White Paper

Achieving Storage Efficiency with EMC Celerra

Best Practices Planning

Abstract

This white paper outlines strategies for reducing costs associated with storage capacity, power consumption, and storage administration. Specific features of the EMC® Celerra® storage system that increase storage efficiency are discussed, including deduplication, Virtual Provisioning™ and storage tiering.

May 2009

Part Number H6265.1

Table of Contents

Executive summary

Storage efficiency encompasses numerous strategies to reduce the costs of storing and managing data. These techniques include data reduction techniques like deduplication, tiering strategies that match storage characteristics to data value and access requirements, and newly available disk technologies that increase the performance of frequently accessed data while decreasing storage costs and power consumption for older static data.

The EMC® Celerra® storage platform has a rich feature set that maximizes storage efficiency through storage pooling, deduplication, Virtual Provisioning™ and tiering. Automated management features like policy-based tiering and deduplication, automatic volume provisioning, and automatic extension of file systems and storage pools bring additional benefits to storing data on the Celerra platform by reducing the amount of administrator involvement required for typical storage management activities.

With the advent of low-power SATA drives and solid-state storage, storage administrators increasingly have more choices about how to provision storage. Different drive technologies present a range of cost, density, power consumption and performance characteristics to the storage administrator, who must decide where to best store a given data set.

The value of data changes over time, just as the access characteristics associated with individual files change within a data set. Celerra automated storage tiering helps the administrator automatically match the access requirements of individual files with the most appropriate drive technology. This minimizes storage costs by limiting the amount of high-performance storage to the amount of data that requires high performance access. It achieves this while also minimizing the power and cooling requirements of storing data that does not require high-performance access. Automated storage tiering takes advantage of policy-driven features like tiering and deduplication to transparently place data on the most appropriate tier of storage based on file access characteristics, without requiring administrator intervention.

The core technology that enables automated storage tiering is Celerra FileMover used in conjunction with the Rainfinity® File Management Appliance (FMA). FMA provides the policy engine that is used to select files to be relocated to different storage tiers, while FileMover provides the underlying file system capabilities that gives end users transparent access to relocated files.

Automated storage tiering can be combined with other Celerra features to further increase the inherent efficiency of tiered storage, such as deduplication to reduce the number of disk drives required to store secondary tier data and file-level retention (FLR) to enforce data retention policies on archived data.

Advanced storage features like automatic extension, Virtual Provisioning and storage pools further increase the efficiency of storing data on Celerra by minimizing the amount of overallocation required to allow for the future growth of a given data set.

Combining these techniques minimizes storage costs and administrator overhead. This results in more efficient use of storage resources.

Introduction

This paper describes Celerra features that maximize storage efficiency and explains how these features can be used in combination with the latest disk technology to reduce the costs of storing and manage an ever-increasing amount of data without impacting end users or incurring additional storage administration costs.

Audience

This paper is intended for storage administrators engaged in planning deployments of network attached storage.

Terminology

Archiving: The process that walks through the share/export and performs policy-based file archiving.

Celerra FileMover: Policy-based system used to determine where files should be physically stored. In most cases, policies are based on file size or last access time (LAT) or both and are used to identify data that can be moved to slower, less expensive storage.

Common Internet File System (CIFS): File-sharing protocol based on the Microsoft Server Message Block (SMB). It allows users to share file systems over the Internet and intranets.

Extended ACL: Container used to store extended attributes. In a Celerra FileMover system, extended ACLs are used to back up and restore stub files through a CIFS network backup.

File version: Multiple copies on secondary storage of the same file or path.

Network Data Management Protocol (NDMP): Open standard network protocol designed for enterprise-wide backup and recovery of heterogeneous network-attached storage.

Network File System (NFS): Distributed file system providing transparent access to remote file systems. NFS allows all network systems to share a single copy of a directory.

Orphan file: Files on the secondary storage with no reference to the primary storage because it is no longer referenced by a stub file on the Celerra Network Server. An orphan file is kept intact on secondary storage; it can be managed by the system administrator.

Passthrough: Archiving method option that instructs the Celerra Network Server to read the requested data from secondary storage. The data is returned to the client, but is not recalled to primary storage; thus the stub file remains intact.

Policy and archiving application: Software through which archiving policies are defined, implemented, and managed in order to archive data from the Celerra Network Server to secondary storage according to customer-defined policies and schedules.

Policy engine: For tiering, the external server on which the policy and archiving application runs. For deduplication, the policy and deduplication application runs inside the Celerra Network Server.

Primary storage: Data storage on the Celerra Network Server that provides clients access to normal files as well as archived files through the stub files that represent them.

Recall policy: Attribute that is set per individual file and is in the stub file when a file is archived to secondary storage. The Celerra Network Server uses the recall policy to determine the amount of data that is recalled to primary storage to satisfy the client read request.

Secondary storage: Data storage that is an archiving destination for primary storage and contains original files associated with stub files.

SnapSure™: On a Celerra system, a feature providing read-only point-in-time copies, also known as checkpoints, of a file system.

Stub files/offline files: Files that appear as normal files on the primary storage but point to data content that is stored on the secondary storage.

Storage efficiency overview

The goal of storage efficiency is to reduce costs associated with storage capacity, power consumption, and storage administration. The Celerra storage system provides many features that can be used to meet this goal. Features include support for multiple disk drive types (with varying prices), performance and power characteristics, automated management tools that provide storage provisioning, monitoring and file system extension. These features also include policy-based file system tiering that automatically distributes files to appropriate drive types and policy-based deduplication that minimizes the amount of space consumed be duplicate and static data.

Techniques for achieving storage efficiency with Celerra

The biggest gains in storage efficiency are best realized during the planning phase of storage deployment. For example, it is easier to initially deploy a data set on the proper disk drive type than to migrate the data to a different tier later. Likewise, on traditional storage systems, it is better to provision space for future growth than to run out of space and have to migrate data to a new storage system or larger file system. The Celerra system includes features that facilitate good planning and minimize the need for reactive administrative actions and temporary data access outages. This section describes a proactive approach to improving storage efficiency by utilizing Celerra efficiency-related features.

Match data characteristics to drive characteristics

Celerra supports several different disk drive technologies that have a range of price, capacity, performance and power consumption characteristics. Supported drive types include solid-state enterprise Flash drives (EFD), Fibre Channel drives and Serial-attached ATA drives (SATA). Each drive type is available in multiple capacities and, in the case of FC and SATA, different rotational speeds. As a general rule, larger capacity drives have a better cost per unit of storage ($/GB), while higher rotational speeds usually result in incremental performance improvements at the expense of greater power consumption.

Enterprise Flash Drives

For performance-sensitive applications, it can be useful to view drive costs in terms of the cost of a given performance target instead of cost of capacity ($/IOPS versus $/GB). The high IOPS performance of Enterprise Flash Drive technology makes it a cost-effective replacement for data sets that have been traditionally spread across a large number of spindles to meet performance goals at the expense of wasted capacity. For certain I/O profiles, a single five-disk EFD RAID group can match the performance of approximately 24 five-disk FC RAID groups, eliminating the power and cooling requirements associated with over 100 spinning drives.

One of the simplest and most effective techniques for taking advantage of differences in drive characteristics is to match the performance requirements of a data set to the performance characteristics of a specific drive type when allocating storage space for a file system.

Automated tiering

In practice, not all files within a data set share the same performance requirements. For example, older files are less likely to be accessed or modified than newer files. Files that are rarely accessed have lower performance requirements than files that are frequently accessed.

The Celerra automated storage tiering architecture allows an administrator to define policies that match file characteristics to drive characteristics. Once defined, these policies automatically move files across from one drive type to another as file drive types as file access characteristics change over time, while providing continuous transparent user access to the relocated files. A typical use case for this architecture is to provide a limited amount of high-performance storage (EFD or FC) to store a data set’s active data, while providing a larger amount of capacity-optimized storage (SATA) for the data set’s static data. The

Rainfinity FMA enables this architecture by providing the policy engine and file movement component of the architecture as shown in Figure 1.

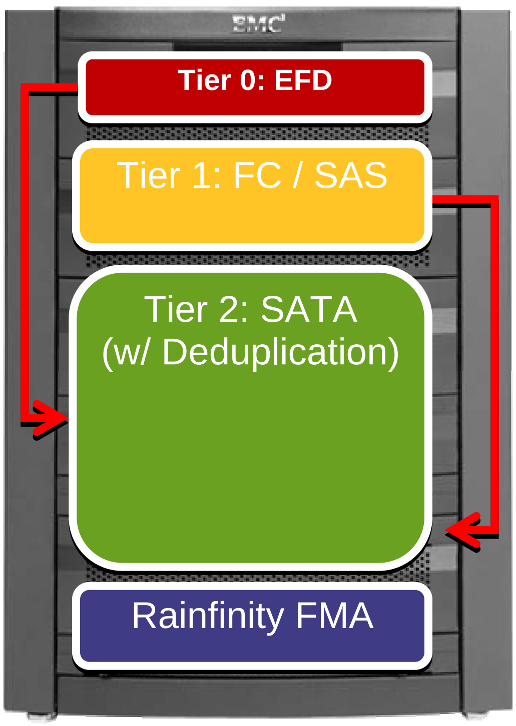

Figure 1 **Automated storage tiering architecture with Rainfinity FMA**

Minimize wasted capacity

Traditional provisioning strategies involve assessing current storage requirements and growth rates to estimate the future size of a data set. Allocating space for future growth helps avoid running out of space in the future, but it comes with a cost: storage pre-allocated for future growth in one file system is unavailable for other uses. If the expected growth does not materialize, the storage cannot be re-allocated to other file systems unless the data is migrated to a new location and the original file system is deleted. When this condition is repeated across a number of data sets, the amount of space that is unusable due to pre-allocation for future needs can represent a significant percentage of the available storage on a system.

Storage pools

A more efficient technique to provide space for future growth is to use a single pool of storage as a source of additional capacity for a group of file systems. With a shared pool of extra capacity, it is less likely that unneeded storage will end up unnecessarily allocated to a data set that does not grow as expected. File systems can be allocated on an as needed basis and extended as they require more space over time. File systems that do not require additional space remain at their originally allocated size. This leaves more space in the storage pool that is available to file systems that do require additional space.

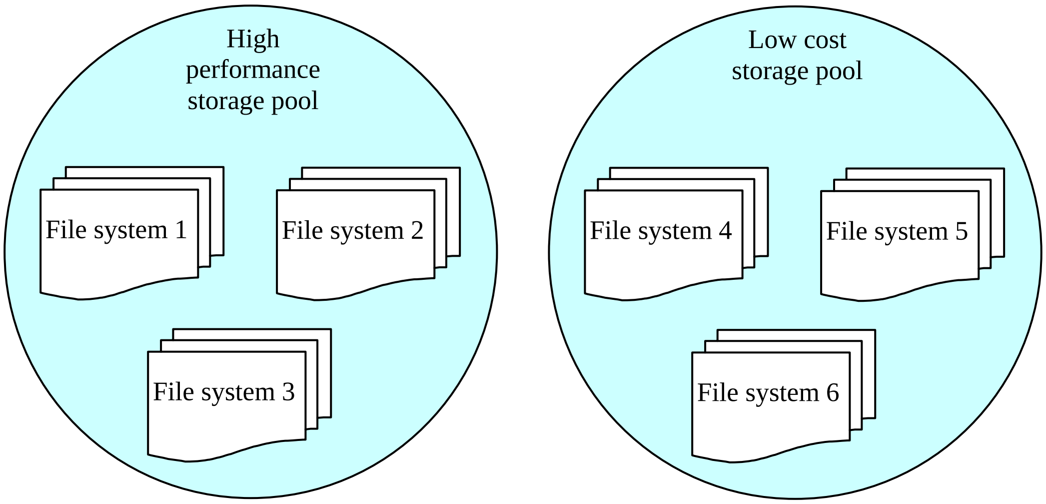

Figure 2 Group data with similar performance requirements into storage pools

Automatic extension

The storage pool concept relies on the ability to grow or extend file systems, as their sizes approach the storage capacity allocated to them. Likewise, as a storage pool fills, it is beneficial to be able to extend the storage pool by allocating additional raw disk space to the pool so it can be used for future growth needs. Celerra allows both file systems and storage pools to automatically grow as needed based on administrator-defined high water marks. This feature is called auto-extend and is configurable for both file systems and storage pools from the Celerra Manager GUI.

For the automated storage tiering architecture, especially in cases where there is a limited amount of primary storage available, it is preferable to respond to capacity growth by archiving older data to a secondary tier instead of growing the size of the primary file system. Therefore auto-extension should be disabled on the primary file system in cases where there is a desire to conserve primary tier storage. On the secondary tier, auto-extension should be enabled to avoid running out of space in the secondary file system as more primary tier data is offloaded to the secondary tier. Likewise, secondary tier storage pools should have auto-extend enabled to allow the pool to grow as needed.

Virtual Provisioning

It is possible to define an upper limit to the amount of space that can be allocated to a file system by the auto-extend feature. This maximum size represents the potential size of the file system as differentiated from the current allocated size. Virtual Provisioning is a Celerra feature that causes clients to see the available space in the file system in terms of the potential maximum size instead of the current allocated size. For example, a file system that needs to initially store 7 GB of data may be initially provisioned with a size of 10 GB and given the ability to auto-extend up to a maximum size of 30 GB. If Virtual Provisioning is enabled, clients will see the size of the file system as 30 GB instead of 10 GB. Virtual Provisioning thus allows a storage administrator to allocate storage on an as-needed basis, delaying allocations for potential future growth while communicating planned maximum capacities to clients.

Deduplication

Deduplication saves storage space by eliminating multiple copies of duplicate files. In addition, deduplication compresses all files that meet the criteria defined by the Celerra deduplication policy engine regardless of whether or not there are multiple copies of the files. If there are multiple copies of a file, only one compressed copy of the file is stored in the file system. An internal policy engine is used to define the criteria that files must meet in order to be single-instanced and compressed. One purpose of the policy engine is to exclude active files from deduplication processing. By default, the policy excludes files that have been accessed in the last 30 days or modified in the last 60 days. In general, deduplication reduces the amount of space required in a file system by 30-40 percent, but the exact savings rate depends on the distribution of sizes, file types, and level of activity associated with the files within the file system.

Deduplication is especially useful on the secondary tier for file systems utilizing the automated storage tiering architecture. The reason is that files on the secondary tier are by definition inactive (for example, any modification of a file on the secondary tier will automatically recall the file to the primary tier instead of modifying the file on the secondary tier). As a result, a much more aggressive deduplication policy can be configured on the secondary tier without any risk of deduplicating files that are soon to be modified. For example, it is beneficial to set the deduplication policy on the secondary tier to deduplicate files without regard to when they were created or last modified, so that deduplication will occur as soon as the deduplication engine detects the files' presence on the secondary file system. This is accomplished by setting the deduplication policy engine parameters *accessTime* and *modificationTime* to 0 using the *server_param –f dedupe* Celerra CLI command.

An additional benefit to deduplicating on the secondary tier is that it allows deduplication to work across multiple file systems. With the automated storage tiering architecture, multiple primary file systems can share a single secondary file system. When this shared secondary file system is deduplicated, duplicate files from multiple file systems are replaced on disk with a single compressed copy. This increases the space saving effects of deduplication.

Minimize power consumption

Reducing the power consumption of storage systems saves money in two ways: it lowers the cost of the power consumed by the storage system directly and it also lowers the costs associated with cooling the storage system. Using low power drives and avoiding allocation of unnecessary capacity are two techniques that can reduce power consumption.

Use low power disk drives

EMC offers a wide range of disk drive types, capacities and rotational speeds. The choice of drives can have a profound effect on power consumption as shown in Figure 3.

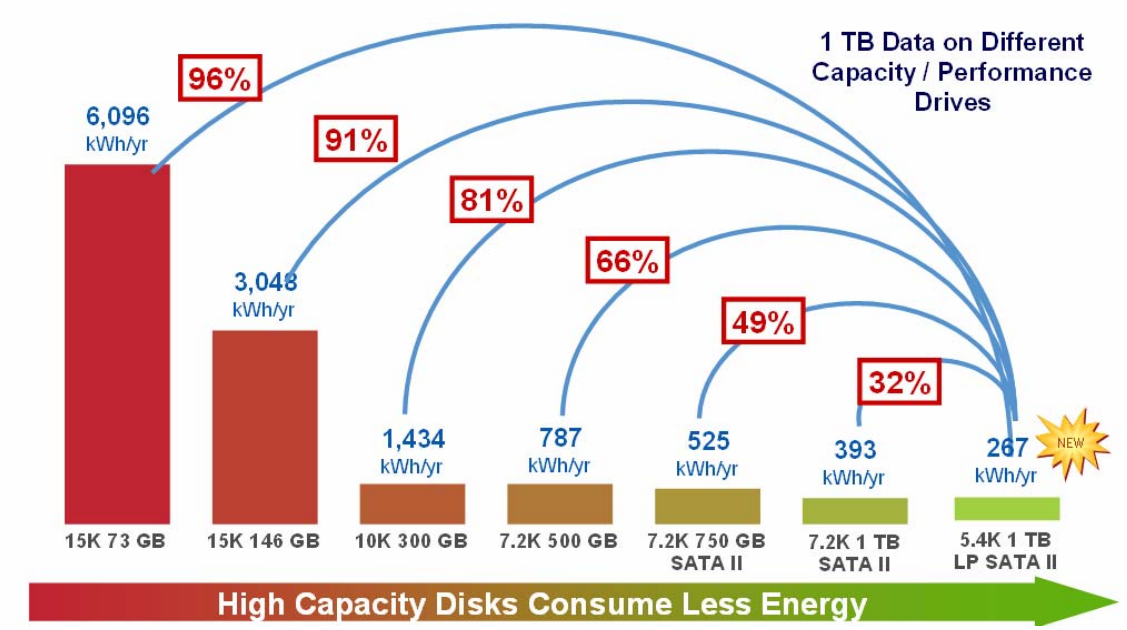

Figure 3 Power consumption of various drive types

Tiering techniques like automated storage tiering facilitate placing static data on lower power drives thereby limiting the amount of power required to store the data.

Minimize the number of drives powered on

Another power saving technique is to minimize the amount of capacity that is unused, yet powered on. The power consumed by drives allocated to a file system is a function of the number of drives allocated, not the amount of space used on them. The performance enhancement technique of spreading a file system across a large number of short-stroked spindles wastes disk space and power. If high spindle counts are being allocated for the purpose of providing high IOPS rates, power can be saved by relocating data with high performance requirements to EFDs. Unused space that is allocated to file systems for future growth also requires more drives being powered on than for current capacity needs. Use storage pools to share excess capacity for future growth amongst multiple file systems as a way to minimize the total amount of unused capacity on the storage system. Use auto-extension/Virtual Provisioning to avoid overallocating space when creating new file systems and provisioning new storage pools so that space is only allocated on an as-needed basis. The predictive monitoring feature in Celerra Manager tells when a storage pool will require additional capacity. Administrators can avoid adding shelves of storage that are reserved for future growth until the additional capacity is likely to be needed by the storage system. These techniques can reduce the total amount of powered-on storage capacity, reduce drive power consumption and decrease system cooling requirements.

Minimize administrative overhead

The Celerra storage system and automated storage tiering architecture include several features that allow an administrator to automate common administrative tasks. These include policy-driven functions like deduplication, storage tiering, and file system and storage pool extension. In addition, an administrator can register to receive automatic notifications when critical storage provisioning events occur, such as when a storage pool is over-provisioned, or when a file system or storage pool reaches a high water mark based on currently allocated or potentially available capacity. Use of these features can reduce the amount of hands-on time that an administrator must spend monitoring and administering the storage system.

Policy-driven tiering

Rainfinity FMA allows the administrator to determine specific file selection criteria as part of a policy used by the system to automatically choose which files in a primary file system should be moved to a lower tier of storage. These policies consist of rules based on file attributes like date last accessed, date last modified, size, and filename. The attributes are combined with user-selected operators and attribute values that allow the administrator to precisely define file types, sizes, or ages that will be archived to lower storage tiers. The policies may be previewed to test the effect the policy would have if run against a target file system. After the administrator has defined the policy, it can be scheduled to be run periodically based on a time schedule (like every day, once a week on a specific day, on a specific date of the month, and so on) or based on the file system reaching a user-defined capacity usage threshold called the Archive threshold.

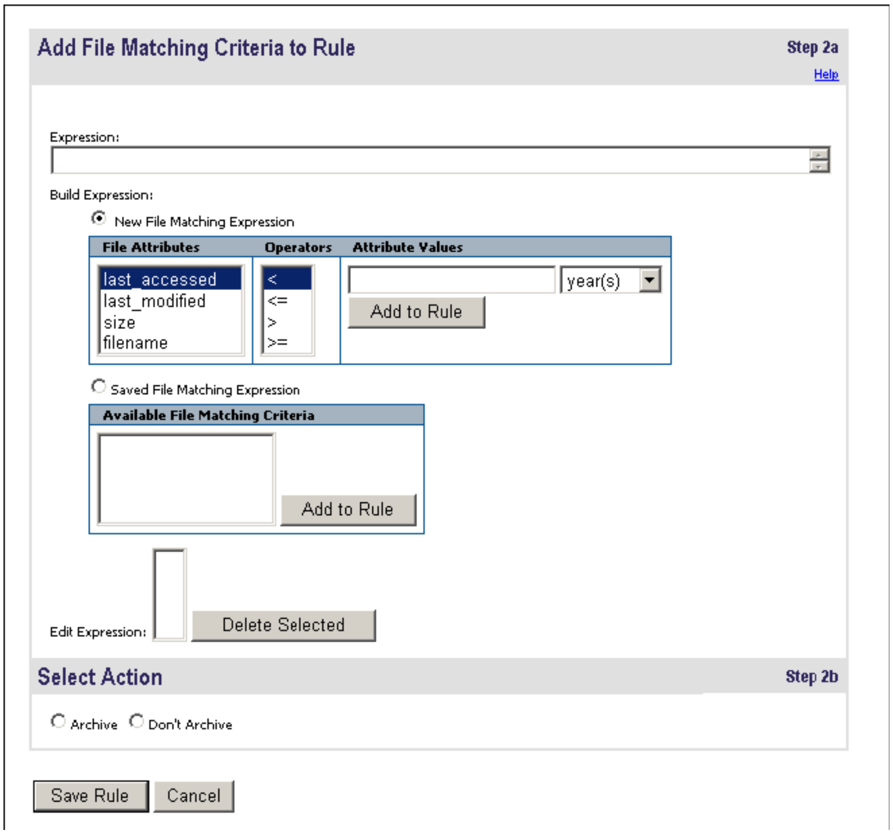

Figure 4 **Defining the file selection policy for archiving with FMA**

Policy-driven deduplication

Celerra deduplication is also policy-driven. It is enabled by a single checkbox in the Celerra Manager GUI as part of the file system creation. Deduplication can also be enabled or disabled at any time on existing file systems. Celerra includes a default policy that is sufficient for most file systems. Like the policies defined for tiering with FMA, the Celerra deduplication policy includes parameters for file types, sizes, and number of days since the file was last accessed or last modified. Unlike tiering with FMA, the deduplication policy engine is part of the Celerra system and does not require any external components. Figure 5 shows the default deduplication policy.

```
[nasadmin@europa ~]$ server_param server_2 -f dedupe -l
server_2 :
param_name                 facility       default    current    configured
fileExtensionExcludeList   dedupe           ''         ''
singleInstancingEnabled    dedupe            1          1
minimumSize                dedupe           24         24
savVolThreshold            dedupe           90         90
accessTime                 dedupe           30         30
maximumSize                dedupe          200        200
caseSensitive              dedupe            0          0
throttle.cpuLowTrigger     dedupe           40         40
dirExcludeList             dedupe           ''         ''
throttle.cpuHighTrigger    dedupe           75         75
minimumScanInterval        dedupe            7          7
modificationTime           dedupe           60         60
```

Figure 5 Default Celerra deduplication policy

By default, the deduplication policy is run once a week. During the weekly scan, it checks to see whether there are any new files that meet the policy criteria. The age-related parameters (*accessTime* and *modificationTime*) compare a file's access and modification times to the time of the scan to assess if a file has been static long enough to deduplicate. Over time, all files that are not periodically accessed will be deduplicated and compressed, freeing up space in the file system without any administrator involvement.

Automatic file system extension, Virtual Provisioning, and predictive monitoring

Another way that the Celerra system automates day-to-day administrative tasks is by monitoring file system growth and taking predefined actions when file system capacity usage reaches administrator-defined high water marks. This is accomplished with the Auto Extend Enabled checkbox in Celerra Manager. The administrator can define a maximum size to which the file system will be automatically extended as well as a usage threshold (called a high water mark) that is used to initiate automatic file system extension. A checkbox to enable Virtual Provisioning indicates whether or not users of the file system see its total capacity as the current allocated capacity, or the maximum size that the file system can grow to using automatic extension. The predictive monitoring feature analyzes file system growth rates and indicates the number of days it would take at the current growth rate for the file system to consume its allocated capacity. Analysis of the file systems storage pool is included along with a prediction of when the storage pool will reach its maximum capacity. Figure 6 on page 13 shows a portion of the Celerra Manager GUI page where these features are enabled.

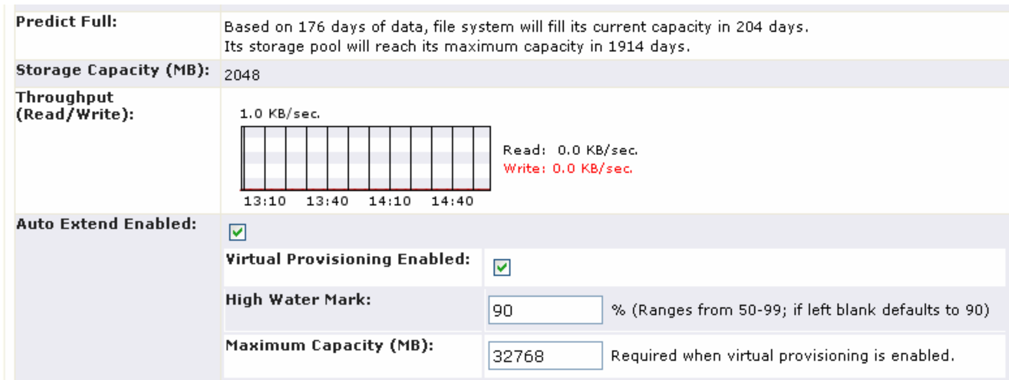

Figure 6 Automatic extension and Virtual Provisioning settings

Together, these automated features reduce the administrative overhead of managing file system growth on the Celerra system.

Data protection in the efficient storage environment

Any data storage methodology requires a strategy to protect data from component failures, operator errors and site disasters. Data stored using the efficient storage techniques described in this paper can be protected using traditional data protection strategies, such as backups, snapshots and replication.

System high availability

The first step in protecting data is to make sure that data will remain available in the event of a storage system component failure. Every tier of storage on a Celerra system is protected against disk failures through the use of RAID groups. Furthermore the CLARiiON® and Symmetrix® storage arrays utilized by the Celerra system include multiple redundant components to protect against failures of other base storage array components. The Celerra itself is protected by built-in n+1 clustering where one or more of the Celerra data serving components, called Data Movers or X-blades, can be protected by a dedicated local standby Data Mover that has all of the processing power and network connections of the active Data Movers. This ensures seamless failover with no change in I/O performance or network connectivity in the unlikely event that a failure occurs. These built-in hardware and software high-availability features protect all data stored on the system, regardless of how it is arranged across storage tiers and/or deduplicated and compressed.

Backups

To back up data sets that span multiple storage tiers present special requirements for backups use one of the following general approaches:

- Back up the data set as a single file system
- Back up the data set as separate file systems (one backup for each storage tier)

Backing up the data as separate file systems require that restores of multi-tier data sets must not break the links between the stub files on the primary tier and the archived file content on the secondary tier. The Celerra FileMover feature that enables tiering requires that the archived version of a file stored on the secondary tier must exactly match the version recorded in the primary tiers stub file that represents the archived file.

For example, there may be adverse effects if a restored backup on one tier does not match the current state or restored backup state on the other tier. For this reason, it is easier to back up a tiered data set as a single file system, accessed through the primary tier.

Backing up from primary only

When backing up from primary only, the contents of both the primary tier file system and the archived files stored in the secondary tier file system are included in the backup of the primary tier file system. This occurs through the use of automatic passthrough read recalls that read the archived files from the secondary tier during backup processing, without relocating the files onto the primary tier file system. The technique to enable passthrough reads during backup processing is dependent on the backup method, as described below.

NDMP

For NDMP backups, set the NDMP environment variable in the backup job definition to tell the software to back up file contents of archived files on secondary storage tiers:

```
EMC_OFFLINE=y
```

Note that the default setting is EMC_OFFLINE=n, which means that only the primary storage tier is backed up.

CIFS

By default, CIFS backups of the primary tier file system will include archived file data from the secondary tier. CIFS backups of the primary tier file system read file content from the secondary storage tier in passthrough mode, which does not relocate the data to primary tier storage.

NFS

NFS backups of the primary file system always include associated secondary tier file content. Set the Celerra FileMover – *read_policy_override* option to passthrough to ensure that NFS backup operations do not relocate secondary tier file content back to primary storage tier. Execute the following command at the Celerra CLI prompt to set this option:

```
$ fs_dhsm –modify <fs_name> -read_policy_override passthrough
```

Note: Restoring a primary file system backup that contains both primary and secondary file system content (That is, backups created with the read passthrough option) writes content from both tiers into the primary tier file system. The content that was originally on the secondary tier can be moved back to the secondary tier by subsequent passes of the FMA policy engine scan. To avoid overrunning a limited capacity primary tier file system, it may be necessary to create a special FMA policy that more aggressively seeks files to archive, and run this policy continuously during the restore, while monitoring primary file system capacity usage.

Backing up primary and secondary separately

The advantage to separately backing up the primary and secondary tier components of a multi-tier data set is that it decreases the backup window required for backing up the primary tier data. This is because the primary tier backup contains less data than the primary and secondary tier. Also, the secondary file system is typically modified only by policy engine activities and can be backed up less frequently. A best practice is to back up the secondary file system after every FMA policy engine scan or orphan file cleanup operation.

The disadvantage of the separate backup strategy is that it requires special considerations during backup and restore operations to ensure that the stub files in a restored primary file system can locate their associated file contents on the secondary file system:

- The primary tier file system and its associated secondary tier file system must be backed up independently
- When restoring a backup to either the primary or secondary tier, follow these guidelines:
  - If the primary backup is more than 30 days old and orphan file cleanup procedures are regularly used, you must also restore the secondary tier from a backup taken more recently than the primary backup but within 30 days of the primary backup.
  - Restoring a backup of the secondary file system includes restoring a backed up copy of the primary file system that is older than the secondary version

- Single file or directory level restores from primary file system backups that are more than 30 days old may require administrative action to determine if it necessary to also recover older versions of archived files from secondary file system backups

Note: When the secondary file system is shared by several primary systems, it may be necessary to restore all primary systems to the same recovery point,. This is because all stub files on the primaries must have corresponding files on the secondary. As a result, the primaries may need to revert to an earlier data version.

Taking separate data backups on primary and secondary tiers requires the ability to do stub-aware backups of the primary file system. The technique to enable stub-aware backup processing is dependent on the backup method.

NDMP

For NDMP backups, set the backup software's NDMP environment variable in the backup job definition to avoid including secondary tier content in the primary file system backup:

```
EMC_OFFLINE=n
```

Note: This is the default setting.

CIFS

Set the Celerra FileMover *–backup* option to offline to back up stub files and files that have not been archived when backing up primary tier file systems. Execute the following command at the Celerra CLI prompt to set this option:

```
$ fs_dhsm –modify <fs_name> -backup offline
```

NFS

It is not possible to configure NFS backups of the primary file system to be stub-aware. All NFS backups of the primary tier file system include all associated secondary tier file content.

Snapshots

Celerra SnapSure creates point-in-time copies of a file system. These copies, called checkpoints or snapshots, use very little disk space and have a variety of uses. Snapshots are useful as the source for backups because they provide a static view of the file system and they can also be used as temporary online backups that allow users to self-restore deleted files or directories using Microsoft's VSS-based *Previous Versions* tab in Windows Explorer. Administrators can also use snapshots to restore entire file systems back to a specific point in time.

Tiered data set snapshot considerations

For tiered data sets, snapshots must be taken separately on the primary and secondary file systems. When restoring entire file systems from snapshots, follow the guidelines listed in the previous section for restoring separate primary and secondary backups.

Note: Single file and directory level restores from primary file system snapshots may result in stub files on the primary file system that do match corresponding file versions on the secondary file system. This situation is more likely to occur with snapshots that are older than 30 days and can be generally avoided by only keeping snapshots for shorter periods of time.

Snapshots can reduce the effectiveness of deduplication and tiering because space saved in a file system during either of these operations is added to the snapshots SAVVOL storage usage until the snapshot is deleted. For this reason, snapshots of primary tier file systems should be stored on the secondary tier if primary tier capacity conservation is a priority. However, if high IOPS write performance to files on primary storage is a higher priority than conserving space, then snapshots should be stored on the primary tier. To realize the full capacity saving benefits of the automated storage tiering architecture, snapshots should only be kept for short-term use. This is easily accomplished through the Celerra Manager checkpoint scheduling feature. It allows an administrator to configure the system to automatically take snapshots at specific times of day, on specific days of the week or month, and to define a specific number of snapshots to keep. For example, snapshots scheduled to be taken once a day can be configured to only keep seven copies, ensuring that there will never be a snapshot on the system that is more than a week old.

Replication for disaster recovery

Celerra Replicator™ provides asynchronous replication between Celerra systems for protection against catastrophic disasters. The Replicator feature is compatible with FileMover tiered data sets and Celerra deduplication.

Replication of tiered data sets

Although it is possible to replicate only the primary file system in a tiered storage architecture and have the stubs on the replicated copy of the primary file system point back to the non-replicated secondary file system, this is not recommended for the automated storage tiering architecture. Since the automated storage tiering architecture stores all of tiered data set's content on a single Celerra system, better protection is realized by replicating both primary and secondary tier file systems in separate replication sessions so that both primary and secondary tier content is fully protected against site disasters that affect the source Celerra. Only a single FMA device is required (at the replication source site) for basic disaster protection. However, if there is an extended site outage, it may be helpful to have a second FMA device (at the replication destination site) to allow for continued FMA policy engine scans during the outage.

Replication of deduplicated file systems

Deduplicating the contents of a file system before it is replicated can greatly reduce the amount of data that has to be sent over the network as part of the initial baseline copy process. Once replication and deduplication are running together, most new or modified files will be replicated before they will be deduplicated. For that reason, deduplication will have little effect on the amount of data replicated. However, deduplication will save the same amount of space in the replicated file system at the source and destination sites reducing the overall disk usage at both sites.

Anti-virus scans and content search considerations

It is important to avoid inadvertently recalling archived data from a secondary tier back to the primary tier file system. This is particularly true if the space available on the primary tier is smaller than the total size of the tiered data set. Care should be taken to ensure that applications do not cause archived files to be recalled unexpectedly. Applications like anti-virus scans, Windows Explorer file content searches and Windows offline folder synchronization can all cause inadvertent recalls of archived files that can be prevented by setting the FileMover *-read_policy_override* option to passthrough.

Anti-virus scans

By default, the Celerra AntiVirus Agent (CAVA) does not recall data during scan-on-read operations; archived data is read using the passthrough mechanism. Using CAVA to scan all files on a primary file system will skip stub files and only scan files whose contents are stored on the primary tier. Use the *–fsscan –create offline* option to change this behavior so that scans of the primary tier will also scan archived content on the secondary tier using the passthrough mechanism. Although EMC recommends the use of CAVA for antivirus protection of tiered data sets, it is also possible to use network based anti-virus scanners. In this case, it is possible that the anti-virus scanner will recall files back to the primary tier file system unless the FileMover read policy for the file system is explicitly set to passthough by using the *read_policy_override* option to the dhsm Celerra CLI command. Therefore the passthrough read policy should also be used on file systems that will be scanned by network based anti-virus scanners.

Content searches

When you perform a content search on the primary file system of a tiered data set, the Celerra system reads the content of archived files from secondary tiers as specified in the FileMover read recall policy. To avoid having data mistakenly relocated to the primary tier during a content search, set the FileMover archiving method to passthrough. This will cause reads of archived data to return the data to the client without recalling archived files to the primary tier file system.

Storage efficiency case study

XYZ Corporation is currently storing 5 TB of data in 10 file systems spread across 10 Windows and Linux file servers. Each file server hosts a single file system that contains 500 GB of data. Each file system was designed to allow for 100 percent future growth by creating each file system on an internal SATA disk with 1 TB of capacity. This architecture can be seen in Figure 7.

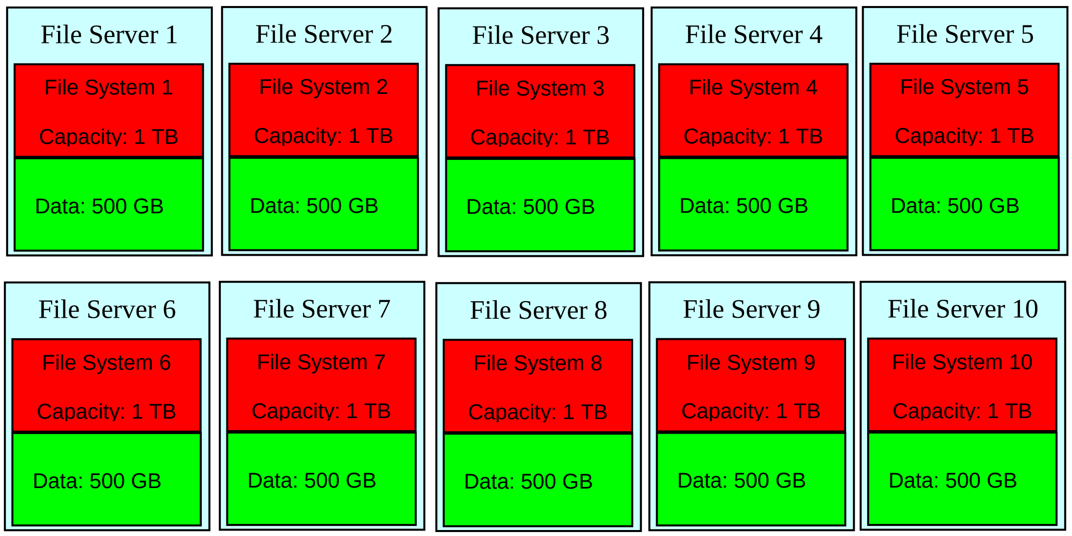

Figure 7 XYZ Corporation storage architecture

Storage architecture problems at XYZ Corporation

Several problems are present with their current storage architecture.

Large amounts of wasted capacity

The company is currently wasting 50 percent of their disk capacity (5 TB of data is stored on 10 TB of disk). This translates into a proportional amount of wasted power as well as associated cooling capacity in the data center. Likewise disk capacity used to store multiple copies of identical files is also wasted.

Space provided for future growth cannot be redistributed

The disk space allocated for future growth is equally distributed across all of the file systems and cannot be redistributed if future growth rates of each data set do not match initial projections. Differences in the actual growth rates of different file systems cannot be accommodated. Since it is difficult to precisely project growth rates, this architecture requires XYZ Corporation to allocate space for future growth based on worst case projections. While some of the data sets will indeed experience high growth rates, it is also possible that some data sets will grow at a considerably slower rate; unfortunately, the excess space allocated to the slower growing data sets cannot be reclaimed for use by faster growing data sets.

Monolithic performance and storage costs

Only a single disk type is available; there is no way to improve the storage performance of active data or decrease the cost or power consumption associated with storing static data.

Distributed and manual administration
Each file server must be administered and monitored separately. Multiple administrator skill sets are needed due to the use of multiple file server architectures. Each system will have to be manually monitored to assess whether its hosted file system is approaching the limits of its allocated size. If a file system fills the 1 TB allocated to it, an administrator may need to migrate the file system to a new machine with more disk or install additional or larger disks into the original file server and manually recreate or expand the original file system (if the host operating system provides this capability). These operations may require temporary periods of data unavailability.

Recommended efficiency improvements

Using Celerra features and the automated storage tiering architecture, XYZ Corporation can make many improvements over their existing storage architecture. The new architecture is shown in Figure 8.

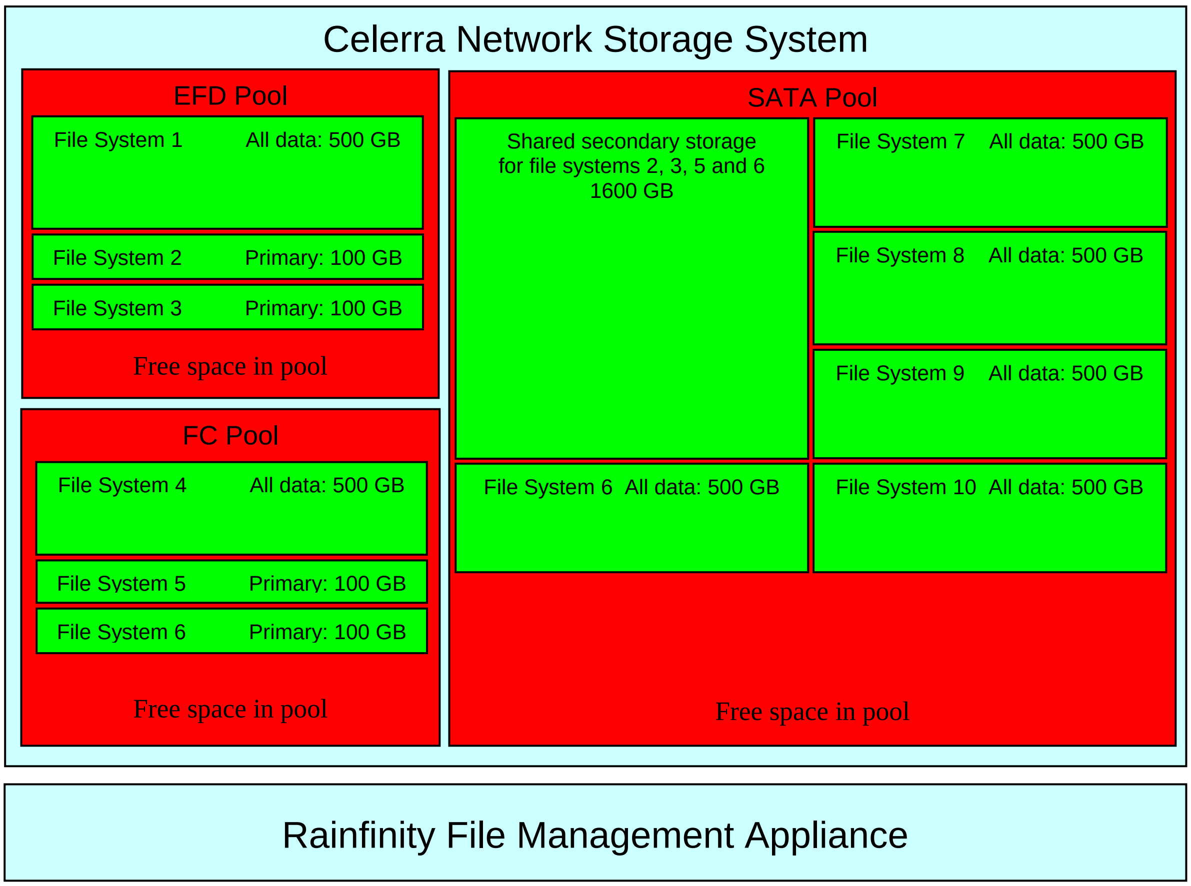

Figure 8 XYZ Corporation architecture with efficiency improvements

Minimize wasted capacity by sharing excess capacity
Creating file systems out of shared storage pools would allow XYZ Corporation to minimize the total capacity set aside for future growth, while still allowing for concentrated high growth rates amongst a subset of their file systems. For example, instead of setting aside 50 percent of their total capacity to allow for future growth, they could reduce this to 25 percent because differences in growth rates of individual file systems will be distributed across all of the file systems. Virtual Provisioning will allow file system clients to see the total potential capacity of the file system (but the extra capacity will not be assigned to the file system - unless it is required for automatic file system extension).

File systems that have difficult to predict rates can use auto-extension without a maximum size defined so that any space required by the file system will be available on demand. Because all file systems in a shared pool can make use of new disk space added to the pool, accommodating additional growth does not require migrations or other administratively-intensive procedures or outages.

Minimize current capacity usage with deduplication

Each individual file system can free up currently allocated space (that can later be used for future growth needs) by deduplicating static data. For most data sets, deduplication will reduce the storage required for the data set by 30-40 percent, which translates into a savings of 1.5 TB to 2 TB for XYZ Corporation. These savings rates should remain constant over time (as new active data is added in the future, a corresponding portion of currently active data becomes static). Deduplication also reduces the amount of future capacity required by 30-40 percent. This further justifies reducing the amount of excess capacity that must be held in reserve for future growth.

Customize storage cost and performance characteristics with storage tiering

Since not every data set (nor every file within a data set) has the same performance requirements, XYZ Corporation should tailor the type of storage used for each file system to the specific performance requirements of the data set stored on the file system. This can be accomplished by creating file systems from storage pools populated with drive and RAID types that have specific cost and performance characteristics (for example an EFD-based performance pool, a RAID 5 FC pool, a RAID 6 based low power SATA pool, and so on). Further customization at the individual file level can be achieved by using the automated storage tiering architecture to distribute individual files amongst storage tiers based on file access patterns. This will not save overall capacity, but it will save capacity on the higher-performing tiers by limiting their usage to active files that will actually benefit from the improved I/O rates offered by high performance drives.

Centralize and automate administration

Consolidating all of their file systems onto a single Celerra centralizes the administration of the storage architecture. File services for Windows and UNIX clients are now administered from a single system using storage features that are consistent regardless of whether the data set is served to clients as a CIFS share or an NFS export. File system expansion, snapshots, tiering and deduplication policies are automatically enforced by the storage architecture with minimal administrator involvement required. Predictive monitoring and notifications give the administrator information about usage trends and advanced warning about anticipated future administrator interventions that may be required.

Conclusion

Storage efficiency techniques enable reductions in allocated disk capacity, power consumption, cooling requirements, and storage administration overhead. In addition, placing active data on higher performance storage tiers improves I/O rates and reduces access latencies. Archiving static data to deduplicated, low cost, low power disk drives dramatically reduces the costs of storing static data, without restricting client access since the archived content remains online and fully accessible. Automated proactive enforcement of pre-defined storage administration policies minimizes the need for reactive management intervention to respond to routine conditions such as a file system reaching a capacity high water mark or the gradual accumulation of static data in a file system.

The combined use of Celerra advanced features like auto-extension, deduplication, storage tiering, EFDs, Virtual Provisioning, snapshots and storage pools facilitates implementing efficient storage architectures and enables automated storage tiering.

References

The following documents, located on Powerlink®, provide additional, relevant information. Access to these documents is based on your login credentials. If you do not have access to the content listed below, contact your EMC representative:

- *Achieving Storage Efficiency through EMC Celerra Data Deduplication – Applied Technology*
- *EMC Rainfinity File Management Application Installation and User Guide*
- *EMC Rainfinity File Management Application Release Notes*
- *EMC Rainfinity File Management Appliance – Applied Best Practices*
- *Improving Storage System Efficiencies with EMC Celerra – Applied Technology*
- *Managing Celerra Volumes and File Systems with Automatic Volume Management – Technical Module*
- *Using Celerra Data Deduplication – Technical Module*
- *Using EMC Celerra FileMover – Technical Module*