White Paper

# EMC DATA DOMAIN ARCHIVER

## A Detailed Review


### Abstract

This white paper introduces the EMC® Data Domain® Archiver, the industry's first long-term retention system for backup and archive data, and explains how it delivers system throughput and resilience advantages typical of Data Domain systems along with new optimizations for cost-efficiency, fault-isolation and improved data management.



October 2011

Part Number h8140.2

# Table of Contents


Executive summary .... 4

- Introduction .... 5
- Audience .... 5

Data Domain Archiver Overview .... 5

- Data Movement Policies .... 8
- Single Scalable Filesystem .... 9
- Technical Implementation .... 10
- Fault Isolation and Granular Recovery of Archive Units .... 12
- Granular Recovery from a Replica .... 14
- Cost-Effective Scalability .... 15
- Future Architectural Extensions .... 16

Typical Deployment Scenarios and Best Practices .... 16

- Long-Term Retention of Backup Data .... 17
- Archiving and Tiering Data to the DD Archiver .... 18

Comparing DD Archiver to Tape for Long-Term Retention .... 19

Conclusion .... 19

## Executive summary

Many companies have minimized the use of tape automation in their IT infrastructure by deploying deduplication storage for backup and operational recovery - EMC Data Domain deduplication storage systems have been the market leaders in this category. In general, operational recovery includes retention periods from a few weeks to a few months. For longer retention requirements, many users simply keep backups on tape for longer and some use specialized disk-based archive approaches.

While tapes offer portability, redundancy and fault-isolation, there are several drawbacks to using them for data retention:

- The perceived capital cost advantages of tape cartridges are often outweighed by the costs of tape automation and operating expenses;
- Accessing a retained file stored on tape takes a significant amount of time, especially if it is offsite;
- Evolving tape drive technology can require expensive and time-consuming bulk migration of old tapes every 5-7 years.

An alternative to using tape and backup processes for long-term retention has been to use specialized archiving processes or applications to move data from primary storage and its corresponding backup process to an archive platform as shown below in figure 1.

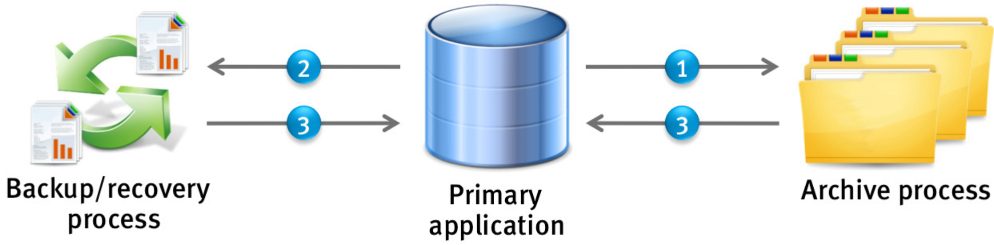


Figure 1: Best Practice: Archive before Backup: 1) Archive valuable information to a tiered infrastructure, 2) Backup active production information to disk, and 3) Retrieve from archive or recover from backup.

By implementing archiving processes, primary storage is freed up, data retention is accomplished, and backup processes can run more efficiently. However, this alternative has not become nearly as universally adopted as backup due to perceived complexity, lack of uniform policies or mandates, budgetary restrictions, software limitations and many other reasons.

As a consequence despite the known operational challenges, keeping tape backups longer has become the most widely adopted alternative for long-term retention.

As backup and archiving processes evolve, there is a need for a platform that bridges the common practices of today with the best practices of the future. During this evolution, customers may move away from using their backup infrastructure for retention purposes in favor of deploying archiving processes, but the timeframe for this transition is still unknown. For now, the requirements on the storage platform need to include support for quickly ingesting backup data (high throughput) and retaining it cost-effectively with a strong retention model.

The EMC Data Domain Archiver is the industry's first system for long-term retention of backup and archive data, supporting today's use of backup processes for data retention and the evolving adoption of archiving workloads. DD Archiver extends a mid-range Data Domain system configuration, which is already widely deployed and proven in backup environments and transparently incorporates a large-capacity tier dedicated to static data. New architectural enhancements allow the system to incorporate very large capacities that can expand over time, reduce system cost, and ensure long-term availability and integrity of data. When archiving software approaches do fit the environment, users can target their archive workloads to the same DD Archiver in the same infrastructure.

## Introduction

This white paper introduces the EMC Data Domain Archiver and explains how it delivers throughput and resilience advantages typically found in Data Domain systems along with new optimizations for cost-efficiency and fault-isolation. Read this white paper to find out how DD Archiver supports today's common practice of using backup processes for data retention and the evolving adoption of archiving best practices.

In the following sections, this paper will describe the unique characteristics of the DD Archiver, including data movement, fault isolation, planned architecture enhancements as well as comparisons to other long-term retention options.

## Audience

This white paper is intended for EMC customers, technical consultants, partners, and members of the EMC and partner professional services community who are interested in learning more about the Data Domain Archiver.

# Data Domain Archiver Overview

The Data Domain Archiver addresses the long-term retention requirements of scale and cost by extending the proven Data Domain architecture with an internal tiering approach. There are two tiers, sharing a common controller, management and namespace:

- An ***active*** storage tier, which operates like a standard Data Domain system used for operational recovery;
- An ***archive*** storage tier, to which files are migrated internally when a policy threshold (time of most recent modification) is passed for long-term retention.

These tiers are logical divisions of the storage attached to a single controller. Data is stored first to the active tier, and once it has been on the active tier without modification for a user-defined period of time, it is moved to the archive tier. The archive tier incorporates additional fault isolation and granular recoverability, larger capacity for archival data, improved manageability and enhanced compression.

Since data is separated and stored in different tiers, the system can scale to a very large capacity for the archival data. This new architecture with larger scalability behind a single Data Domain controller lowers the average cost of the system.

The DD Archiver is a separate system, and not an optional expansion to a standard Data Domain system. However, it leverages the architecture of traditional Data Domain systems with a single controller running a standard version of the Data Domain Operating System (DD OS), which connects to a set of standard Data Domain storage shelves. In addition to long-term retention of backups, customers looking to incorporate archiving use cases (e-mail, file, projects, tiering) can cost-effectively incorporate these data types on a DD Archiver.

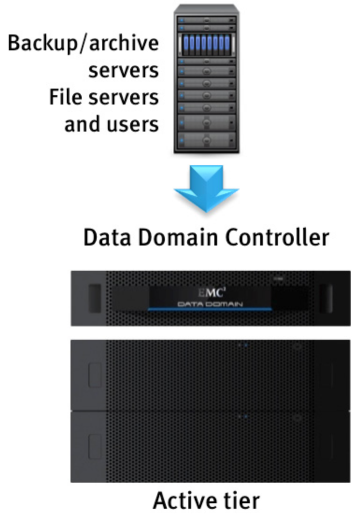


Figure 2: Common components of a traditional Data Domain system that are leveraged by the DD Archiver

**Storage tiers of the Data Domain Archiver.** The DD Archiver ***active tier*** operates essentially the same way as a standard Data Domain system and is sized based upon the same guidelines. For example, one could size the active tier to hold weekly fulls and daily incrementals for up to 90 days. The active tier can be configured with 1 to 6 storage shelves, with up to 142TB usable capacity.

The ***archive tier*** consists of one or more subsidiary ***archive units***. Each archive unit is a logical grouping of storage shelves, with its own deduplication context. Each archive unit can have between 1 and 6 storage shelves, with up to 142TB of usable capacity.

A user-defined, policy-based process runs periodically to move aging data out of the active tier and into the archive unit currently being filled. It continues to do this until that archive unit is determined to be full. See Figure 3, which shows data being moved from the active tier to an archive unit in the archive tier.

Tiers and archive units are transparent to end-users and applications, which see DD Archiver simply as a larger Data Domain system.

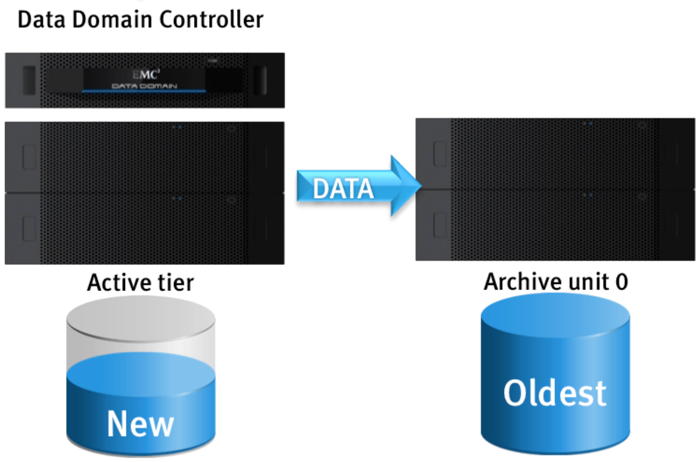


Figure 3: A Data Domain Archiver with its controller, active tier, and one archive unit in its archive tier. The internal data movement process is shown moving aging data from the active tier to the archive unit currently being filled.

**Filling Archive Units sequentially-** DD Archiver offers a unique fault isolation capability that seals archive units as soon as they are full. Each unit is then a completely self-contained unit of data preservation (covered in detail below). Once an archive unit is full and sealed, the system moves on to the next available archive unit, filling it with newly aged data. In this fashion, data is laid out onto archive units in age order as shown in figure 4 below. The active tier holds the most recent active backup data, such as incremental backups, while the archive tier holds relatively static data, such as archived full backups from months or years ago. This allows the archive tier to scale up to four times the capacity of an equivalent traditional Data Domain system with the same type of controller.

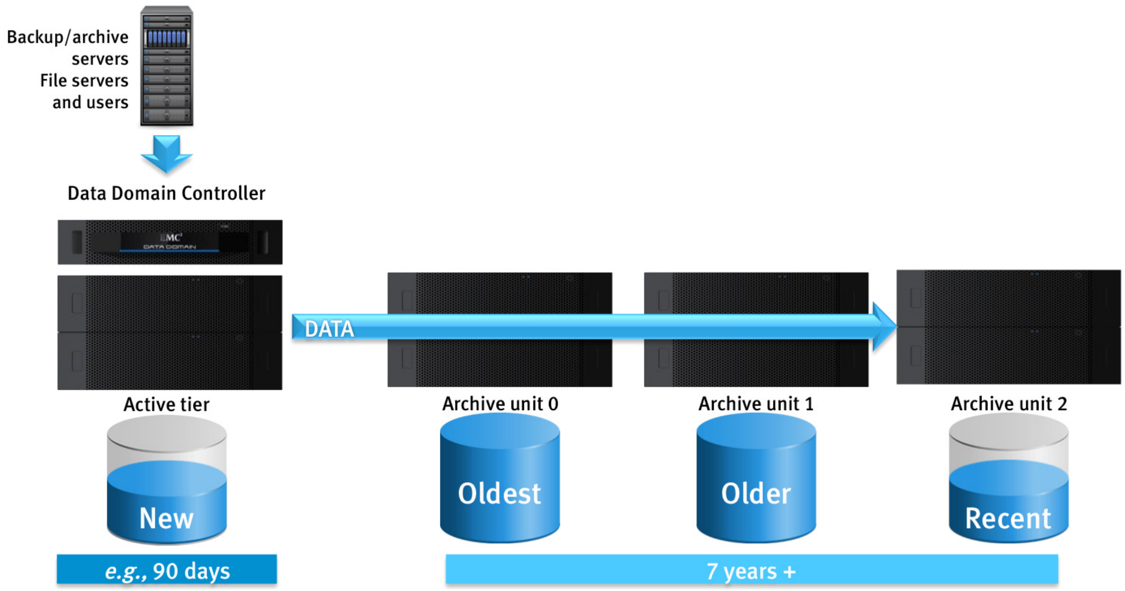


Figure 4: A DD Archiver with all of its components. Archive units 0 and 1 are full and sealed. Archive unit 2 is currently being filled with newly aged data moved from the active tier.

## Data Movement Policies

The system administrator can configure the policy that guides data movement from the active tier to the archive tier of a DD Archiver. Data movement policies are managed on an *mtree* basis and only those files that meet the policy threshold in that mtree are moved to the archive tier. An mtree is a logical partition of the namespace in a Data Domain filesystem that can be used to group a set of files for management purposes, e.g. for a distinct snapshot schedule that applies to a set of files associated with a specific use case (for more information on mtrees, please refer to the Data Domain User's Guide, 5.0 or later). If different data sets require different data movement policies in the same DD Archiver, they should be placed in different mtrees.

The data movement policies are driven by two user-defined elements:

- a selected last-modified-time of a file (e.g. after 90 days without modification);
- a periodic schedule for moving the data (e.g.. every week before the cleaning process).

During the data movement process, files are moved out of the active tier and into the archive unit currently being filled. Note that each file is only moved once. Since the data movement process is a background process that can be scheduled to run when the system is relatively idle, there is opportunity to apply further compression on this data. Specifically, during data movement the system can implement more processing-intensive compression algorithms, so that when data is moved out of the active tier, it can be recompressed and packed more tightly into the archive tier as it is moved. This enables even greater storage efficiency in the archive tier.

The data movement process runs on a periodic basis and moves individual files from the active tier to the target archive unit based on a user-defined file age policy. However only segments unique to that archive unit are physically copied, a process similar to the EMC Data Domain Boost's managed file replication feature (see EMC Data Domain Replicator White Paper for details).

The periodic data movement schedule initiates the data movement policies in a very similar way to that of the "cleaning" space reclamation process found in Data Domain systems: the data movement process can be scheduled to run at a specified time; it can be stopped, restarted or throttled. It is recommended that the data movement process run just ahead of the cleaning process so that all available space in the active tier can be reclaimed as soon as possible after the long-term retention files have moved to the archive tier.

For example, in the case of long-term retention of backups, a user may keep weekly fulls and daily incrementals for the duration of the backup cycle (e.g. 90 days) and after that period, the user may want to keep only monthly full backups for a specified long-term retention period (e.g. 7 years). To configure this data movement policy, an administrator sets the data movement process for this mtree to run every week and to move all files that were last-modified 90 days prior to that date. The backup policy retention settings can be set from the backup application to ensure only monthly fulls are kept beyond that date. During each weekly data movement sweep, the remaining files aged beyond 90 days will be moved to the archive tier.

## Single Scalable Filesystem

Just like traditional Data Domain systems, the DD Archiver presents a single, filesystem across all storage to enable ease-of-use and simple management. Therefore, the DD Archiver does not expose the different units and tiers and simply looks like a much larger Data Domain system to end-users and applications.

Of course, this single filesystem can be completely or partially exposed as CIFS shares (for Windows), NFS mount points (for Unix/Linux), and/or a specialized backup target through DD Boost for EMC Avamar, EMC Networker, Symantec NetBackup or Backup Exec. The EMC Data Domain Virtual Tape Library software option will be available for use with DD Archiver in a future release.

For very large implementations holding a lot of old data, the frequency of access to aging data is expected to decline significantly over time. The Data Domain Archiver can scale to store petabytes of data without any read performance degradation except when the system approaches its maximum capacity. At that point, it optimizes access of the more recent data. Therefore, reads of very old data in a fully expanded system may experience a slight delay when first accessing data stored in the oldest archive units. This is due to the fact that not all archive units' metadata will remain in the memory of the system's controller and as the system approaches the maximum capacity, one or two units' metadata are swapped out of memory. Because data is written onto archive units in age order, the system tracks the access frequency of each unit and can de-prioritize units with the oldest, least accessed data to ensure the lowest probability of a potential delay. However, even when the unlikely event of accessing a swapped-out archive unit occurs, the delay to first-byte is only approximately 30 seconds. During that time, the unit being read is brought into the controller's memory in favor of another unit and every subsequent read after that will occur at standard performance.

Since DD Archiver is aimed at further tape minimization, most users considering the system compare this low-probability delay - that only happens when the system reaches maximum capacity and only affects the oldest, least accessed data - to the delay of accessing data from offline tapes. Offline tapes are typically sitting on a shelf or have been shipped offsite to an external facility. Therefore, when comparing 30-second latency for access to the first byte of a file in a swapped-out unit to the hours or days of delay of getting an off-line tape into a tape library to access its data, DD Archiver is clearly a favorable option.

## Technical Implementation

Within a traditional Data Domain system, there are several levels of logical data abstraction above the physical disk storage, as illustrated in Figure 5.

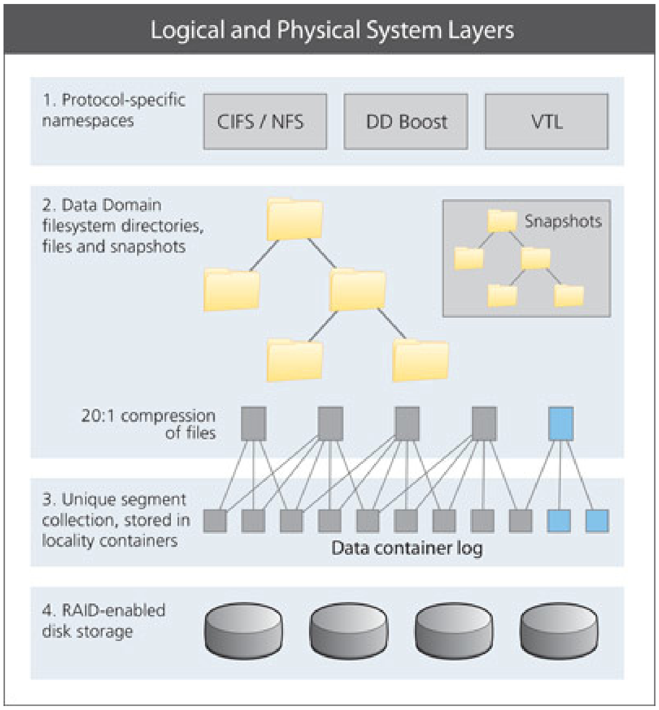


Figure 5: Data Domain Operating System filesystem – protocol-specific namespaces are presented to clients/applications for accessing the logical filesystem layer. The files and directories in each mtree, as well as mtree snapshots, all reference the same pool of unique segments, called a collection, which is made up of log-structured containers that organize the segments on disk to optimize throughput and deduplication effectiveness.

This figure illustrates several components and layers of the Data Domain Operating System filesystem:

1. **Protocol-specific namespaces:** As an external interface to applications, there are protocol namespaces, such as CIFS/NFS file shares (over Ethernet), virtual tape libraries (over Fibre Channel, not currently available in DD Archiver) and DD Boost storage units (SUs). A Data Domain deployment may use any combination of these simultaneously to store and access data.
2. **Filesystem MTrees, directories, files and snapshots:** Files and directories for each namespace are stored in an MTree in the DD OS filesystem. In non-CIFS/NFS cases, data (such as virtual tape cartridges) are stored as files under special directories. MTree snapshots in DD OS are logical and very space-efficient because they share the same underlying data segments.
3. **Unique segment collection, stored in locality containers:** A 'collection' is the set of files (or virtual tapes) and logical mtree snapshots. The system identifies and eliminates duplicate segments within each container and then writes compressed deduplicated segments to physical disk. Segments are unique within the collection (not including specific duplicates maintained in DD OS to enable self-healing or fast recovery). Each Data Domain system has a single collection that is stored in a log of segment locality containers. For more about segment localities, see the white paper, Data Domain SISL™ Scaling Architecture.
4. **RAID-enabled disk storage:** These collection containers layer over RAID-enabled disk drive blocks to ensure a high level of integrity. Data Domain deduplication storage systems use Data Domain RAID-6 internal disk and storage expansion shelves to protect against dual disk failures.

In addition, Data Domain Archiver introduces a few architectural extensions. Figure 6 illustrates how the DD Archiver incorporates an archive tier composed of archive units.

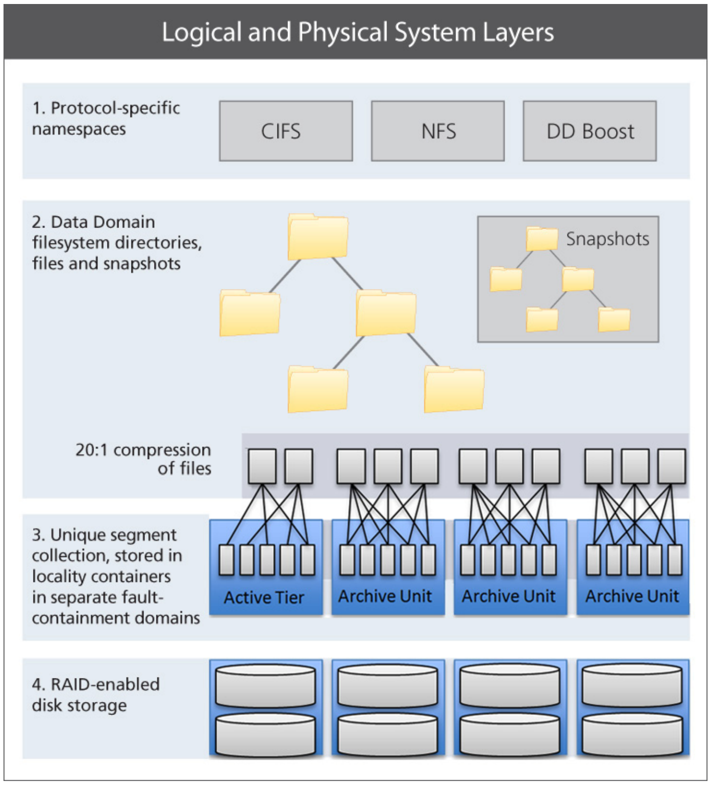


Figure 6: Data Domain Operating System filesystem – protocol-specific namespaces are presented to clients/applications for accessing the logical filesystem layer. The files and directories in each mtree, as well as mtree snapshots, reference separate fault-containment domains of unique segments, called collection partitions, which are made up of log-structured containers that organize the segments on disk to optimize throughput, scalability, deduplication effectiveness, and introduce a degree of fault isolation.

The architectural innovations in the DD Archiver are reflected in layers 3 and 4 of Figure 6. In order to implement its internal tiering, large scalability, and fault isolation capabilities, the DD Archiver divides the storage representation of a collection into collection partitions that are used as the active tier and archive units. Physically, each collection partition, whether the active tier or an archive unit, is stored in a corresponding logical set of storage shelves with RAID-6 protection. In the archive tier, each storage unit has its own deduplication index. New writes to the active tier do not check against these archive unit indices. With this design, data movement from the active tier to archive units can happen transparently without any visible change in the namespace of the moved files. This architectural design also allows for possible future extensions, such as drive spin-down in the archive tier. However, units and tiers are invisible to applications or users. Externally, all the manageable elements of the namespace look like a traditional Data Domain system.

## Fault Isolation and Granular Recovery of Archive Units

The DD Archiver is designed as a long-term retention platform for backup and archive data, and as such it is expected to grow to a large scale and operate for a much longer lifespan than traditional storage systems. To optimize for long– term storage, the system design is focused on fault isolation and granular recovery.

As noted above, DD Archiver is protected by Data Domain RAID-6 functionality to protect any storage shelf from up to 2 simultaneous disk failures while maintaining availability. Additionally, by leveraging DD OS, DD Archiver is protected by the EMC Data Domain Data Invulnerability Architecture. Data Domain Data Invulnerability Architecture includes continuous fault detection and self healing for all stored data, checking data integrity and preventing long-term effects (like bit rot) from ever affecting data integrity. For more information, see the EMC Data Domain Data Invulnerability Architecture white paper.

In addition to RAID-6 and Data Invulnerability Architecture, the DD Archiver offers the unique capability of fault isolation by sealing archive units. Sealing an archive unit consists of storing a snapshot of the filesystem and deduplication metadata into each archive unit once it is at full capacity. Once a unit is sealed, no more data is written to it and all subsequent data movement policies are then sent to the next available archive unit. Since each unit is sealed for fault-isolation, all archive units have separate deduplication indexes from other archive units and the active tier. Although fault-isolation comes at the expense of deduplication efficiency across the system, stronger compression algorithms applied to the archive tier and the cost-optimized scalability of the DD Archiver compensate for the cost of fault isolation with the added benefits of long-term data access and recoverability.

If the DD Archiver experiences an issue or there is a minor catastrophe at the primary site, which caused an archive unit to become unavailable, the system would continue to operate with all unaffected components. Other filesystems in a similar situation (experiencing a major component failure, beyond a single RAID group) would be completely unavailable and the user would likely experience partial or total data loss. With DD Archiver, the system is up and available and all unaffected data is still accessible. The system continues to operate as long as the controller and the active tier are functioning.

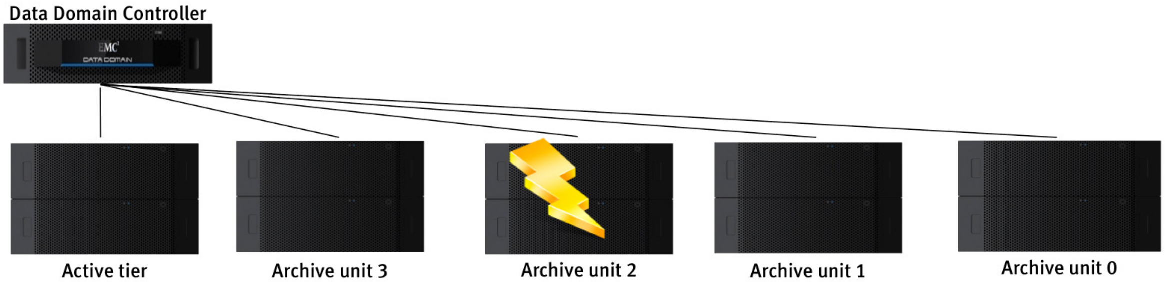


Figure 7: A DD Archiver sustaining a fault affecting archive unit 2. All unaffected data continues to be available and system continues to function.

When such a fault or failure occurs as depicted in Figure 7, one of three things can happen:

- If it is a minor failure (cable, connector, fan, etc), the archive unit can be fixed and reconnected to simply rejoin the system.
- If it is a major component fault, where the affected archive unit is no longer usable and the system is replicated to a remote site, a new archive unit can be seeded at the remote site and sent back to the data center to join the system.
- If the affected archive unit is no longer available and no secondary DR system is present, the filesystem can be pruned and the system can continue to operate beyond the fault.

In addition, fault isolation goes even further. If this catastrophe is larger in scope and most of the system is affected and without a replica, then all units that are recovered can be added to a new DD Archiver system and all data that survived the disaster and was salvaged will be available in the new system.

## Granular Recovery from a Replica

DD Archiver, like many storage platforms, needs to offer a disaster recovery configuration to keep a full replica of all stored data in a separate system in a remote site that is protected from disasters and catastrophes. However, for very large traditional storage systems, a significant system fault can cause replication to get out of sync or stop completely. In this case, the last thing an administrator wants to do is send petabytes of data over the network from the remote site to the local data center. Sending this large amount of data over the WAN is physically impractical as it would take weeks, and shipping the data on trucks would include over a dozen devices, depending on the stored capacity and could become difficult and expensive to manage.

The DD Archiver leverages its unique architecture to provide more recovery granularity and instead of having a system-to-system replication relationship between the source system in the data center and the replica system in the remote site, it establishes a replication relationship between each active unit and archive unit in the source system with its corresponding unit in the replica system.

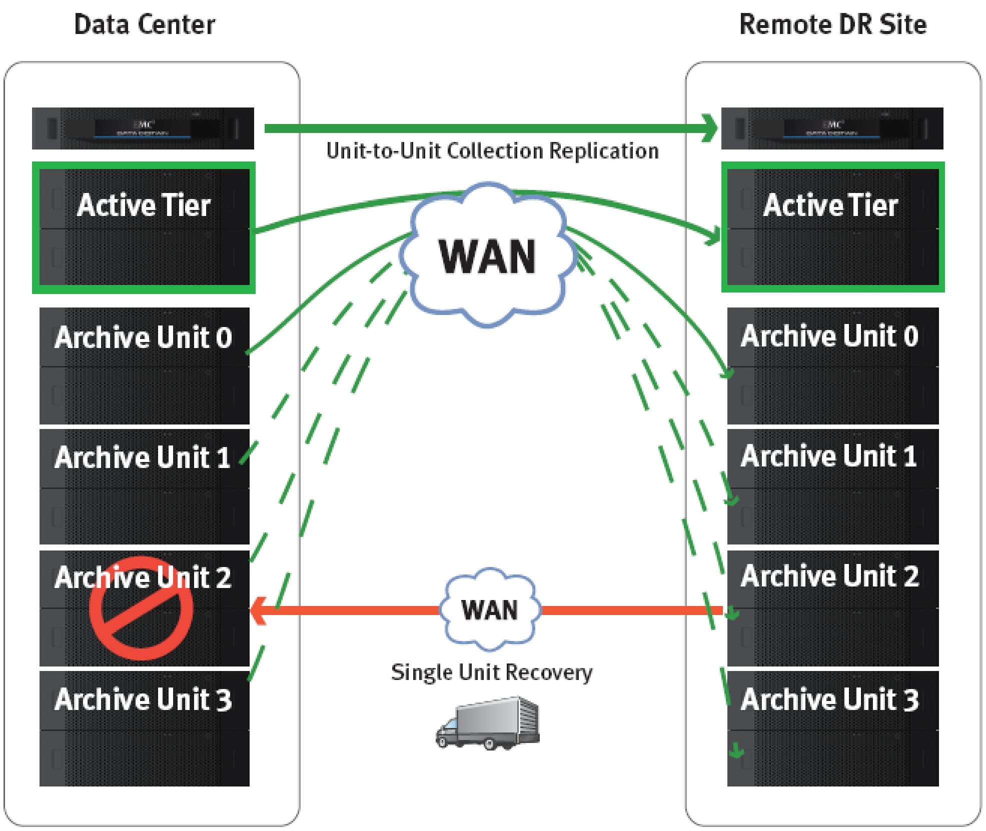


Figure 8: Two DD Archiver systems in their Disaster Recovery (DR) configuration, which is a more granular form of collection replication. The main benefit is illustrated by allowing a single-unit-recovery to replace archive unit 2 instead of recovering the whole system.

As Fig 8 above illustrates, if something affects the availability of a major component, for example unit 2 in the source system, only the affected unit needs to be re-synched or recovered (seeded in the remote site and shipped to the data center). Without this replication granularity, one would have to send all data from the remote site to the data center, which at scale would include over 20 components sent in trucks.

This more granular replication relationship in DD Archiver enables more specific recoverability if an isolated unit within a petabyte system is affected. This granular recovery can go faster and reduces risk to the rest of the system than in standard file server approaches.

## Cost-Effective Scalability

Another benefit of the DD Archiver is large scalability in the archive tier with a single mid-range controller. Currently DD Archiver supports four times the capacity of the standard Data Domain system with the same class of controller. With up to 142 TB of capacity in the active tier, the DD Archiver can scale up to a total of 570 TB of usable capacity. Logically, assuming backup compression ratios from 10x to 50x, the DD Archiver could scale up to 28.5 PB of long-term retention backup data. Amortized across so many storage shelves, the cost of the controller at scale becomes minimal. In a backup case, with assumed data reduction ratios above, the list price of DD Archiver can range from below $0.06/GB to $0.30/GB.

While tape cartridges in a cardboard box can be cheaper than this, the operating characteristics and risks of tape are also known to be suboptimal and can add unforeseen costs, as previously discussed.

### Future Architectural Extensions

In addition to all the benefits mentioned thus far, the DD Archiver architecture is very extensible and in future releases the system could incorporate additional features and functions that leverage the unique architecture of the current release. Specific plans may be discussed with EMC field representatives under an appropriate non-disclosure agreement. Some of these extensions could include:

- **Modular upgrades and migrations**. The DD Archiver is designed to be upgraded in a modular fashion over time. Data Domain high-end systems already support data-in-place upgrades, to enable upgrades to the latest controller technology without data migration. Additional extensions could also further simplify storage subsystem upgrades and consolidation over time.
- **Power management / spin-down**. Archive unit sealing and the separation of deduplication contexts into independent units could allow the archive units to be spun down to lower power consumption without affecting the system's ability to store more data.
- **More aggressive data reduction techniques**. Today, default compression in the active tier uses the standard LZ compression algorithm, but moving data to the archive tier the system recompresses data with the more aggressive GZ compression algorithm. In the future, the archive tier could be an area of focus for additional capabilities and features to more cost-optimized storage, such as denser shelves/drives or more aggressive global/local compression techniques.
- **Compliance regulations support**. Currently, DD Archiver meets internal governance standards with EMC Data Domain Retention Lock software. However, in the future DD Archiver could support stringent compliance regulations including SEC 17a-4.

## Typical Deployment Scenarios and Best Practices

Since the Data Domain Archiver is part of the Data Domain product family and leverages a common operating system, it can be deployed as a stand-alone system, or can be used as a consolidated target for other Data Domain systems in larger customer environments. DD Archiver can be used in many situations and customer use cases, but the main use case is long-term retention of backups.

## Long-Term Retention of Backup Data

For users seeking a single system for short and long-term retention of backups, the DD Archiver can be deployed as a stand-alone deduplication storage system. This use case, as illustrated in figure 9 below, leverages a standard backup application like EMC NetWorker or Symantec NetBackup to send backups directly to the DD Archiver.

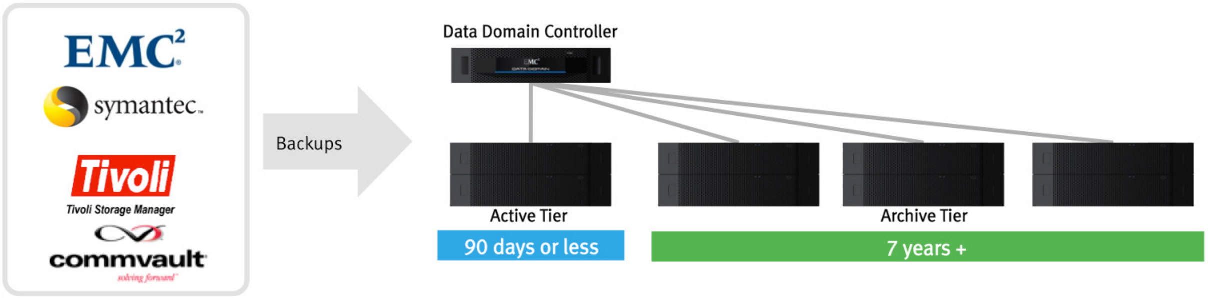


Figure 9: illustrates a configuration where backups are sent directly to the Data Domain Archiver. The active tier is sized to store the data of the short-term backup cycle, while the archive tier grows with long-term retention data over time.

By leveraging multiple shelves in the active tier, the DD Archiver can store both the short-term backup data for operational recovery and also keep long-term backup data for retention purposes. In this configuration, the active tier would be sized like a standard Data Domain system and the archive tier would be sized based on the required retention policy (length of retention period).

DD Archiver can also be used to consolidate long-term backups across a distributed enterprise. As shown in figure 10 below, other Data Domain systems are replicating into one centralized Data Domain Archiver. In this configuration, backup application at each remote site would be sending backup data to a local Data Domain system and then would leverage DD Replicator and DD Boost to send only unique data over the WAN to the DD Archiver for long-term retention.

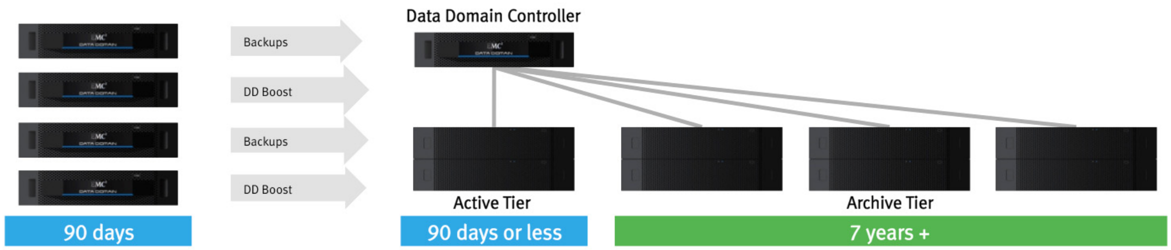


Figure 10: illustrates a configuration where only long-term backups are replicated to the Data Domain Archiver from other Data Domain systems. The active tier is sized to support the incoming data, while the archive tier grows with long-term retention of aggregated data over time.

Unlike the first use case in figure 9, since the active tier is not being used for operational recovery in this deployment, it may only require a shelf or two depending on how much data is being sent. In addition, since DD Archiver supports DD Boost, replicating between systems and implementing different retention periods on the replica is simple because all management can be done from the backup application. This ease of replication management is only available through DD Boost with EMC Avamar, EMC NetWorker, Symantec NetBackup or Backup Exec.

## Archiving and Tiering Data to the DD Archiver

Besides protecting standard backup data, the DD Archiver can simultaneously store archiving and file tiering data for long-term retention. Just like regular Data Domain system, the DD Archiver is a multi-purpose device that can be used for backup and archive use cases. Specifically, DD Archiver is an excellent target for virtualized file tiering (using an EMC File Management Appliance or an F5 ARX file virtualization device), archiving project data (archived manually or via scripts) as well as email, file and Microsoft SharePoint archiving that is performed by archiving applications (including EMC SourceOne, Symantec Enterprise Vault and others). In each case, older data is moved from primary storage into the DD Archiver (more cost-effective) and the data that remains in primary storage can still be backed up into the same DD Archiver system.

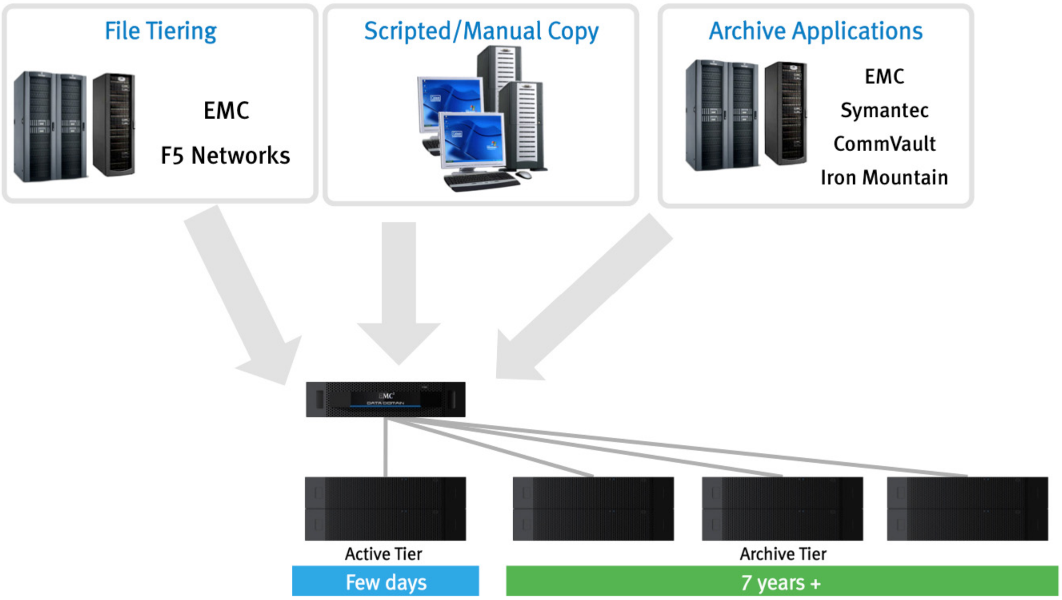


Figure 11: In addition to storing long-term retention backup data, the DD Archiver can also incorporate archiving workloads for cost-effective retention.

Unlike the backup only use cases above, all data in this DD Archiver deployment is already determined to be archive data before it lands on the system. Therefore although a formal sizing analysis is required, the active tier can likely be sized with fewer shelves and the data movement policy can be very aggressive.

## Comparing DD Archiver to Tape for Long-Term Retention

DD Archiver provides an impressive scalability behind a single controller. This scability combined with deduplication efficiencies enables the average system cost to be between 6 cents and 30 cents per gigabyte at list price. This average cost may not compare favorably to the tape media costs that range between 7 and 13 cents per gigabyte. However, the bulk of the tape costs include tape library costs, tape library management software costs, tape infrastructure management personnel costs, tape shipping and offsite storage costs, as well as media migration costs required every 5 to 7 years to upgrade to newer tape drive technology. When examining the total cost of ownership of operating a tape infrastructure, tape media costs are a small fraction of the equation. Therefore, 6 to 30 cents per gigabyte for DD Archiver is cost-effective.

When one combines the average system cost at scale of DD Archiver with tremendous improvements in RPO and RTO, time to DR, online data access, data integrity, fault isolation and granular recoverability, and improved manageability, DD Archiver is game-changing in the world of long-term backup and archive retention on deduplication storage.

## Conclusion

The DD Archiver was designed as a storage platform that supports the long-term retention practices of today and tomorrow. Customers can cost-effectively store and keep their backups on disk for longer periods of time as they implement traditional archiving solutions over time. DD Archiver is a long-term retention system that supports both specialty archive and existing backup workloads- it supports the transition from long-term backup retention as a de facto archiving method to specialized archiving processes over time.

The DD Archiver incorporates fault-isolation to make archive units self-contained units of data preservation and for long-term granular recoverability. The system's cost-effective scalability combined with this additional resilience, modular upgrades and migrations, simple management and extensible architecture enable the long-term retention requirements that such a backup/archive bridge platform must meet.

In addition, support for backup workloads requires throughput performance and the right deduplication approach to ensure backups complete in time and are stored in a cost-effective manner. Using a standard Data Domain controller enables DD Archiver to provide optimized throughput and variable-length segment deduplication for backup data stored on the active tier.

Since DD Archiver leverages industry leading Data Domain technology, the system comes with Data Domain proven resilience, including Data Domain Data Invulnerability Architecture, DD Replicator software option for network-efficient replication and Data Domain Retention Lock software option for hardware enforced retention of files.

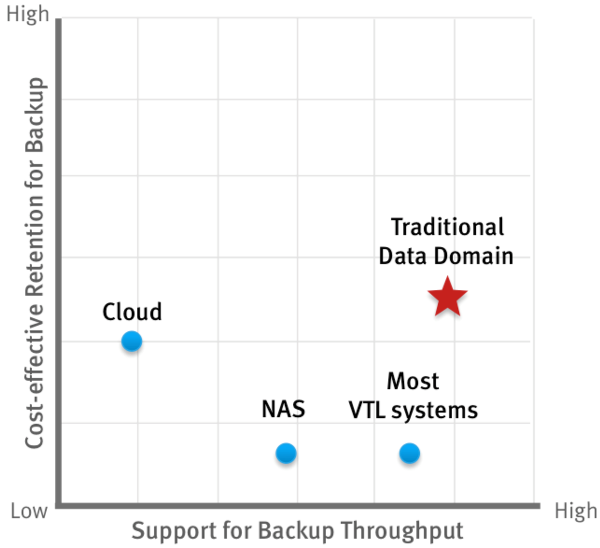


Figure 12: Graph showing several platforms as they rank according to requirements for long-term retention of backup and archive data. High backup throughput required to quickly ingest and cost-effective store backups, while a retention model is required to ensure cost-effective long-term retention.

Since 2003, Data Domain systems have been an ideal solution for eliminating tape for operational recovery, and now the DD Archiver can take it to the next level to further reduce tape, including long-term retention of backup and archive data. As shown in figure 12, with its high backup throughput, scalable long-term retention capabilities, cost optimized design and extensible architecture, the DD Archiver is the ultimate data preservation platform, moving to tape's major hideout in the data protection market landscape.