BACKUP DEDUPLICATION EFFICIENCIES AND CAPACITY PLANNING DEMYSTIFIED

Peter Marelas
Principal Systems Engineer
EMC Backup and Recovery Systems Division
peter.marelas@emc.com

Table of Contents

Abstract

Many organisations have embraced disk-centric backup architectures by adopting purpose-built backup appliances to overcome the reliability and performance challenges associated with tape-centric architectures.

The market[1] for purpose-built backup appliances reached 2.4 billion in 2011 and continues to experience growth. This has resulted in many new vendors' releasing solutions. The market is now saturated with a variety of solutions, from software-only to purpose-build backup appliances and combinations thereof. Each implementation has its strengths and weaknesses. This article will attempt to provide an objective comparison of the functional and architectural properties associated with deduplicated disk-centric backup implementations.

Furthermore, for those that have already adopted deduplicated disk for backup, we discuss capacity planning and why traditional planning models that we apply to primary storage do not work well for deduplicated disk backup systems. To support this discussion, we will provide a generic overview of deduplicated disk backup sizing and how backup requirements and data profiles effect storage consumption. Equipped with this knowledge, the reader will be in a better position to understand and forecast deduplication storage consumption.

[1] IDC's Worldwide Purpose-Built Backup Appliance 2012–2016 Forecast and 2011 Vendor Shares

Introduction

To characterise the qualities of deduplicated disk backup systems it is important to establish how deduplication works in the context of backups. Once we understand how deduplication works, we can start to dive into the different implementations and properties of deduplication systems, and how they impact the backup environment. To be clear, when discussing deduplication moving forward, we are only considering the use of deduplicated disk-based systems to satisfy backup use cases.

Towards the end of this article we will also discuss how the unique properties of backup data and requirements affect our ability forecast and predict deduplication storage consumption. Finally, the article will discuss sizing capacity (not performance) for deduplication systems and offer ideas to improve capacity planning and forecasting of deduplication systems.

How Deduplication of Backups Work

A backup represents all the data associated with a system that the data owner would like to preserve for a given time (i.e. retention). An important characteristic of backups is that from one day to the next, the amount of data that experiences change is typically only a fraction of the total data requiring backups. However, in a traditional tape-centric backup architecture, the tape backup system will write and store all the data from one backup to the next, on tape. Typically, in a well-managed environment, the backup from one full backup cycle would not be co-located with tapes from another backup cycle. This ensures that if one tape in the first cycle fails the problem does not propagate to the set of tapes that make up the second backup cycle. This is referred to as fault isolation. With disk-based deduplication backup systems, the system is designed to eliminate the need to store duplicate data on disk. Because backup environments are effectively backing up the same data over and over again, there are significant amounts of duplicate data sent to and stored on the system. A deduplication backup system employs special techniques that try to determine whether data is unique or has been seen before, and therefore should only be stored once on disk. This characteristic is fundamentally what has allowed disk-backup to provide an economic alternative to tape-centric backup architectures.

Our first quality to consider in the effectiveness of a deduplication system is the data reduction techniques it employs to identify common data and store only the unique data.

Data Reduction Techniques

The effectiveness of deduplication is relative to the stream of backup data (“backup stream”) that enters the deduplication system and the techniques it applies to identify and eliminate redundant data in the stream. There are certain characteristics of a backup stream that can reduce the effectiveness of data reduction techniques.

The first is the amount of data churn that has entered the backup stream. Data churn is a function of the application and determines the amount of volatility the backup stream experiences between backup cycles. Another concern is data order. The order in which backup data is interpreted and transmitted by the backup application can significantly alter the presentation of the backup stream—from the deduplication systems perspective—between backup cycles.

Deduplication systems must deal with these situations effectively or they could find themselves storing significantly more data than anticipated. This is where different data reduction techniques are more effective than others. There are two widely deployed implementations in the market. They are fixed block and variable block deduplication systems.

Fixed block deduplication systems are arguably the simplest to implement. They rely on fixed offsets and block sizes in the backup stream to create chunk boundaries. These chunk boundaries produce chunks of data that are fixed in size, and that are used as the basis for identifying common chunks or blocks of data. The fixed block algorithm is easily defeated when data changes between backup cycles. The changed data enters the backup stream and results in fixed chunk boundaries that are no longer aligned with previous backups. This defeats the algorithm and produces more storage consumption. Similarly, when small block changes take place and these consume small amounts of data within the fixed block chunk boundaries, the algorithm is once again compromised. Different fixed block systems provide different degrees of control to try and overcome these problems. For the most part, these systems allow the user to configure a fixed block size for particular data types to try and minimise alignment issues. However, one significant disadvantage of utilising different block sizes for different workloads is it requires that multiple deduplication pools are maintained. We elaborate on this topic later in this article.

There are variations to fixed block systems that try to overcome the issues associated with chunk boundary misalignment. To achieve this, the deduplication software looks for specific content markers in the backup stream to force the start (and potentially end) of chunk boundaries. These content markers serve as a trigger to reset the chunk boundary in the hope that it will result in chunk boundary re-alignment with previous backups.

There are benefits to using content markers with fixed block algorithms versus using fixed block algorithms alone. However, there are also challenges. For one, content markers need to be maintained. As application file formats and data structures change, content markers also need to change and depending on where content markers are implemented this can require the installation or updating of agents on application hosts. Content markers also assume data is being interpreted by the deduplication system in its natural form. If the content is wrapped in a container—such as the tape-archive (tar) file format—the additional metadata introduced by the container may defeat the content markers.

Beyond the technical challenges, there are also practical challenges with implementing content markers in a backup environment. For the most part, the user must have knowledge of what type of backup data is being generated in order to apply the correct content markers. This relationship needs to be managed consistently in order to sustain the benefits. Similarly, backup policies need to be separated into their respective content types to ensure no data overlaps with foreign data sources. This tight coupling means the deduplication system becomes reliant on a particular backup product. This limits the deduplication systems use to the backup software product it was designed to work with and will be unable to support other backup software products or backup capabilities available with native applications such as Oracle, Microsoft SQL and DB2 to name a few.

The inability to support protection capabilities available with native applications should not be underestimated, as it is the native applications that are best positioned to driving optimised workflows to support ever increasing backup, recovery, and even archiving requirements. We elaborate on the benefits of native application interfaces in subsequent topics.

Variable block deduplication systems typically implement a rolling checksum algorithm that is used to parse the incoming backup stream looking for natural breakpoints. The natural breakpoints result in chunks of data that are variable in size, and that are used as the basis for identifying common chunks or blocks of data. A property of these rolling checksum algorithms is they are designed to yield the same chunk boundaries and sizes if the data passing through them remains the same. Naturally, data does change, and so the algorithm does its best to localise the changes into new chunks (which may or may not be unique) whilst all data leading up to the changes yield the same chunk boundaries and sizes. Once the algorithm passes over the changed data it is able to recalibrate to the common chunk boundary pattern it established in previous backups. This behaviour allows variable block systems to overcome a variety of changes in the backup stream that can defeat basic algorithms. These include block inserts, block deletions, block reordering, and small block changes.

Deduplication Approaches

Deduplication of backup data can take place at different points in the backup process. The three most common approaches implemented are client-side (also referred to as source-side), target-side, and a hybrid approach.

Client-side deduplication

Client-side deduplication systems are typically implemented by the backup application software and are designed to eliminate duplicate data at the source of the data (backup client) before it reaches the backup storage. The unique implementation details of client-side deduplication will not be discussed as they can vary greatly. However, common to most client-side deduplication systems is the concept of a client-side cache. In a client-side approach the cache which resides on the backup client is used as a reference point to determine whether a block of data is a duplicate of a block that has been backed up previously. This hit-or-miss processing occurs for every block that is backed up. If the cache returns a hit, there is no need to send the data to the target backup storage, avoiding a network round-trip. If a cache-miss is returned, the client asks the target backup storage whether it already contains the block of data by sending it a unique fingerprint that is representative of the data. If the target responds positively, the client avoids sending the block of data. If the target responds negatively, the client sends the data to the backup storage for safe keeping.

The idea behind client-side deduplication systems is they eliminate the need to ask and send data to the target if it is storing a block of data that the client (or another client) has sent previously. This stepped approach lends itself to backup use cases that occur over high latency wide area networks, since the interaction required to identify duplicates is in the first instance localised using the cache. If the cache returns a negative result only then does the client rely on a network round-trip to determine whether the target system is storing the same block.

The downside to client-side deduplication systems is that often the client being backed up has to perform all the work necessary to identify and eliminate duplicate data. This process can consume client resources for the duration of the backup window.

Target-side deduplication

Target-side deduplication systems identify and eliminate duplicate data once the data reaches the deduplication backup system. Compared to client-side approaches, the target-side systems do all the work necessary to deduplicate backup data. Consequently, target-side systems rely on the speed of the network between the client and target system in order to complete a backup.

The benefit of target-side systems is they consume moderate levels of client resources over what are typically longer backup windows in comparison to client-side deduplication.

The downside to target-side systems is they do not improve the backup or recovery window if the process is constrained by network performance. Similarly, they are not suited to operating over high latency wide area networks.

Hybrid deduplication

Hybrid deduplication systems implement a blended approach of client-side and target-side deduplication methods. Hybrid deduplication distributes some of the process to the client and the remainder to the target. With this approach the client determines the blocks of data and fingerprint of the block that needs to be tested for duplicity. However, instead of testing it against a local cache, the client sends the fingerprint to the target. The target is now responsible for identifying whether the data is a duplicate or whether the client should compress and send it down the network.

The benefit of this approach is the client consumes fewer resources to identify and eliminate duplicate data as the workload is distributed between the client and target. This approach typically yields comparable network savings to the client-side approach and can also lead to improvements in backup throughput, particularly if the client is starved of system and network resources during the backup window.

The downside of hybrid systems is they are not particularly good at backing up over wide area networks due to the constant interaction required between the client and target system in order to eliminate duplicates.

Post-process versus Inline

There are two predominant methods used to process deduplication data as it enters a deduplication system.

The first method is post process deduplication. This is where duplicates are identified after they have been written to some form of persistent storage on the deduplication system. Systems that implement post process deduplication are analogous to batch systems that accumulate and defer work until a condition is met, e.g. utilisation has reached 50%. The challenge with batch-based systems is that the design assumes there is sufficient time available between non-batch activities (i.e. backups, restore and replication to/from the system) and batch activities (post-processing) to complete the required workload. In practice, the deduplication systems have no control over the backup workload (which drives the replication and to some extent restore workload) that they are required to service and therefore it becomes increasingly difficult to balance the finite resources between all workloads. Consequently, post-process deduplication systems have diminished in relevance for backup workloads.

The second method is inline deduplication. This is where duplicate data is identified in real-time as the data enters the system. Inline deduplication systems rely on balanced architectures to ensure the process of identifying duplication data (which is computationally expensive) can sustain high levels of system throughput. To achieve this, implementations typically rely on filtering and caching algorithms that are designed to consume CPU and memory resources in order to make fast deduplication decisions (i.e. is this block unique?). This design choice enables inline deduplication systems to address larger quantities of deduplicated data which can significantly improve deduplication performance and storage density. The design also allows the architecture to scale as new CPU's are introduced and memory capacity increases.

Deduplication Chunk Size

An important property of the deduplication algorithm is the chunk size or in the case of variable block systems, the average chunk sizes the algorithm yields.

Fixed block algorithms typically leave it up to the user (or a default) to determine the chunk size to use based on the type of data being backed up. The appropriate chunk size to use can vary widely by data type and implementation. Conversely, variable block algorithms will typically force a chunk boundary if a certain number of bytes in the backup stream have been processed and the trigger condition to delineate a chunk boundary has not been met.

The important point to consider is that smaller chunk sizes yield higher deduplication ratios compared to larger chunk sizes, as they are able to localise small block changes with minimal overlap to unchanged and common chunks. Regardless of the deduplication algorithm used and the chunk sizes they yield, not all deduplication systems can support small block deduplication at scale, due to object counts.

To support small chunks a system must provide the ability to manage many objects, and for backup requirements which can be vast in volume, this requirement tests the design and implementation of deduplication systems. To reduce the reliance on the ability of the deduplication system to manage many objects, it is common practice to set guidelines for very large chunk sizes (e.g. 128-512KB) with fixed block systems so as to provide adequate backup performance. This effectively reduces the systems' requirement to manage many objects at the expense of data reduction.

A system that supports extremely large object counts (or chunks) is the Data Domain® 990 system. As with all Data Domain systems, a variable block algorithm is used that yields an average chunk size of 8k. The Data Domain 990 system is designed to accommodate up to 2PB of deduplication storage per system which represents approximately 250 billion objects. Conversely, compare this to typical software-based[2] deduplication systems which, for the most part, rely on databases that can only track a limited number of objects. In the case of Commvault Simpana 9, each deduplication system can accommodate up to a maximum of 750 million records[3] and the smallest block size supported is 32k[4].

Another quality to consider in the effectiveness of a deduplication system is the number of objects it supports and how this translates to the size of the deduplication system.

Deduplication System Size

To scale, many deduplication systems implement the concept of small isolated pools of deduplication. In a backup environment, many small pools of deduplication can represent ongoing management challenges and introduce deduplication inefficiencies for large scale consolidated requirements. This is because isolated pools of deduplication operate independent of other deduplication pools. This results in data stored in one pool not deduplicating against data stored in another; effectively, they operate independently.

For the backup administrator, this creates a significant challenge. To maintain deduplication efficiency it becomes mandatory to ensure the same backups for each client are always located within the same pool. It also becomes necessary to ensure similar clients (i.e. all Windows servers) use the same pool in order to benefit from deduplication between clients. On paper, maintaining this relationship may sound simple however, in practice, when we consider how many clients a typical backup environment supports and how deduplication storage is consumed over time it becomes quite difficult and impractical situation to manage.

---

[2] Software-based deduplication systems such as those included with Commvault Simpana 9 and Symantec NetBackup 7.5

[3] http://documentation.commvault.com/dell/release_9_0_0/books_online_1/english_us/prod_info/dedup_building_block.htm

[4] http://documentation.commvault.com/dell/release_9_0_0/books_online_1/english_us/prod_info/dedup_disk.htm

Consumption of deduplication storage can yield quite erratic use patterns due to the way data changes, grows, accumulates, and expires relative to retention policies. When faced with multiple deduplication pools it becomes necessary to operate with low utilisation levels in order to deal with these capacity fluctuations. Once a deduplication pool fills up, it results in the spill and fill effect. This is where a client's backups end up spreading across multiple deduplication pools. While in a traditional storage environment this would not present a problem because one set of data does not depend on the other to retain storage efficiencies, in backup deduplication this results in significantly more deduplication storage consumption. We have broken the principle that all backups must be co-located in order to deduplicate against each other. This results in having to store a base copy of the client's backup data in each pool. For a one month retention requirement, this base backup yields about 30% of the overall backup storage requirement. When a client's backups spill over into a new pool an additional 30% is consumed. This occurs for each spill and fill event, and is the cause for dramatic reductions in the overall efficiency of deduplication solutions. Therefore, the larger the deduplication pool, the higher the deduplication efficiency and ease of management.

Deduplication system size matters! But the way in which deduplication systems can be scaled also matters. Next we will discuss scale-out versus scale-up architectures.

Scale-Out versus Scale-Up Deduplication Architectures

Previously we touched on the subject of deduplication pools and how the size of the pool contributes to the effectiveness of deduplication solutions. The way in which deduplication systems scale can also determine how systems can be introduced into environments and how easy they are to scale.

For comparison, there are two predominant scaling implementations; scale-out and scale-up.

In scale-out architectures we have to consider the spill and fill dilemma discussed previously. We can either procure a single deduplication pool then move data and policies around as the pool fills up and we are forced to add new pools, or we have to establish multiple pools when the environment is first procured, to provide a platform to evenly distribute backup workloads. In practice though, provisioning pools of under-utilised storage is not common practice for cost-conscious environments. Additionally, data between different applications and systems grows at different rates. As demands increase and variability in the environment becomes the norm, solutions that employ small deduplication pools require constant workload management. For large scale and fluid environments this becomes increasingly impractical to manage. Any cost benefit that may have been perceived up front is eroded over time by solution complexity and management overhead.

Scale-up architectures such as EMC Data Domain implement a file system (DDFS) and platforms that can be grown in-place and online. This property allows a minimal base footprint to be established to satisfy initial backup requirements and then, as retention requirements are absorbed, storage can be added and the file system grown. The benefit of this architecture is the ability to start small and grow as required. This avoids the need to over-provision storage or the need to move data and policies around in order to partition and re-balance workloads. Scale-up architectures also allow environments to run at higher utilisation levels, since there are fewer storage pools to maintain. Similarly, they are less likely to suffer from the effects of spill and fill.

However, at some point, if the requirement is large enough and data grows beyond forecasts, even scale-up architecture may need to scale-out. Thus, another property of deduplication systems that requires consideration is the platforms lifecycle capabilities, and whether they can help ease the process of scale.

Platform Lifecycle

A perhaps unique property of the Data Domain 4200, 4500, 7200 and 900 family of deduplication systems is the concept of system controller upgrades. The system controller is the brains of the solution and can be upgraded to support more capacity and performance capabilities. This in-place upgrade process enables one to retain the investment in storage capacity between generations. There are some caveats[i] to consider as not all models or shelf generations support this capability. However, for the combinations that are supported, the property provides opportunity to scale an existing investment without the need to replace the entire unit or migrate some or all of the backup data to other systems.

Few alternatives in the market possess this property. Most require data to be migrated between platform and generations either through the features available with the system, if supported, or at the most basic level via the supported backup applications.

Without appropriate lifecycle considerations, deduplication systems—particularly those that are required to preserve data for years to come—can pose significant service continuity and management challenges. To scale and leverage benefits from next generation technology platforms, it is important to review the properties of the platform to support the lifecycle process.

Homogenous versus Heterogeneous

Tape-based backup systems are considered heterogeneous in the sense that if an investment is made in tape media, tape drives, and tape robots then the solution can be used to satisfy the tape requirements of many different backup applications that support tape. This heterogeneity does not apply equally to deduplication solutions.

There are two types of deduplication solutions in the market. Homogenous solutions designed to work with one backup software product only and heterogeneous solutions designed to work with many different sources of data, including backup software products and backup and recovery capabilities native to applications.

Homogenous solutions are typically implemented by software-based deduplication systems that are tightly coupled with backup application software. Heterogeneous solutions often work with many different backup application software and native applications.

In order to provide more options to integrate, heterogeneous solutions often support widely deployed protocols such as NFS, CIFS, NDMP, and tape interfaces. It is these interfaces that allow them to be used by many different data sources. Similarly, some heterogeneous solutions also implement their own transport protocol that can be integrated with application vendor's software to further optimise the data and control path between the application and the deduplication system. These applications are not limited to backup software and can include applications such as databases. An example of one such solution is EMC's Data Domain® system integrated with Oracle Data Domain Boost[5].

The choice of which solution is best is ultimately determined by the user's requirements and how they wish to interface and integrate with application protection requirements. However, one clear advantage of heterogeneous systems is they provide more interfaces and application integration to satisfy the backup and recovery requirements of application owners. Furthermore, application native interfaces implement native data formats. For backup and in particular long term retention of backups, adopting native data formats avoids the long-standing issue of maintaining a recover capability for proprietary data formats. These proprietary data formats often lead to software lock-in, which for backup use cases, creates significant barriers of entry to adopting emerging technologies and methods. Finally, storing data in native data formats also opens the door for new opportunities to do more with the data (e.g. analytics) and expand the value of the service and the role of the deduplication system.

[5] http://www.emc.com/collateral/hardware/white-papers/h10683-dd-boost-oracle-rman-tech-review-wp.pdf

Trust Your Backups

The final point concerning deduplication system properties is trust. The ability to trust a backup system is fundamental to data protection, as they represent the last line of defence in a data protection strategy, to protect and preserve data.

When disk-based backup systems are used, the requirement to trust the disk-based solution is elevated, because data that is stored on disk is by its very nature, sharing disk spindles with other data. In a deduplication system this sharing of blocks is amplified, as we are retaining multiple recovery points that are sharing the same blocks and same underlying disks. Consequently, the software and hardware properties implemented by a deduplication system to trust the data stored in them is of paramount importance.

Interestingly, very few deduplication systems provide elevated levels of protection for the data they store. Many systems employ the standard properties one would expect from any storage system. These include RAID 6 and NVRAM and are limited to hardware-based solutions. Software-based solutions rely on the user to make their own choices. Beyond the basic properties there are few solutions that make any attempt to provide elevated levels of protection both for the data and metadata required to provide assurances that backup data can be restored when required.

As a point of comparison, Data Domain systems go to great lengths to provide data assurance. They employ offensive design principles such as real-time read-on-write data verification and error correction, a log-structured file system to avoid overwriting old data with new, performance of full stripe writes to avoid partial stripe updates, and verified checksums throughout the system including memory and reliance on battery-backed NVRAM for fast and safe system restarts. However, as good as they are, offensive design principles are not enough, because backup data remains largely dormant and this dormant data represents the last copy of data in the environment. Thus, to protect against latent errors either caused by logical software or physical media failures, the Data Domain system employs defensive design principles such as data scrubbing to verify the integrity of data and correct errors. As important as the data is, metadata is just as, if not more important. Metadata holds the key to unlocking the sequence required to turn deduplicated data into a verified, recoverable backup image. To provide this level of assurance, Data Domain systems implement a unique self-describing metadata format that can be verified as consistent through referential integrity properties of the data structure. This allows the metadata to be recreated and verified consistent in the event of a system failure.

Understanding Deduplication System Capacity Planning

Now that we have an understanding of how deduplication works and how properties of deduplication effect efficiency we can start to discuss how backups consume storage and the challenges this creates for capacity planning.

In a traditional primary storage array, storage is consumed relative to the data stored on the device. The type of data stored on the device and the amount of churn that data goes through during its lifetime does not impact primary storage consumption. Therefore, one can conclude primary storage systems consumption is impacted by data growth only.

In a deduplication system, consumption is impacted by the type of data stored, how often and how much data churns, and how data grows. Two other unique characteristics of backup environments that also impact consumption are the backup cycle and the backup retention.

The backup cycle determines how often and what type of backup is created (i.e. full, incremental, cumulative, etc).

The backup retention determines how long the backup recovery points should be preserved.

Putting this together, the rate at which a deduplication system consumes storage is relative to the following:

- Data type
- Data churn rate
- Data growth rate
- Backup cycles
- Backup retention

Most deduplication systems look for common chunks in a backup stream and then compress these chunks before storing them. Thus, the compressibility of data also plays a role in how much deduplication storage will be consumed. However, this attribute is usually factored into the data type. The data type determines for the same set of attributes—data churn, growth, backup cycles and retention—how one data type deduplicates relative to the next. When sizing deduplication systems, the data type is used to determine what deduplication rate should be applied (higher for data types that are known to deduplicate better) for a backup of the given data type. These rates should be provided by the vendor of the deduplication system and should not be applied to other vendor solutions.

The data churn rate is typically represented as a percentage of the underlying data that is changed every day. For example, 5% of 1 TB suggests 50 GB of the data is churning every day and the deduplication system needs to be able to isolate this data, deduplicate it where possible (i.e. some of the data churn may be unique), and compress it.

In addition to churn, a backup can experience data growth. For the purpose of understanding deduplication consumption, data churn and growth are treated equally as they result in new data that the deduplication system must determine is common or unique.

The amount of data a deduplication system needs to process is relative to the backup cycle. The backup cycle determines how often a backup is taken and what type (i.e. full, cumulative, and incremental) of recovery point is stored. If we perform full backups of a 1TB dataset daily, then we are required to process 1TB * 84 days = 84TB worth of data in 12 weeks.

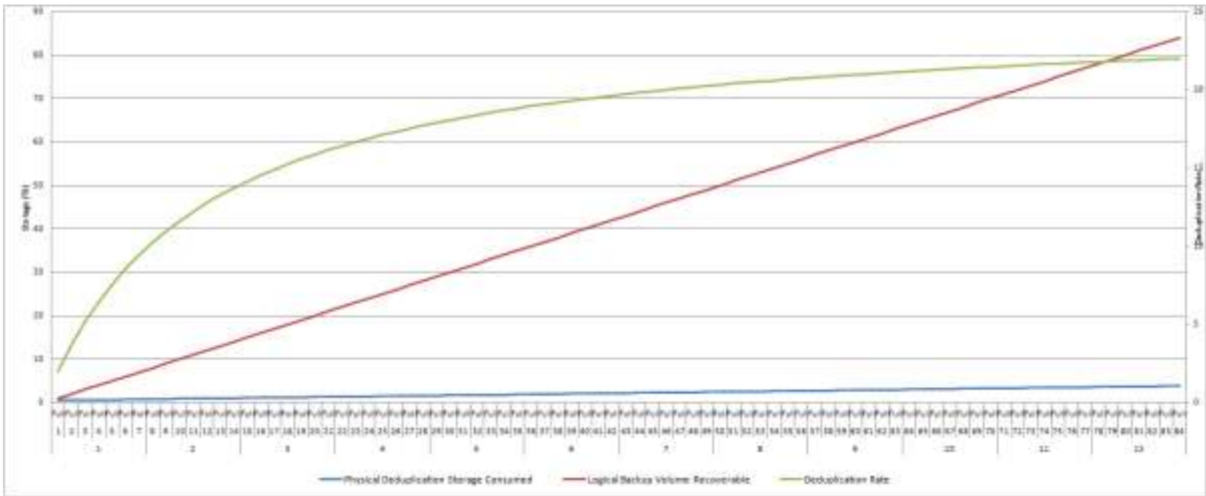
If we assume the first backup of 1TB does not benefit from chunking and only benefits from compression then the first backup consumes 500GB (2:1). On the following day there was 1TB processed of which 5% represented data churn. In most cases significant amounts of deduplication and compression will occur relative to the strengths and weaknesses of the deduplication system. However, for the 5% data churned we cannot expect the same levels and so we must take this into consideration when applying the deduplication rate for subsequent full backup. For the purpose of this example we will assume 25:1. At the end of day two the deduplication system is consuming 500GB for the first backup and 40 GB for the second. Logically, the amount of data recoverable is 2 TB and so after day two the deduplication rate becomes:

(2 TB) / (500 GB + 40GB) = 3.70x

Assuming the data churn rate continues at 5% per day and the data requiring backup experiences no growth, we can estimate the deduplication rate for the first week as:

(7 days * 1 TB) / (500 GB + (6 * 40GB)) = 9.45x

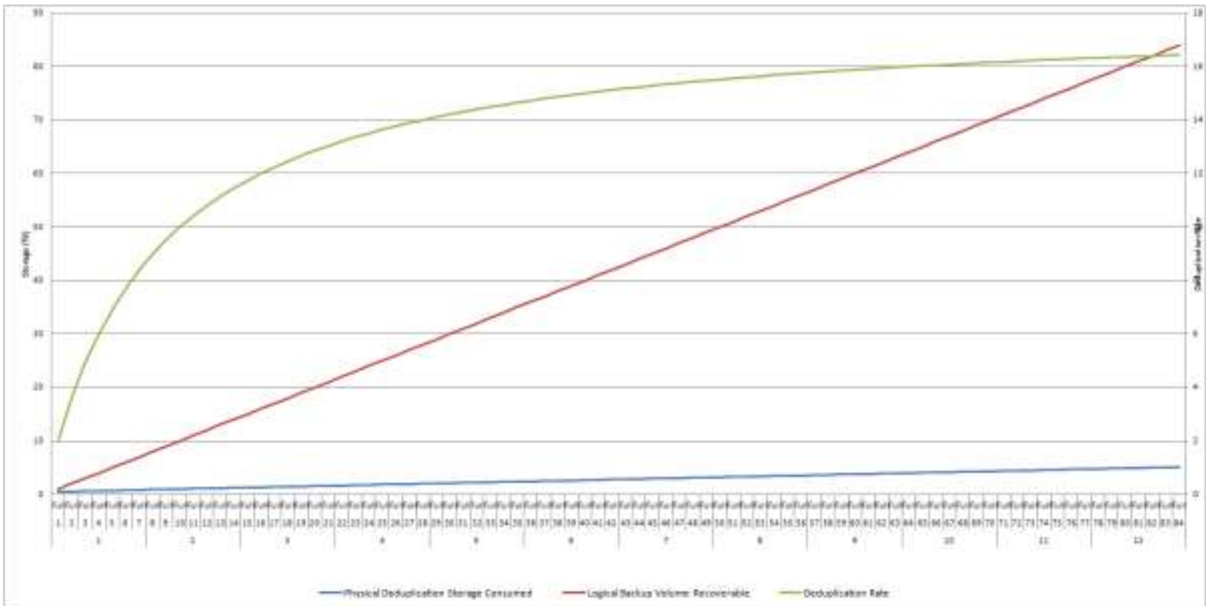
The chart below illustrates how storage is consumed over the 12-week period using daily full backups at 5% daily churn rate.



The deduplication rate is lower in the first week because the first backup had nothing to deduplicate against and consumed the majority of the deduplication storage. As each subsequent full backup is created, the deduplication rate starts to increase. With a 5% churn rate each subsequent full backup experiences approximately a 25:1 deduplication rate. After 12 weeks the overall deduplication rate reaches 21.99x.

If we adjust the data churn rate to 10% the resulting deduplication rate each subsequent full backup experiences drops as the expectation is more data churn results in more unique data. For this example we will adjust the rate to 18:1.

The following chart illustrates how storage is consumed over the 12-week period using daily full backups at 10% daily churn rate.



Deduplication rate is a function of the amount of data recoverable over the amount of data stored. For incremental backup workloads, deduplication rates decrease at the expense of retaining full recovery points. If recovering from a full is going to be quicker, consider exploiting this characteristic when moving from tape-based systems to deduplication systems for backup.

Thus far we have determined that deduplication systems—unlike primary storage systems—experience storage consumption over time that increases with data churn, data growth, and retention requirement. When thinking about how to manage and plan for capacity we must consider all these factors in order to ensure there is sufficient capacity available to accommodate the total backup environments' data and retention requirements.

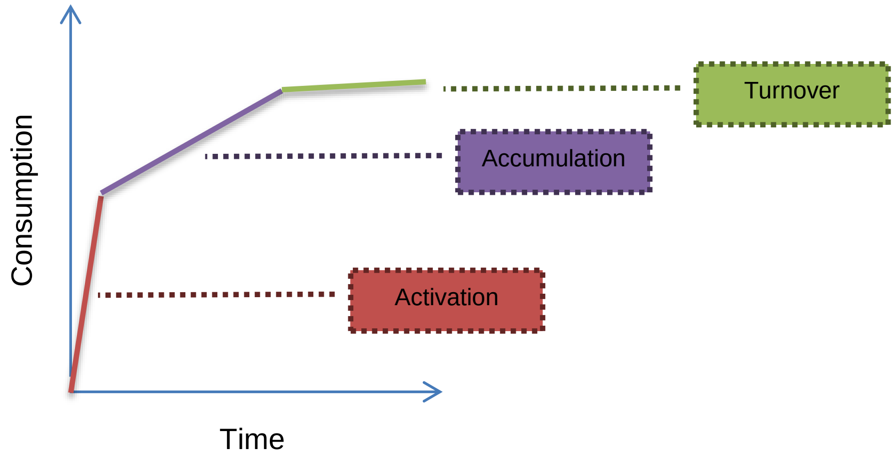
To understand this in more detail it is important to discuss at a high level the consumption phases a deduplication system experiences; activation, accumulation, and turnover.

The activation phase is experienced at the time a deduplication system is first used. In this phase the system experiences a tremendous amount of consumption as all backups that are sent to the system are experiencing relatively low amounts of deduplication (as there is no baseline to deduplicate against) and therefore are consuming storage rapidly. This cycle typically occurs for the first week since most systems will experience one full backup cycle within a week.

After the first week, the deduplication system enters the accumulation phase. In this phase the system is experiencing significant amounts of duplicate data. The storage consumed is predominantly driven by the daily data churn and any data growth. In this phase the consumption curve reduces substantially and typically follows a linear trajectory. The accumulation phase continues until the retention period for the data already backed up expires. This period could be short (i.e. 30 days) or very long (i.e. multiple years).

Once the retention period is exhausted, the deduplication system enters the turnover phase. This is where recovery points start to expire and the deduplication system is able to reclaim and return space to the system whilst subsequent backups consume space. This is also sometimes called the steady state where by the amount of data being reclaimed is relative to the amount of new data being consumed. In the turnover phase, the consumption rate reduces further to the point where if data sets are not growing and the churn rate is consistent then the system exhibits an almost flat consumption line. However, if the underlying data set experiences growth or higher churn, the system's consumption continues to grow.

The three phases are illustrated below.



This pattern represents a significant challenge for capacity planning since a deduplication system may be supporting backups for hundreds of clients which have different retention requirements, different data churn rates, different growth rates, and are activated during different times in the lifecycle of the deduplication system. In order to bring some order to this chaos, there are a number of techniques to experiment with to measure and plan the capacity requirements.

First and foremost, we have established that the consumption a deduplication system experiences is relative to the volume of data the system is supporting and the phases this volume of data is in relative to the deduplication curve. When looking at the overall capacity of a deduplication system it is near impossible to determine what portion of data is in which phase as that requires knowledge of the data types and policies which are typically transparent to the deduplication system. We need to combine backend statistical analysis with knowledge of what is changing in the environment to estimate how deduplication systems storage will be consumed over time.

For the statistical analysis it is important to establish both short- and long-term trends in order to forecast relative- to short- and long-term movements in consumption. These trends can be applied to several metrics independently, most importantly is the amount of deduplication storage consumed and the amount of virtual space (backups recoverable) this storage represents.

For the long-term trend it is best to establish the trend using data points that occur after the deduplication systems' initial activation phase, as this is considered a one-time event. The long-term trend should give a good indication over what period of time the capacity will be consumed given a regular pattern. In addition to the long-term trend it is important to establish a short-term view of capacity as we know that new activations or unanticipated changes in data churn or data growth can cause consumption to spike. In order to accomplish this, combining a short-term moving average with a medium-term moving average can help gauge the momentum and velocity of capacity that is being consumed. Monitoring consumption this way can provide an early indicator of abnormal or unanticipated behaviour by taking particular notice of the divergence between averages and crossover points. For example, if the short-term average is above the medium-term and the two averages are diverging away from each other this may be an indication that the system is experiencing significant new activations. However, if the short-term average is consistently on par with the medium-term average, this suggests the system is predominantly in the accumulation or turnover phase.

The suggested period for short- and medium-term moving averages is best determined by experimentation. However, one should ensure the medium-term moving average is not greater than the most common retention period and the short-term moving average is chosen based on the time capacity planning requires to react to a change in momentum.

In addition to these indicators it is also important to plot the amount of data (pre-deduplication) that enters the deduplication system within a given period. The length of this period should be aligned with a full backup cycle which most commonly would represent a calendar week. Plotting this indicator will provide evidence of new activations or data growth. Without these events, the full backup cycle should remain relatively constant week over week.

To determine what contributes to spikes in consumption one must supplement statistical analysis with data that is reported by the backup applications. The most important metrics to analyse from the backup applications are:

1. Amount of data backed up relative to previous backups of the same policy
2. Amount of data backed up from new policies or additions to existing policies

If after analysing 1) and 2) we determine there is no substantial change between current and previous backups or new policies or additions for a given time period, then this suggests a rise in consumption is due to a rise in data churn which would result in an increase in deduplication storage consumption. A significant rise in data churn can occur for many reasons (e.g. marketing campaign, merger and acquisition, and so on) and individual fluctuations do not create significant cause for concern. However, a constant rise requires further investigation, particularly if the data represents a large percentage of the population.

If data churn is not the cause, the most likely cause is data growth or new activations. To estimate consumption for data growth or new activations, capacity planning can establish some simple guidelines designed to anticipate and forecast future capacity requirements for new and existing backup workloads.

The simplest method to derive such guidelines is to establish a set of primary storage to deduplication storage consumption ratios. That is, for every TB of primary storage requiring backup we want to establish a ratio that determines how much deduplication storage this TB will consume to satisfy the backup requirements.

In the previous section we explained how the characteristics of data and backup policies impact deduplication storage consumption. We can use similar methods to derive ratios for the most common scenarios our environment is going to cater.

As an example, the most common backup requirement may be to back up databases in full on a daily basis and keep these backups for 12 weeks. Previously, we calculated that for 1 TB that experienced 5% data churn we required 500GB of deduplication storage on day one and 40GB for each subsequent full backup. From this we can calculate we would require in total 3.8TB after 12 weeks to store 84 full recovery points for 1TB of database data requiring backup. We can derive a rule that says for every 1 TB of database data requiring full backups for 12 weeks, we need to have provisioned 3.8 TB of deduplication storage by the end of the 12-week period. In other words, for this scenario we have established 3.8:1 deduplication storage to primary backup storage requirement ratio.

Similarly[6], if we assume a higher data churn rate then we should also assume a higher ratio.

It is important to note with the establishment of ratios one must determine whether all backups are going to be stored in the same deduplication pool, or whether backups will spill and fill into other deduplication pools. If the latter, it is important to reflect the inefficiencies of small deduplication pools to the ratio by applying typically a 30% overhead for each additional pool across which a backup is spread.

Establishing these simple ratios can help capacity planning estimate how backup requirements consume deduplication storage over time and provide the data point's

[6] More complicated backup scenarios can be defined that take into account mixed backup cycles, a variety of data types and retention requirements (e.g. daily incremental, weekly/monthly/yearly full). For these, it is best to work with your vendor to determine the most appropriate ratios based on the relative capabilities of the deduplication systems.

necessary to forecast requirements based on organic statistical analysis and inorganic consumption requirements for the life of a deduplication storage system.