backup save sets (also known as "tarballs") to identify recurring or redundant files and save a single copy with a pointer to reduce data storage capacity needs. Some dedupe-enabled solutions, such as virtual tape libraries, also combine basic data compression with dedupe to further reduce data footprint requirements.

Current industry and market focus on dedupe is targeted on backup, given its predominance of redundant data. This is not to say that there are not other opportunities; some vendors are finding success with VMs or VDIs, where there are additional duplicates. Focus is also on ratios, where the need is to expand to rates to enable transition to larger, more performance-sensitive environments that are still dominated by tape.

8.5.1. Dedupe Fundamentals

A common technique to check for duplicate data is to use a hash key lookup based on a checksum or chunks of data being seen. Hash keys, computed based on some amount of data being viewed, are compared to stored keys in a database, dictionary, knowledge base, or index of previously stored or known data. When there is a match of a hash of incoming data to known existing data, there is duplicate data. If there is a miss, then there is new data to be stored and a new hash to be added to the index or knowledge base. SHA-1 (Secure Hash Algorithm-1) has been used as an algorithm for creating a comparison or lookup hash in many dedupe solutions.

Other hashing algorithms include SHA-2, with SHA-3 in development, along with MD5. The importance of algorithms being enhanced with more bits is to produce a unique hash key to span larger amounts of data without collisions to maintain data integrity while boosting performance. For example, SHA-1 produces a 160-bit (20-byte) hash, which is adequate for many deployments; however, with larger amounts of storage and expanding data footprints, larger hash keys are needed to avoid collisions.

Where dedupe has more intelligence than traditional compression is in the extensiveness of the dictionary, index, or knowledge base of what has been seen combined with the algorithms for computing hash values. The challenge with dedupe, and why it trades time for space capacity savings, is that time is needed to compute the hash key and look it up to determine whether it is unique.

The ingestion rate for dedupe, or how fast a given amount of data can be processed, depends on the specific algorithms, the size of the available dictionary, and the pool of reduced data, along with available processing performance. As such, dedupe is typically not well suited for low-latency, time-sensitive applications, including databases or other active changing storage use scenarios.

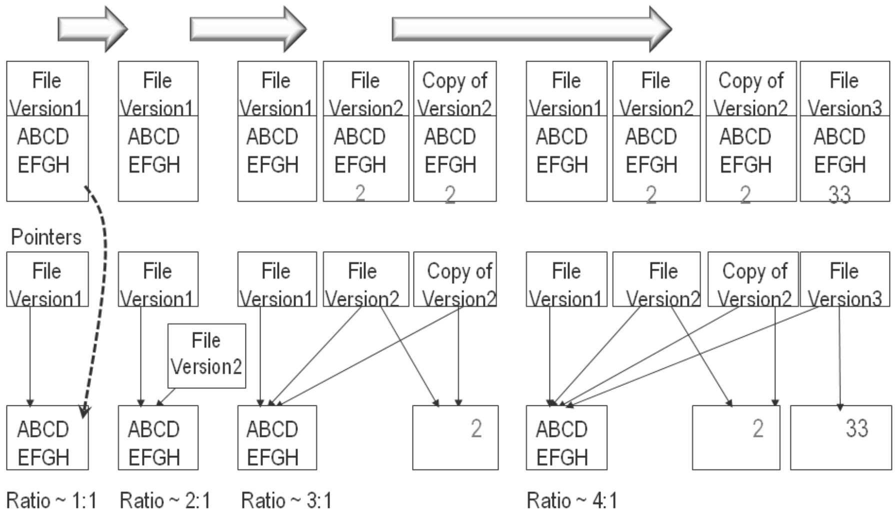


Figure 8.1 Example of dedupe benefit on files with similar data.

Some data dedupe solutions boast spectacular ratios for data reduction given specific scenarios, such as backup of repetitive and similar files, while providing little value over a broader range of applications. This is in contrast with traditional data compression approaches, which provide lower yet more predictable and consistent data reduction ratios over more types of data and applications, including on-line and primary storage scenarios. For example, in environments where there are few or no common or repetitive data files, data de-duplication will have little to no impact, while data compression generally will yield some amount of DFR across almost all types of data.

In Figure 8.1, on the left is an example of a file that is initially processed (ingested) by a generic dedupe engine (actual reduction and compaction will vary with specfic vendor product). Depending on the specfic implementation, the initial savings may be minimal, but after a copy of the first file is made, then some changes made to it and saved, there begin to be reduction benefits. Moving from left to right in Figure 8.2, as additional copies and changes are made, resulting in more duplicate data being seen, additional reduction benefits occur, resulting in a higher reduction ratio.

Figure 8.2 builds on Figure 8.1 in that the data reduction or dedupe ratio continues to rise over time as additional copies or duplicate data are seen by the dedupe engine. As more copies of the data are seen, such as with daily backups, the potential for recurring data increases and, thus, the opportunity for a higher dedupe reduction ratio appears. This capabiltity has contributed to why dedupe has been intially targeted for backup/restore, given the potential for a high degree of duplicate data occuring over time and the resulting space-saving benefit.

Figure 8.3 extends the same example by showing how the DFR benefit continues to improve as more copies of the same or similar data are seen. Depending on the specific dedupe implementation, the reduction benefit may be higher or lower, depending on a number of factors including the specfic algorithms used, local or global view of data, type of data, file or block based, and others.

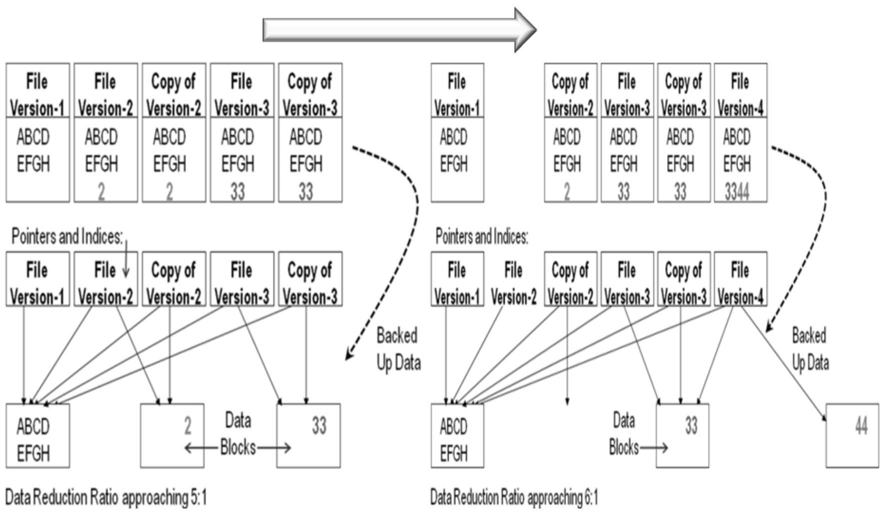


Figure 8.2 Example of dedupe benefit on files with similar data.

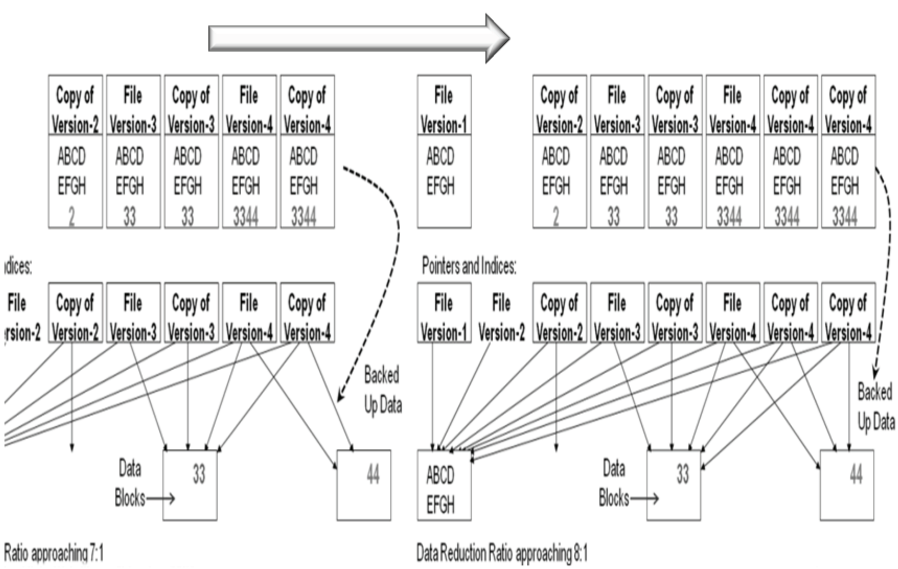


Figure 8.3 Example of dedupe benefit on files with similar data.

8.5.2. How and Where Dedupe Is Implemented

Where is the best place for doing dedupe, and what is the best method? Unless you have a preference or requirement for a particular approach or product, the answer is "It depends."

Some solutions have been optimized for single streams of data, others for dual or multiple streams; some for parallel, others for multithreading to support high concurrency; some are good at ingesting data and others at re-inflating it, while still others do immediate or post or a mix. Some are optimized for scaling, with multiple nodes sharing a global memory, while others have multiple nodes yet are independent. Some are a single-target, others support complex topologies. Dedupe can be implemented in many different ways (Figure 8.4) and locations, even in nonstorage devices such as networking applications or protocol optimizers.

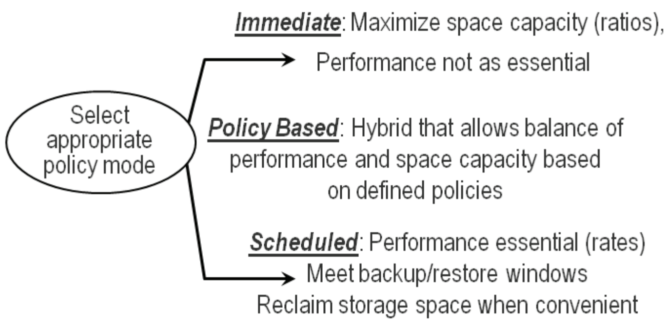


Figure 8.4 Dedupe modes to meet different service-level objectives.

8.5.2.1. Immediate Mode

One mode of operation is to dedupe data immediately as it is processed, either via source-side software or using a target device or appliance. The principle of immediate mode, also known as inline, inband, and real-time or synchronous, is to process (dedupe) the data as it is moved. The benefit of immediate mode is the reduction of the amount of storage space needed at the target and, in the case of source, the reduction of the amount of data being moved across networks.

A drawback of immediate mode is that the time required to process data on the fly can impact data movement. This is very implementation-specific, and vendors have been making great strides in both real-time ingestion (deduping) and real-time restoration (re-inflation) of data to boost performance. This aligns with a growing trend of awareness that data transfer rates (performance), or the amount of data that can be deduped as well as restored in a given time frame, are as important for some environments as reduction ratios or capacity space savings.

8.5.2.2. Deffered or Postprocessing

Whereas immediate-mode dedupe enables DFR to occur during or before data lands on the target device or at the destination, deferred or postprocessing trades space savings for performance. By using additional capacity at the target destination, deferred dedupe enables data to be streamed to meet backup windows and reduced at a later time in the background. This is similar to how some compression solutions work by ingesting the data in its normal format so as not to impact application performance, then transparently reducing the footprint in the background.

As with immediate mode, implementations vary by different vendors' products, so "your mileage may vary" based on type of data and usage. Another thing to keep in mind with postprocessing mode is the restoration rate or performance for a single file as well as for larger amounts of data, including an entire volume or file system. The amount of time delay before reduction occurs also varies by implementation. Some solutions wait until all data has been placed on the target destination, others start the postprocessing almost immediately, after a short time interval or after a certain amount of data is captured.

Some solutions that were initially focused on immediate or deferred techniques are now adding hybrid support for both. These solutions can be configured to process in either immediate or deferred mode, based on specific service-level objective requirements. Essentially, hybrid (Figure 8.4) or policy-based dedupe provides the best of both worlds, by adapting to customers' different application needs. With policy-based enabled solutions, it is possible to, for example, have one stream support immediate mode while another is operating in deferred mode.

Table 8.2 Where to Perform Dedupe

	Source	Target
Drawback	Disruptive to existing BC/DR and backup/restore tools. New software or software upgrades may be required. CPU and memory are consumed on server where dedupe is done.	Extra functionality added to VTL or VxL or storage system or via an appliance or gateway. Does not address reducing data footprint for data that must be moved over networks on a local or remote basis.
Benefit	Performs reduction closer to data source, enabling less data to be moved, making more efficient use of networking resources for local as well as for sending from ROBO or sending to cloud-based target destinations.	Plug and play with existing BC/DR as well as backup/restore software and skill sets. When combined with global capabilities, a larger knowledge base can be leveraged for additional DFR benefits.

8.5.3. Dedupe Locations (Hardware, Software, Appliance, Source, and Target)

In addition to when dedupe is done (immediate or deferred), other variations include where it is implemented, such as in hardware storage systems or appliances as a target destination (Table 8.2), or in software as part of a source backup or data protection tool.

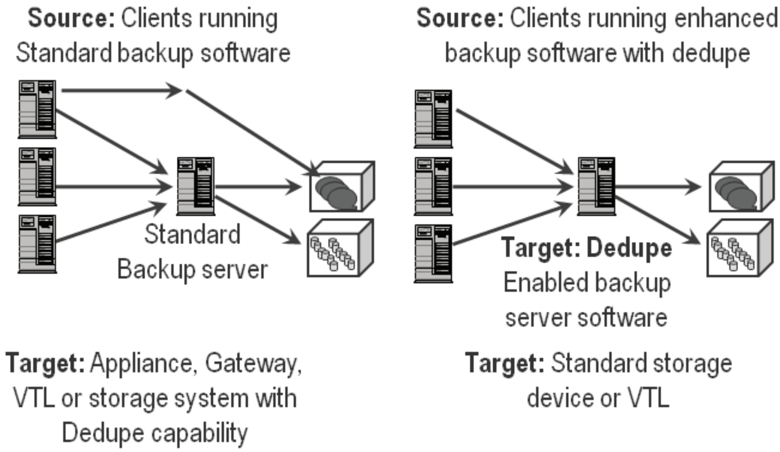


Figure 8.5 Source- and target-based dedupe.

General characteristics of dedupe include:

- Solutions tend to be software-based (or positioned that way).
- Most solutions are "tin wrapped" software (e.g., appliance/gateway).
- Data reduction occurs at or near data target/destination.
- Flexibility with existing backup/data protection software.
- Enables robust scaling options for larger environments.
- Multiple "ingest" streams from sources enable high reduction ratios.
- Examples include dedupe enabled VTLs, gateways, or appliances.
- A knowledge base, dictionary, index, map of what data has already been seen.

Basic aspects of source- and target-based dedupe (Figure 8.5):

- Source can be standard backup/recovery software without dedupe.
- Source can be enhanced backup/recovery software with dedupe.
- Target can be a backup server that in turn dedupes to storage target.
- Target can be a backup device, storage system, or cloud service with dedupe.

Common characteristics of source dedupe include:

- Data reduction occurs via software, at or near the data source.
- Network (LAN, MAN, WAN) activity is reduced.
- Some backup software dedupes at the client, some at the backup server.
- Dedupe is independent of target devices.
- Source dedupe may not achieve the same reduction ratios as target dedupe.

Common characteristics of target or destination-based dedupe include:

- Supports immediate, deferred, or policy mode to adapt to different needs
- Can be a backup server or a storage server, appliance, or gateway
- Optimized for backup/restore, archiving, or general storage needs
- Access via block tape emulation (VTLs)
- File access using an NFS or CIFS interface
- Some support replication to other devices, including VTLs
- Optional support for path to tape, including support for the Open Storage Technology (OST) backup API

8.5.4. Global vs. Local Dedupe

Local dedupe is so named because the dictionary has a scope of only what has been seen by a given dedupe engine instance. A dedupe engine instance might be either source- or target-based, functioning in either immediate or deferred mode. Since the dedupe engine (the algorithm or software functionality) sees only what it processes, the view is localized to that instance. For example, on the left in Figure 8.6, three sources are backing up to a dedupe-enabled device, a target storage system, a VTL, or a backup node. This means that data can be reduced based only on the three streams or backup sources that are seen. As another example, if an environment has six servers being backed up, following the Figure 8.6 example, the dedupe engine on the left does not know about the servers and their data on the right, and vice versa. If dedupe is local, then for a dedupe engine that has multiple nodes, each with a separate instance to boost scaling of performance, availability, and capacity, each instance or engine has its own knowledge base and is limited to only what it has seen.

Consequently, since dedupe reduction ratios or benefits are a product of what data has been seen previously, a localized dictionary or knowledge base can limit the full DFR capabilities for a given environment, and the DFR is limited to the localized environment. This is not all that different from basic compression algorithms, whose DFR benefit or ratio is limited to what they see. What this means is that multiple instances of the dedupe engine cannot benefit from the economics of scale and knowledge of having seen a broader scope of data. It should also be mentioned that local dedupe does not mean that data cannot be sent to a removed destination. For example, a backup solution on a server enabled with source-side dedupe could have a local knowledge base that reduces the amount of data being sent across a network to a remote device target or cloud service.

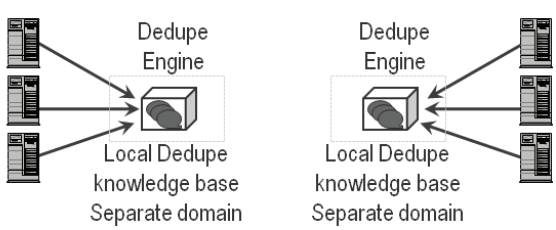


Figure 8.6 Local (nonglobal)-based dedupe.

It is possible that the target destination or cloud or managed service provider also leverages some form of DFR, including target dedupe, to further reduce data by examining multiple input streams. Likewise, a devise enabled as target-based dedupe could be physically at the same local site as the source of the backup or data movement, or it could be remote, however, any data sent over a network with target-based dedupe does not realize the benefits of DFR.

Many dedupe-enabled solutions initially appeared with localized scope–based knowledge bases, because these are easier to engineer and deploy. An emerging trend with dedupe solutions, for source-based as well as target-based solutions, is the ability for multiple dedupe engine instances to communicate with each other or some other entirety, enabling a shared knowledge base. While it is more complex to engineer and coordinate multiple dedupe instances while maintaining performance and data integrity, the benefit is increased DFR capabilities. By having a shared or global database, knowledge base, or index where the results of what has been seen by different dedupe engine instances can be compared, additional redundancies can be eliminated. For example, if six different servers, as shown in Figure 8.6, are configured and load-balanced across a pair of dedupe engines (applications, nodes, backup nodes, or other targets), and they are backing up similar files or data that has duplicate occurrences across the nodes, de-duplication can be enhanced.

Global dedupe is not limited to targets; source-based dedupe implementations also have the ability or potential, depending on where specific vendors are with their technology deployments, to share information. An example would be six different servers backing up or copying or storing data on a common target, which might be a destination storage system, a VTL, an appliance, or even a cloud provider server. The different sources or clients can communicate with the target to get maps or fingerprints of information for comparison, to further reduce data before sending to the destination.

How much information is shared with the source nodes is a vendor- and product-specific implementation, so check with your solution provider as to its specific capabilities. Note that while a common knowledge base or dictionary is used to help enable DFR on a broader basis, a side benefit can be increased resiliency or redundancy. The knowledge base should not be a single point of failure, as it is very important to protect that information in order to be able to re-inflate or undedupe data that has been reduced.

Also note that global dedupe does not have to be limited to a shared or common storage pool where data is actually placed. Another note is that the mere presence of multiple nodes, for example, in a grid or clustered dedupe solution, does not necessarily mean that global dedupe is in use. For some scenarios, multiple nodes may look on paper as if they have global dedupe capabilities (Figure 8.7), yet they may actually have only local dedupe capability.

In addition to where (source or target), how (hardware or software, local or global), and when (immediate or deferred), other dedupe considerations include how the solution works with other DFR techniques, including compression, granularity, or flexibility in data comparison size, support for replication, as well as retiering of data, including path to tape. Another consideration is the different topologies that various solutions can support. One variation is many sources sending data to a single target destination (many to one). Another has one target destination that replicates to one or many other targets in the same or different locations, including those from managed service or cloud providers.

When looking at deduplication solutions, determine whether the solution is designed to scale in terms of performance, capacity, and availability, along with how restoration of data will be impacted by scaling for growth. Other items to consider include how data is re-duplicated, such as in real time using in-line or some form of time-delayed postprocessing, and the ability to select the mode of operation. For example, a dedupe solution may be able to process data at a specific ingest rate in-line until a certain threshold is hit, and then processing reverts to postprocessing so as not to cause performance degradation to the application writing data to the dedupe solution. The downside of postprocessing is that more storage is needed as a buffer. It can, however, also enable solutions to scale without becoming a bottleneck during data ingestion.

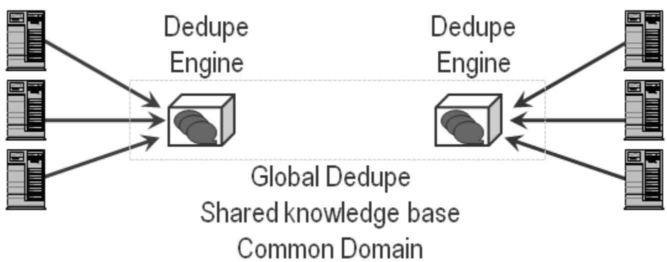


Figure 8.7 Global dedupe.

8.6. DFR and RAID Configurations

Redundant Arrays of Independent Disks (RAID) is an approach to addressing data and storage availability and performance. As a technique and technology, RAID is about 20 years old, with many different types of implementations in hardware and software. There are several different RAID levels to align with various performances, availability, capacity, and energy consumption levels, as well as cost points.

When your budget is restricted, you may have to make do by reconfiguring to get more usefulness out of what you already have. RAID may seem like a very low-tech approach, but ou should reassess your service-level agreements, including RPO/RTO expectations, to verify that they are in fact what is expected rather than what is assumed, and then align the technology. Granted, changing RAID levels may not be easy for some systems; take some time to reassess what RAID levels are tied to your SLA commitments.

Different RAID levels (Figure 8.8) will have a partial impact on storage energy effectiveness similar to various HDD performance capacity characteristics; however, a balance among performance, availability, capacity, and energy (PACE) needs to occur to meet application service needs. For example, RAID 1 mirroring or RAID 10 mirroring and striping use more HDDs, and thus more power, but yield better performance than RAID 5. RAID 5 yields good read performance and uses fewer HDDs, reducing energy footprint at the expense of write or update performance. An effective energy strategy for primary external storage includes selecting the applicable RAID level and drive type combined with a robust storage controller to deliver the highest available IOPs per watt of energy consumed to meet specific application service and performance needs.

In addition to the RAID level, the number of HDDs supported in a RAID group set can have a performance and energy efficiency impact. For example, in Figure 8.8, N is the number of disks in a RAID group or RAID set; more disks in a RAID 1 or RAID 10 group will provide more performance with a larger power, cooling, floor space, and energy (PCFE) footprint. On the other hand, more HDDs in a RAID 5 group spreads parity overhead across more HDDs, improving energy efficiency and reducing the physical number of HDDs; however, this should be balanced with the potential exposure of a second HDD failure during a prolonged rebuild operation. A compromise might be RAID 6, or even emerging triple parity, along with distributed protection schemes, particularly with solutions that accelerate parity calculations and rebuild operations.

General notes and comments regarding RAID include the following:

- Larger RAID sets can enable more performance and reduce overhead.
- Some solutions force RAID sets to a particular shelf or drive enclosure rack.
- Match performance and availability to type of data, active or inactive.
- Boost performance with faster drivers; boost capacity with large-capacity drives.
- Rebuild time will be impacted by drive size for large-capacity SAS and SATA.
- Balance exposure risk during drive rebuild with appropriate RAID level.
- Design for fault containment, balancing best practices and technology.

	Performance	Availability	Performance Overhead	Availability Overhead
RAID 0	Very Good	None	None	N + 0 = 0%
RAID 1	Good	Very Good	Minimum	50%
RAID 5	Poor Writes	Good	High on Write	(1P / N) 6%
RAID 6	Poor Writes	Better	High on Write	(2P / N) 12.5%

Figure 8.8 Summary of RAID levels balancing PACE for application service levels.

8.7. Space-Saving Snapshots

Part of backup or data protection modernization that supports data footprint reduction includes space-saving snapshots. Space-saving or space-efficient snapshots can be used for more than data protection. They also support making copies of production data that can be used for quality assurance or testing, development, decision support, and other uses. The importance of a space-saving snapshot is to reduce the overhead of extra space needed every time a copy or snapshot of a snapshot is made. First-generation snapshots which have been deployed in various systems for many years, if not decades, have continued to improve in terms of performance and space efficiency.

The next wave has been to enable copies of copies using change tracking and redirection on write techniques to reduce the amount of storage required while enabling fast copies to be made. The importance of space-saving as well as traditional snapshots is that as part of a data protection modernization, changing the way information is copied can reduce the overhead of storage needed, enabling a leaner data footprint. The object is to be able to store and retain more data in a smaller, more economical footprint.

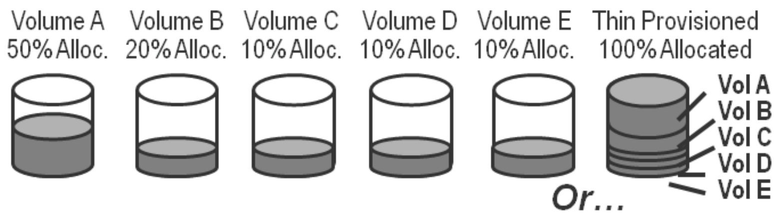


Figure 8.9 Example of thin provisioning.

8.8. Thin Provisioning

Thin provisioning is a storage allocation and management technique that presents an abstracted or virtualized view to servers and applications of how much storage has been allocated yet is actually physically available. In essence, thin provisioning, as seen in Figure 8.9, allows the space from multiple servers that have storage allocated but not actually used to be shared and used more effectively to minimize disruptions associated with expanding and adding new storage.

In Figure 8.9, each server thinks that it has, perhaps, 10 TB allocated, yet many of the servers are using only 10% or about 1 TB of storage. Instead of having to have 5 × 10 or 50 TB underutilized, a smaller amount of physical storage can be deployed yet thinly provisioned with more physical storage allocated as needed. The result is that less unused storage needs to be installed—and consuming power, cooling, and floor space—until it is actually needed. The downside, however, is that thin provisioning works best in stable or predictable environments where growth and activity patterns are well understood or good management insight tools on usage patterns are available.

Thin provisioning can be thought of as similar to airlines overbooking a flight based on history and traffic patterns. However, like airlines' associated disruptions and costs when overbooking a flight, thin provisioning can result in a sudden demand for more real physical storage than is available. Thin provisioning can be part of an overall storage management solution but needs to be combined with management tools that provide history and insight on usage patterns.

8.9. Common DFR Questions

Storage is getting cheaper; why not buy more? It is true that the cost per gigabyte or terabyte continues to decline and that energy efficiencies are also improving. However, there are other costs involved in managing, protecting, and securing data. In addition to those costs, there is the complexity of how many gigabytes or terabytes can be effectively managed per person. A data footprint reduction strategy enables more gigabytes, terabytes, or petabytes to be effectively managed per person.

Why not use dedupe for everything? Perhaps in the future, as processors become even faster, algorithms more robust, and other optimizations occur, dedupe may become more widespread. However, in the near term, dedupe will continue to evolve and find new deployment opportunities where it can be used with data that can be reduced. Other types of data, including graphic or video images, lend themselves better to compression or other forms of reduction, so having multiple tools in your DFR toolbox enables more opportunities.

8.10. Chapter Summary

Organizations of all shapes and sizes are encountering some amount of growing data footprint impact that needs to be addressed, either now or in the near future. Given that different applications and types of data along with associated storage mediums or tiers have various performance, availability, capacity, energy, and economic characteristics, multiple data footprint impact reduction tools or techniques are needed (Table 8.3).

Table 8.3 Data Footprint Reduction Approaches and Techniques

	Archiving	Compression	De-duplication
When to use	Database, email, and unstructured data	Email, file sharing, backup or archiving	Backup or archiving or recurring and similar data
Characteristics	Software to identify and remove unused data from active storage devices	Reduced amount of data to be moved (transmitted) or stored on disk or tape	Eliminate duplicate files or file content observed over a period of time to reduce data footprint
Examples	Database, email, unstructured file solutions	Host software, disk or tape (network routers), appliances	Backup and archiving target devices and VTLs, specialized appliances
Caveats	Time and knowledge to know what and when to archive and delete, data and application aware	Software-based solutions require host CPU cycles, impacting application performance	Works well in background mode for backup data to avoid performance impact during data ingestion

What this means is that the focus of data footprint reduction is expanding beyond that of just de-duplication for backup or other early deployment scenarios. For some applications, reduction ratios are an important focus, so the need is for tools or techniques that achieve those results. For other applications, the focus is on performance with some data reduction benefit, so tools are optimized for performance first and reduction second. In response, vendors will expand their current capabilities and techniques to meet changing needs and criteria. Vendors with multiple DFR tools will also do better than those with only a single function or focused tool.

General action items include:

- Look at data footprint reduction in the scope of your entire environment.
- Different applications and data will need various tools and techniques.
- Some data reduction focuses on space; other techniques trade space for time.
- Have multiple tools in your DFR toolbox.

Vendors with DFR solutions, targets, or enabling capabilities include Amazon, AT&T, CA, Cisco, Commvault, Dell, EMC, Exagrid, Falconstor, Fujitsu, Hitachi, HP, IBM, Index Engines, Iron Mountain, Microsoft, NEC, NetApp, Overland, Oracle, Permabit, Prostor, Quantum, Quest, Rackspace, Riverbed, Seagate, Sepaton, Solix, Spectra, StoredIQ, Sungard, Symantec, Toshiba, Veeam, Verizon/Terremark, WD, and many others. The bottom line is to use the most applicable technologies or combination of technologies along with best practices for the task and activity at hand.

Complements of StorageIO

This chapter download from the book "Cloud and Virtual Data Storage Networking" (CRC Press) by noted IT industry veteran and Server StorageIO founder Greg Schulz is complements of The Server and StorageIO Group (StorageIO). Learn more about the techniques, trends, technologies and products covered in this book by visiting storageio.com and storageioblog.com and register for events and other promotions. Follow us on twitter @storageio or on Google+ among other social media venues.

Visit storageio.com/events to see upcoming seminars and activities

Cloud and Virtual Data Storage Networking has been added to the Intel Recommended Reading List (IRRL) for Developers. Click on the image below to learn more about the IRRL.

StorageIOblog.com site sponsor

Learn more about Cloud and Virtual Data Storage Networking (CRC Press) by visiting storageio.com/books

To become a chapter download sponsor or to discuss your other project opportunities contact us at info@storageio.com .