Object Storage as a Backup Target

HGST Active Archive System Delivers for Commvault Data Platform

A DeepStorage Technology Validation Report

By Howard Marks and Enrico Signoretti

DeepStorage.net

Prepared for
HGST

About DeepStorage

DeepStorage, LLC. is dedicated to revealing the deeper truth about storage, networking and related data center technologies to help information technology professionals deliver superior services to their users and still get home at a reasonable hour.

DeepStorage reports are based on our hands-on testing and over 30 years of experience making technology work in the real world.

Our philosophy of real world testing means we configure systems as we expect most customers will use them thereby avoiding "Lab Queen" configurations designed to maximize benchmark performance.

This report was sponsored by our client. However, DeepStorage always retains final editorial control over our publications.

About the Authors

Howard Marks

Howard Marks is the founder and chief scientist at DeepStorage, LLC, an independent test lab and analyst firm specializing in data storage, virtualization, and data center networking. Before founding DeepStorage, Howard was a New York-based consultant for over 30 years, helping organizations—including BBDO, SUNY Purchase, and the Foxwoods Resort Casino—solve their IT infrastructure problems.

An entertaining and highly rated speaker, Howard speaks regularly at industry events, including VMworld, Interop, SNW, and Microsoft's TechEd. He has written three books and hundreds of articles on networking and storage technologies. He currently writes a column/blog at NetworkComputing.com.

Enrico Signoretti

Analyst, trusted advisor, and passionate blogger (not necessarily in that order). Having been immersed in IT environments for over 20 years, his career began with assembler in the second half of the '80s before moving on to UNIX platforms (but always with the Mac at heart) until now, when he joined the "Cloudland." During these years, his job has changed from deep technical roles to management and customer relationship management. In 2012, he founded Juku consulting SRL, a new consultancy and advisory firm highly focused on supporting end users, vendors, and third parties in the development of their IT infrastructure strategies. He is constantly keeping a vigilant eye on how the market evolves and is constantly on the lookout for new ideas and innovative solutions. You can find Enrico's social profiles here: http://about.me/esignoretti

Contents

The Bottom Line

While the tools and techniques we use change every decade or so, IT professionals will continue backing up their systems and data to protect them from the unexpected. While deduplicating backup targets eased the transition from tape to disk, backup object storage systems like the HGST Active Archive System may be a better solution for today's large backup users.

We tested the Active Archive System with Commvault Data Platform and discovered the following:

- *Native object/cloud support made integration simple*
- *The Active Archive System's erasure coding provides a higher resiliency than the RAID 6 found in most backup appliances*
- *The Active Archive System provides 3PB of useable space at 1/10th the cost of the leading appliance*
- *Users can further reduce costs by using Commvault Data Platform's data reduction*
- *We could back up nine clients to the Active Archive system at 8,500 GB/hr and restore at 4,000 GB/hr*
- *Performance with Commvault Data Platform data reduction was limited only by media server performance*
  - *Still over 2,700 GB/hr with three media servers*

Backups, Like Diamonds, Are Forever

While we like to think that the IT world is driven by Moore's law, which predicts ever-denser and, therefore, more powerful integrated circuits, the most important law for the IT operations group is the one first stated by Dr. Edsel Murphy: "Anything that can go wrong, will go wrong." Backups stand as our last line of defense against failures of both systems and humans.

Since we can never predict when a system will fail, be hacked, or even be stolen from a storefront remote office, we create independent backup copies so that, if anything escapes from our chamber of horrors, at least we can restore the data from a backup. Someone may someday create a system we trust to hold our data without being backed up, but we don't expect that to be the norm anytime soon.

Our collective fear of unexpected data loss means we'll be making backup copies of at least some of our data on a regular basis forever. Given that many organizations satisfy some or all of their retention requirements by retaining backups, it becomes clear that, at those organizations, the backup data is going to be stored forever as well.

When magnetic tape was the default backup medium, organizations could simply warehouse old backup tapes to satisfy the letter of any retention requirement. While disk-based systems offer many advantages as backup targets, systems designed to replace tape libraries as active backup/restore repositories aren't as economical as initially believed.

The Purpose-Built Backup Appliance

The transition from tape to disk as the primary backup target really started to gain steam around a decade ago with the introduction of data depulication, which made backup to disk affordable.

The backup administrator's standard operating procedure of the weekly full and daily incremental backup jobs creates a target-rich environment for data deduplication. This duplicate-rich data stream allows purpose-built backup appliances to reduce data as much as 10:1.

Even data deduplication must follow Heinlein's law, which states, "There ain't no such thing as a free lunch," and the cost for data deduplication comes in both CPU requirements and dollars. The leading vendor's appliances cost between $0.17 and $0.30/GB MSRP[1], even after 10:1 data reduction.

Even data deduplication must follow Heinlein's law: "There ain't no such thing as a free lunch."

The high CPU cost of data deduplication also limits the scalability of deduplicating appliances. Most vendors have multiple models with different capacities, forcing users who buy a system with 20TB of capacity to replace it with a larger model when their

[1] All prices in this technology validation report are MSRP.

data grows over 100TB. Even the top-of-the-line systems have less than 1PB[2] of useable capacity. As a result, large users must operate multiple appliances, which increases the cost of management and reduces the rate of data reduction, as each appliance becomes an independent storage pool and deduplication realm.

While the ingest speed of a deduplicating appliance is typically CPU-limited, restores are limited by the random-access performance of the appliance's disk drives. Since these systems are based on 7200 RPM disk drives, restore performance is typically just a fraction of ingest speed.

Backups Are Becoming More Efficient (Fewer Duplicates)

Today's deduplicating appliances are designed to wring every last byte out of the target-rich data stream created by traditional backup practices. As customers transitioned from tape—which, as a streaming medium, can be efficiently used for sequential access—to disk-based backup targets, new backup methods evolved that took advantage of the fact these new backup repositories could also handle random I/O.

> Backup performance with deduplication is usually CPU-limited.

These techniques reduce the amount of duplicate data the backup application creates, making deduplication less effective on their data streams.

Incremental Forever

Much of the duplicate data in a conventional backup stream comes from the system resetting each week by making a full backup of the protected system. Even an organization that retained its backup data for only 30 days would be storing a minimum of four copies of data in repeated full backups.

Incremental-forever systems use a database to track when each file on the protected system changes. The system can use this database to perform a point-in-time restore, to age out and overwrite files after their retention period has expired, and to export a synthetic full backup, that is, the set of files that would have been contained in a full backup when an incremental job was run.

Changed Block Tracking (CBT)

The incremental backup jobs used by conventional and incremental-forever backup systems use file system metadata to identify which files have changed since the last backup. Those changed files are backed up in their entirety regardless of whether they've been completely overwritten or only had a few bytes change.

Applications that create backups using vSphere's vStorage API for Data Protection make incremental backups at a much finer granularity by using snapshot technology to track changes at the block level. This finer granularity avoids repeated backups of those sections of files that remain unchanged over time.

[2] Useable capacity defined as after the data protection overhead of RAID, replication, erasure coding, and/or related technologies without compression and/or data deduplication applied.

As with incremental-forever backup systems, the backup application is responsible for keeping a database of changed blocks to allow access to the protected volume at any point in time and to catalog changed files to allow individual file restores.

Changed Block Tracking and Database Engines

Incremental backups have never been particularly useful for database applications, from Microsoft Exchange to Oracle or MySQL. Since the database is a single file, an incremental backup of a database server is still a full backup of the database itself. Strictly speaking, an incremental backup of a database backs up the transaction logs. Since replaying the logs forward in a restore can take several hours, most DBAs insist on full backups.

Since the backup engine will merge the changed blocks from the incremental backup with the rest of the protected disk's data at restore time, CBT-powered backups can be treated like full backups. This means that the backup application's VSS provider can perform log truncation, and the DBA can restore a database from a CBT-powered backup as easily as from a full copy. The advantage is that the CBT-powered backup writes a mere fraction of the amount of data a full copy would require, which not only saves space but makes the backup job run faster with less impact on application performance.

Copy Management

Most data-protection vendors have built their portfolios through acquisition. They bought their enterprise backup application, source data deduplication engine, and archiving application as independent applications, each of which created its own data repository.

Better integrated solutions, like the Commvault Data Platform (formerly Commvault Simpana) we used in our testing, leverage both storage system snapshots and a single copy of protected data that can be indexed for both backup/restore and content-based-archiving applications.

Backup Application Deduplication

While data deduplication entered the market on optimized hardware, it, like backup-to-disk support, has become a standard feature of backup applications. Source deduplication uses off-schedule CPU cycles on the systems being protected to save not only disk space on the backup target but also network bandwidth. Most solutions also support data reduction at the media server to protect systems that don't have spare cycles.

While purpose-built appliances may be able to claim higher compression ratios, backup application deduplication, combined with efficient back-end storage, can deliver at a significantly lower cost.

Enter the Object Store

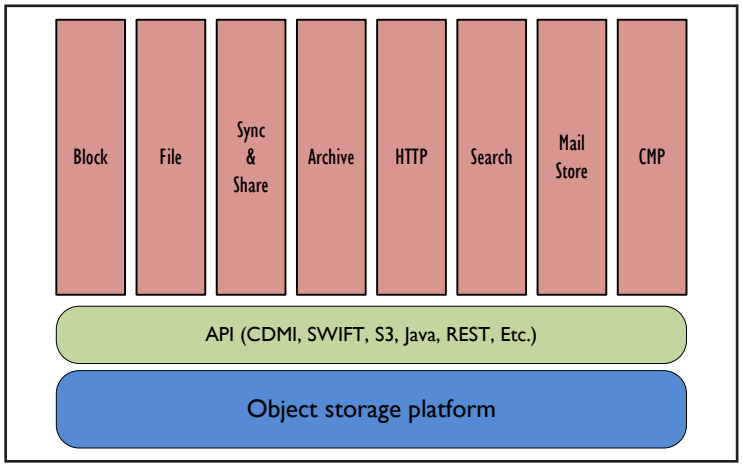
Object storage has been floating around the IT industry for a long time, but has only recently become commonplace, driven by web applications and cloud computing. The first service announced by Amazon AWS was S3 (Simple Storage Service), a public object store service that has seen constant growth in popularity among developers. S3 has been so successful that the number of objects it holds now number in the trillions.

Now, due to the data growth experienced by every IT organization, object storage has finally become popular, not only in hyper-scale environments but also in smaller infrastructures. Since object storage systems are almost always based on a scale-out design, they can start out relatively small and grow over time without concerns about scalability, resiliency, or data availability.

Why Object Storage

Data can be stored and accessed in different ways, and each one of these access methods has its own pros and cons.

Object stores are the most scalable from the standpoint of capacity, access concurrency, and throughput, but due to the nature of their design, they are not suitable for latency-sensitive workloads.



On the opposite side, block devices are the most suited for applications that need to access relatively small data sets as fast as possible, as in the cases of databases or virtual machines.

Traditional network file systems, which are very common, present the easiest way to access unstructured data from local networks, but they usually have many limits and constraints when capacity exceeds a few hundred terabytes. Scalability, resiliency, remote data replication, performance, and management complexity are all issues that contribute to unsustainable TCO figures, especially when consistent data growth is expected to continue.

Lately, object stores have reached a new level of maturity, and thanks to the ecosystem they can leverage, it is now possible to serve many different applications and user needs from within the same infrastructure. In fact, object storage systems can be considered horizontal platforms capable of simultaneously supporting several different workloads and data types. Thanks to direct application integration (through APIs) or specific gateways that can be deployed locally or remotely, using object stores can also be a solid solution to cover traditional access methods (NAS) in local and distributed environments.

How Object Storage Works

Object stores are intended to overcome the limits imposed by file systems and, more generally, traditional storage. They are designed for web-scale applications, multiple-9s of resiliency, and high availability, as well as unmatched efficiency, both in terms of automation and cost-per-gigabyte.

The atomic information stored in an object store is the object. It is composed of the information (usually a file) and a metadata set (which can be fixed or customizable). Objects are not organized in a hierarchical manner but in a flat space. They can also be grouped in what are usually called buckets.

> While purpose-built appliances may be able to claim higher compression ratios, backup application deduplication, combined with efficient back-end storage, can deliver at a significantly lower cost.

Buckets are the first and easiest way to define the limit of a domain space. However, since object stores are architected for the maximum multi-tenancy, individual product implementations utilize other methods to reserve resources for specific applications or users.

These elements have some important characteristics that make the difference when compared to other storage systems.

- Objects are immutable (there are put, get, and delete functions, but there is no modify operation). This has strong implications about safety and how applications can effectively access data.
- Data-integrity mechanisms are implemented to constantly check and validate object state, ensuring that reading from the object store is always valid.
- Data-scrubbing functions are implemented to actively enforce and maintain system and data integrity, and since all data is protected at the object level (not disks or nodes), for each individual failure, only the involved objects are rebuilt or copied to reconstitute their integrity.
- Modern data-protection techniques are implemented to avoid any data loss, even in worst-case scenarios. Multiple data copies—and, in the most sophisticated and efficient systems, erasure coding—are utilized to grant data accessibility, even in the case of a disaster or multiple failures, thanks to embedded transparent geo-replication capabilities.
- Policy-based automation and resiliency engines are integrated to ensure that retention policies and protection levels chosen by end users and applications are constantly met without human intervention.

Furthermore, all characteristics described above aren't enough to illustrate full object storage potential in terms of scalability. Modern object storage platforms are based on a distributed, shared-nothing cluster design. This design allows the systems to start relatively small and scale up to large multi-petabyte environments just by adding more nodes.

Different Types of Efficiency

Usually, when primary storage systems are involved, and considering a cost-per-gigabyte ratio of approximately $1/GB or higher, efficiency is associated with data footprint reduction.

Especially now, with the rise of all-flash arrays, data compression and deduplication techniques are heavily adopted for saving space on precious media, improving its durability through fewer writes and, as a consequence, taking advantage of optimized read/write operations.

In this case, however, data stored is usually uncompressed and easy to deduplicate, such as in the case of plain data files or databases.

This kind of efficiency also comes at a cost if latency must remain low and consistent. Most primary storage systems have limited scalability (generally under 1PB) and they are still based on scale-up or limited scale-out architectures, which don't contemplate single-domain, geo-distributed deployments.

Talking about efficiency in secondary storage infrastructures is totally different:

- Modern unstructured data formats are already compressed. Images, movies, and even word processor files are now efficiently compacted before leaving clients. This compression is done for several practical reasons, and now, recent CPUs implement many compression/decompression features, enabling real-time operations at virtually no cost.
- Data encryption is used more and more often in every organization. Security and privacy policies as well as laws and regulations require data to be encrypted while transferred and at rest. Again, this drastically reduces data reduction effectiveness.
- High-capacity hard disks are not good at processing random-access requests. Unlike flash memory, large SATA drives, already available in 10TB size, have a limited number of IOPS available and a much higher latency. On the other hand, they are very good with sequential reads and writes, and they show incredibly good cost-per-gigabyte and can deliver high throughput. Consequently, scattered access patterns and hash table management imposed by in-line deduplication aren't applicable without introducing limits and constraints to the overall system design.
- Modern backup software, such as Commvault Data Platform, which we used for our testing, is able to perform data deduplication and compression at the client or server level. This action at the server level has advantages that directly improve the overall scalability of the entire infrastructure: it limits the quantity of data transferred over the network between clients, servers, and backup repositories.

Efficiency and Erasure Codes

Object store data protection is usually performed on a per-object basis, which means that different protection levels can be applied at the same time to improve overall flexibility, data durability, and savings. The most common protection schemes currently used in object storage systems are data replication and erasure coding.

When data replication is applied, each single object is copied to different locations (nodes, racks, and/or data centers) to meet the defined protection policy. A minimum of three copies allow data to survive two failures, but more copies mean better data resiliency and availability.

Having multiple copies of the same object spread across different disks, nodes, racks, and data centers is very secure, and maintaining those copies uses a very small number of CPU cycles, but it's inefficient from a capacity-utilization perspective. In fact, storing five copies consumes five times the space of a single copy of that object, and despite the fact that the cost of a gigabyte of storage will continue to shrink in the coming years (by about 20% per year), the growth of data saved is predicted to continue to increase at a pace of 40% per year.

Erasure coding is the answer to obtain the best space utilization while maintaining a level of data protection similar to multiple data copies. An erasure code is a form of error correction in which a message (a chunk of data) is transformed to a set of data segments, or strips, such that the original message can be recovered by reading only a subset of the segments.

> Erasure codes provide higher levels of protection with less overhead than replication or RAID.

A given erasure coding scheme can be simply described by a ratio of the total number of segments generated from the original data to the maximum number that can be lost before the data becomes unrecoverable. This ratio defines the efficiency of the erasure code.

Here are a couple of examples:

- With a 10/2 ratio, an erasure code could be as efficient as RAID 6, meaning that the system can lose two segments before losing the information.
- However, if the ratio is 26/6 (and data chunks are intelligently spread in different fault domains), the system can sustain up to six different failures.

The trade-off for erasure codes is the amount of CPU cycles needed for calculating the segments, which is a much more complex math function than the XOR operation performed for RAID. On the other hand, new CPUs are faster than in the past, and manufacturers are starting to implement some of these functions in hardware. In any case, over time, with the continuous improvements of computing resources, erasure codes are becoming much more common, as well as applicable to more use cases.

In a traditional two-controller storage system, CPU can easily become a bottleneck, especially if compute resources are already heavily engaged with data-footprint optimization. In this case, erasure coding would severely impact both performance and scalability.

On the contrary, in modern object storage systems, thanks to the distributed design architecture, the following are true:

- Each single node has plenty of CPU power that can also be utilized to calculate erasure codes.
- Scalability is not impacted at all, due to object stores' scale-out nature, with each single new node added to the system bringing more CPU and disk space.
- Data segment distribution across the cluster (and even different data centers) brings unmatched availability and resiliency levels.

RAID 6 Limits and Constraints vs. Erasure Coding

Without questioning the specific RAID 6 implementation, this type of protection scheme shows several limits and constraints when compared to higher-order erasure coding, especially when it comes to very large data repositories that are rarely accessed after they are initially written, like backup or long-term archives.

Virtual tape libraries (VTLs) have never solved one of the problems still present in larger data centers: tape management. VTLs are not capable of scaling, and their cost is too high to maintain long-term backup archives.

RAID, no matter how it is implemented, has a long rebuild time when a disk full of data fails, which severely impacts performance. With next-generation hard drives, we are now in a situation where, due to the very long rebuild time, multiple fails are much more probable. For example, today, most purpose-built backup appliances still use 3- and 4TB drives instead of more modern 6TB, 8TB, or 10TB disks. Object stores that implement erasure coding and object-level data protection are not impacted by rebuild times of single or multiple disk failures. All nodes contribute to rebuild the consistency of failed erasure code segments on any other disk or node that matches the protection policy in place, and all in a matter of minutes.

Despite tapes still showing the best TCO for cold storage, object stores are getting closer and closer because they can now leverage high-capacity, power-efficient disks, along with the type of scalability that can't be found in virtual tape libraries. In fact, infrastructure simplicity plays a fundamental role in modern data centers, and object stores enable customers to build a single huge repository for all secondary data, avoiding expensive tiering and backup consolidation activities to move data from VTLs to long-retention archives.

In terms of infrastructure complexity and efficiency, the limited scalability of scale-up, RAID-based backup solutions poses another big set of problems and costs:

- Power and data center footprint—small hard drives consume the same, or more, power than larger and newer models, meaning that long RAID rebuild times can easily double power and space facility costs.
  - Object stores can use the largest and most efficient drives because they don't suffer RAID constraints and risks.
- Deduplication efficiency—adding more VTLs to overcome scalability issues also means separate deduplication domains and, consequently, reduced overall efficiency of the whole system.

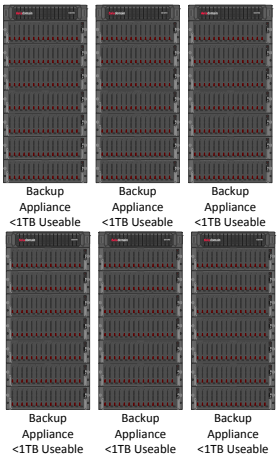
  - Some client/server deduplication solutions can be configured to have a single deduplication domain, increasing their efficiency with the growth of protected data.
- Remote replication efficiency (electronic vaulting)—VTLs use traditional replication methods to copy data between different data centers, which means a complete data copy for each individual site.
  - Object stores like the HGST Active Archive System can leverage geo-distributed erasure coding to optimize data distribution to multiple data centers, limiting space utilization while maintaining outstanding data availability and durability.
- Migration costs—although we are not talking about primary storage systems here, any upgrade that involves forklift upgrades to bigger or newer VTLs has some costs associated with it.
  - Scale-out object stores are designed for 100% uptime, and adding capacity or performing system upgrades is done one node at a time without impacting service levels or data stored in it.

Consolidation

Smaller environments are less affected by these kinds of issues, but in large data centers where consolidation and power savings are big concerns, it's possible to see many benefits from adopting object storage and erasure coding.

A well-designed, highly integrated, erasure-code-based object store can be very efficient in terms of both space and power consumption. A good example comes from the HGST Active Archive System, where a system configured with ~12PB usable space usually consumes approximately 30kW in just four racks.

At the same time, thanks to object storage characteristics, consolidation can drive down TCO even further. In fact, while other types of storage usually need a high number of operators/sysadmins, in the case of object storage the sysadmin/TB ratio is very low. Several petabytes per sysadmin are not uncommon for these types of systems, and thanks to erasure coding and its very high resiliency, it is also possible to rethink maintenance procedures. For example, in large object storage infrastructures, it's not uncommon to have only one day per month dedicated to disk and node replacement.



These benefits improve with the size of the system. The object store can be expanded by adding additional racks, and since most of its operations are policy-driven, the reconfigured cluster will automatically take full advantage of the new resources without necessitating user intervention.

This level of efficiency is unique in the storage market and hard to achieve. In the particular case of HGST, it depends on several factors: good erasure code implementation, state-of-the-art and energy-efficient hard drives, and very good system design. This is also the reason why most traditional enterprises still prefer pre-packaged appliances to DIY infrastructures when it comes to object storage.

Much More Than a VTL

Thanks to the specific erasure coding implementation and characteristics of the HGST Active Archive System, the product can be much more than a backup repository. Thanks to its multi-tenancy capabilities, it can also be considered a fundamental component for building a distributed storage layer suitable for many different needs, including content management (i.e. Microsoft SharePoint storage), active archiving, or even big data analytics. Standard API support (such as S3, for example) makes it also possible to leverage the product for implementing a private cloud storage back-end for sync-and-share or distributed NAS storage infrastructures (edge appliances in remote offices that don't need local backup).

In the last 12-18 months, vendors and end users have radically changed their points of view about object storage:

- S3 is now a common protocol for many more applications and storage gateways.
- Implementation of private clouds is driving private cloud-storage-based services, too.
- The economics of object storage are so strong that it is impossible to not consider this kind of repository for all secondary storage needs.
- With the growth of unstructured data (the last estimate by the International Data Corporation is approximately 62% per year) and longer retention policies, object storage is the safest bet for any type of organization.
- With new big data needs and the Internet of Things around the corner, object storage looks like the only option to store huge amounts of data coming from an undefined number of dispersed locations for later use.

All of these aspects, combined with the low cost-per-gigabyte for both acquisition and TCO and joined by the quick ROI coming from storage consolidation, make the investment in object storage a much better bet than any other single secondary storage system, especially considering its potential to enable new and innovative enterprise applications, including legacy VTLs.

HGST Active Archive System

HGST's Active Archive System evolved out of the Amplidata AmpliStor object storage system we tested a few years ago[3]. Amplidata was a pioneer in the use of high-level erasure coding to provide greater levels of data protection and storage efficiency.

The Active Archive System delivers 4.7PB of raw capacity in an industry-standard 42U rack and includes these components:

- 6 storage nodes
  - 98-drive SAS JBOD
  - 8TB helium-filled disk drives
- 3 controller nodes
  - 2 10Gbps Ethernet ports each
- 2 10Gbps top-of-rack Ethernet switches
- All of the optics, cables, interconnects, and PDUs to make installing the system in your data center simple

In building the Active Archive System, HGST has combined the object storage and erasure coding expertise that they acquired with Amplidata with the extensive base of knowledge they've amassed as one of the world's largest disk drive producers.

The JBODs in the Active Archive System, for example, not only manage to pack 98 large-form-factor disk drives into just four rack units, but they do it while maintaining the high level of mechanical engineering required to keep those disk drives running smoothly. The drives are vertically soft mounted to sleds of fourteen drives each to minimize the impact of vibration on the drives and maximize airflow. The JBOD is also designed to allow tool-less replacement of disk drives, fans, and other major components.

Host systems connect to the Active Archive System's controller nodes over 10Gbps Ethernet and access their data using Amazon S3 object storage APIs, which are becoming the de facto standard. The controller nodes distribute the data across the back-end storage nodes in the system and maintain the system's metadata on their internal SSDs. Users can combine multiple Active Archive System racks into a single system providing massive scalability.

Data in the Active Archive System is spread across the storage nodes in a system via an erasure coding method that can recover the original data from any thirteen of the eighteen data strips the system writes to its storage nodes. This 18/5 (eighteen total strips with five strips of redundancy so that data can be rebuilt even when five

[3] http://www.deepstorage.net/NEW/wp-content/uploads/2013/03/Amplidata-eXtreme-Performance-A.pdf

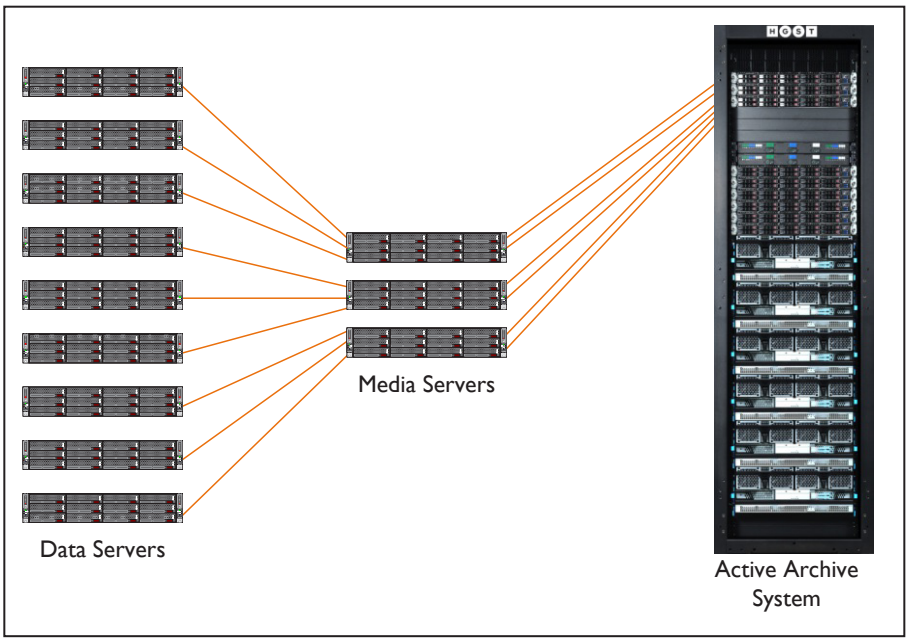
strips are lost) erasure coding allows the system to continue operating with multiple disk failures, providing what HGST calculates as fifteen 9s (99.9999999999999%) of durability.

Users with multiple systems in different locations can extend the erasure coding to spread data strips across three locations so that the system can continue to access data even if one of the three data centers goes offline. In this three-way geo-distributed configuration, the system uses 18/8 erasure coding and writes six strips to each of the three locations, allowing it to survive the failure of one data center and two disk drives before losing data.

HGST, by engineering the whole system—from the hard drives themselves to high-density JBODs and the software that erasure codes the data and retrieves stored objects—can deliver this complete system at a cost of under $300.00/TB or 30¢/useable GB, a price that purpose-built backup targets can match only if the data they store reduces 10:1 or more.

Testing the HGST Active Archive System

Our previous experience with the Active Archive System's progenitor, Amplidata's Amplistor, showed us that HGST's erasure-coded object storage technology could ingest



The test configuration.

large objects at close to wire speed. For this Technology Validation Report, we wanted to see both how the platform had advanced under HGST's tutelage and how it performed in a real-world backup situation.

While we usually perform our testing at DeepStorage Labs, where we can work independently, some products, including HGST's Active Archive System, are just too big to make it worthwhile to ship them to Santa Fe for a few weeks. Instead, we tested an Active Archive System that HGST maintains at a San Jose collocation facility. We performed most of the testing remotely and had access to the physical systems when needed, which we used to verify the system configurations.

Our test environment used a single Active Archive System as a backup target for Commvault Data Platform, which, out of habit, we wish we could just continue to call Simpana. We set up nine Linux hosts as our data sources. Those nine clients were backed up through three Data Platform media servers to the Active Archive System using Commvault's native S3 driver.

We then ran a variety of backup and restore jobs to see how well the Active Archive System performed as a backup target. We ran jobs that moved data directly from the source to the Active Archive System and with Commvault's software data deduplication enabled.

Performance Results

Our first set of tests measured the system's performance as we increased the volume of data we were backing up and restoring. Many backup targets can ingest data very quickly in aggregate across multiple simultaneous sessions but are limited in their single stream performance.

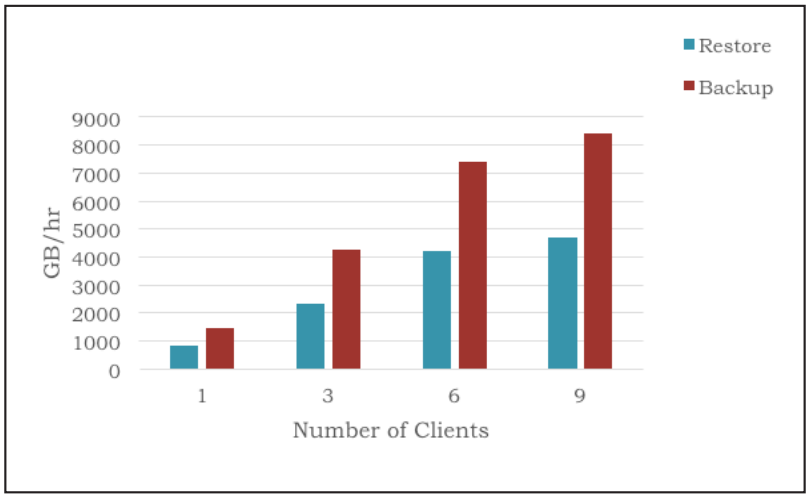


Chart 1. Backup/restore performance w/o deduplication.

We ran backup and restore jobs with increasing numbers of clients. The data rate peaked at 8,555GB/hr when backing up all nine clients, with restore performance just over half that, at 4,700GB/hr. Performance scaled pretty much linearly through six clients, but our full load of nine clients appears to be close to the system's total ingest capacity.

While at 30¢/GB the Active Archive System already provides 3PB of usable storage at an attractive price, we wanted to see what happened when we used Data Platform's internal deduplication so that we could save some more money.

Commvault Data Platform can deduplicate data at the media server, which minimizes the impact of the backup job on the client's CPU, or at the client, which uses more client CPU resources but minimizes network traffic between the client and media server. Client-side deduplication is also useful in hypervisor environments where hosts can become network-bound during backups.

Since data deduplication is compute-intensive, we expected backup performance to be limited more by the media servers than by the Active Archive System. Regardless of whether the data is chunked and hashed at the client or at the media server, the media

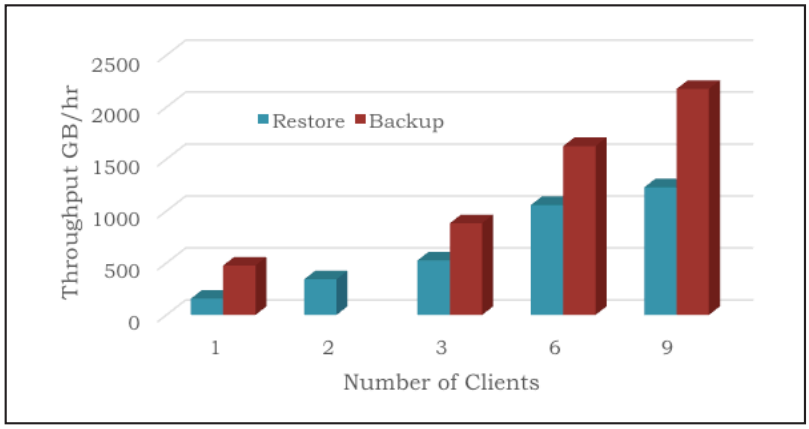


Chart 2. Backup/restore performance with data deduplication.

server must perform the compute-intensive hash lookup to determine if the data is new or duplicates existing data. As a result, we didn't see any significant performance difference between media-server- and client-side deduplication.

We managed to push 2,175GB/hr of backup traffic into the Active Archive System with Commvault data deduplication enabled and restored at 1,228GB/hr. As with any deduplicating system that stores its data on disk, single-stream restore performance is limited by the back-end storage, in this case, the Active Archive System's ability to process random I/O as the data is read.

Users seeking higher performance on individual restores can enable Commvault Data Platform's optional archive/dedupe layer, which would use a small amount of high-performance storage as a "landing zone" for the most recent backup data. This would allow Commvault Data Platform to use the high-performance disk as an immediate dedupe target during backups, and then, when the data has been deduplicated, migrate the data to the Active Archive System where it can be stored at a much lower cost. Conceptually, this is similar to a disk-to-disk-to-tape backup scheme, using a higher speed intermediate layer to speed access to the most recent backup data.

Conclusion

The HGST Active Archive System as a Backup Target

Now that backup applications support writing directly to object storage, eliminating the need for problematical and costly NAS gateways, users should seriously consider using a local object store as their primary backup target.

Whereas purpose-built backup appliances are really only useful for backup data, organizations can use a single multi-petabyte object store for archival and other applications as well as for backup. The largest purpose-built backup appliances top out at under a petabyte of capacity, forcing many user organizations to use multiple appliances and, therefore, multiple independent pools of data.

Each Active Archive System delivers 3PB of useable capacity at a cost of roughly 30¢/GB, verses $3/GB for the leading backup appliances. As data in most organizations' backup streams is producing fewer opportunities for deduplication, it becomes less and less likely that those purpose-built appliances can achieve the 10:1 data reduction they would need to simply achieve price parity with the Active Archive System.

Even if users have to invest in more powerful media servers and additional software licenses to enable data deduplication, those are small investments compared to the cost of a purpose-built appliance. With software deduplication, the Active Archive System can effectively cost 10¢ or less per gigabyte.

Using the Active Archive System as a large-scale backup target will also bring operational savings from running one large storage system rather than three to five backup appliances. Since it's a scale-out system, new higher-density nodes can be added to the system and old nodes retired, eliminating the need to migrate data from old to new systems every few years.

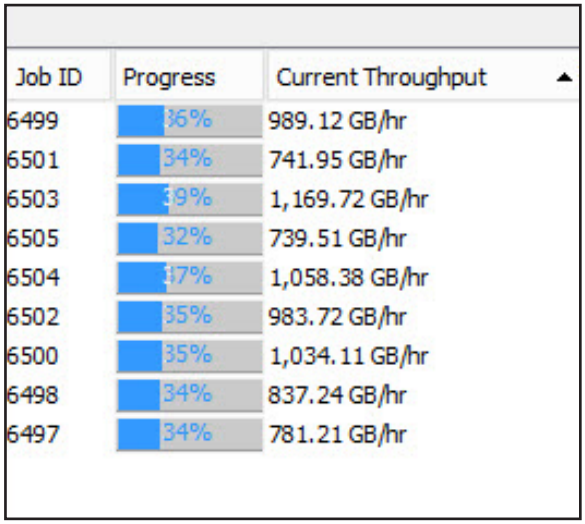
Appendix: How We Tested

Our testing was conducted over two weeks in October and November of 2015 at HGST's San Jose colocation facility. HGST's personnel prepared the Commvault Data Platform environment, and we confirmed that the system was configured as we requested before testing.

The test environment consisted of nine (9) Linux servers acting as data sources and three media servers. All were connected via 10Gbps Ethernet links to the Active Archive System under test.

We used the Commvault Data Platform backup engine to back up data from and restore data to our client systems. We used a 156GB dataset and either backed up or restored the full dataset with each job.

We discovered that restore jobs ran significantly faster when we eliminated the file system metadata updates caused by overwriting files one by one and erased the data from the system before restoring it. All reported restore performance is for restores to systems where the data has been erased.



Job ID	Progress	Current Throughput
6499	36%	989.12 GB/hr
6501	34%	741.95 GB/hr
6503	39%	1,169.72 GB/hr
6505	32%	739.51 GB/hr
6504	47%	1,058.38 GB/hr
6502	35%	983.72 GB/hr
6500	35%	1,034.11 GB/hr
6498	34%	837.24 GB/hr
6497	34%	781.21 GB/hr

Backup throughput with nine running jobs.

Of course, backup software does have its limitations for benchmarking. Jobs start over a period of up to several minutes, and these applications only report the average throughput over the job's duration. Luckily, Commvault Data Platform reports throughput to the screen for all running jobs once a minute. This cadence allowed us to take a screenshot like the one below to capture the throughput.

We believe that this one-minute-average throughput is more representative of the back-end storage system's ability to ingest data than the longer average over the full job. Commvault doesn't report restore throughput as jobs are running, so we used the full job average from Data Platform's reports.

Server Configuration

We used nine data servers (which Commvault calls clients) and three media servers:

- 2 Xeon E5-2680 v2 (10 cores @ 2.80GHz) processors
- 64GB memory
- 10Gbps Ethernet ports
- 256GB SSD used for all backup and restore jobs

Data Sets

Most tests were performed using a dataset of mixed file sizes, created to be 25% duplicate data. Each client had a 156GB set of files:

- 1 MiB – 5%
- 5 MiB – 10%
- 30 MiB – 25%
- 50 MiB – 40%
- 100 MiB – 10%
- 1024 MiB – 10%

Commvault Configuration

All of our data source servers (or clients) and the three media servers had Commvault Data Platform, formerly known as Simpana, 10 installed by HGST's lab staff before our arrival. While these systems were tuned for performance, we did not employ any extraordinary tuning or tweaking of the systems. Backups were made directly to the Active Archive System without the use of an archive/dedupe layer, which could have improved deduplication performance.

Commvault Data Platform Tuning

Storage Policy			
	Value	**Location**	**Description**
Chunk Size	4096MB	Policy copy > Data Path property > Chunk Size	
Block Size	2048KB	Policy copy > Data Path property > Block Size	
Device Streams	120	Storage Policy>Properties>Device Streams	The maximum number of streams a storage policy will open.
Number of Data Readers	12	Subclient property > Number of Data Readers	The maximum number of files the client will transfer in parallel. Ex, with number of device streams at 120 and number of readers at 12, the storage policy can support 10 parallel clients using 12 streams each.

Media Agents			
	Value	**Location**	**Description**
nCloudUseTempFile	0	MediaAgent > Properties > Additional Setting > Add	Use memory buffer instead of a temp file.
nCloudMaxSubFileSizeKB	32768	Same as above	The maximum size in KB for an object.
nCloudNumOfUploadThreads	12	Same as above	Maximum number of upload threads on a per-file basis.
nCloudNumOfReadAheadThreads	2	Same as above	Maximum number of read-ahead threads for upload on a per-sub-file basis.
nCloudNumOfReadAheadFiles	4	Same as above	Maximum number of read-ahead threads for Download on a per-sub-file basis.
nCloudSocketSendBufferBytes	1048576	Same as above	Send bytes that can be buffered before flow control is imposed.
nCloudSocketReceiveBufferBytes	1048576	Same as above	Receive bytes that can be buffered before flow control is imposed.