White Paper

EMC VNX with the Cloud Tiering Appliance

A Detailed Review

Abstract

This paper describes the EMC Cloud Tiering Appliance (CTA), which enables NAS data tiering. CTA allows administrators to move inactive data from high-performance storage to less-expensive archival storage, enabling cost-effective use of file storage. CTA also facilitates data migration, or moving data to new shares or exports.

March 2015

EMC²

Part Number H10777.2

Table of Contents

Executive summary

The EMC® Cloud Tiering Appliance (CTA) optimizes primary NAS storage by automatically archiving inactive files to less-expensive secondary storage. File tiering dramatically improves storage efficiency and shortens the time required to back up and restore data. The secondary storage can be of lower cost, such as an NL-SAS or SATA disk on a NAS device, or it can consist of public or private cloud platforms. After a file is archived, a small stub file remains on the primary storage so that the file appears as if it were in its original location. This solution makes tiering seamless to the user recalling archived data.

The CTA can also permanently migrate files from a source to a destination without leaving a stub, as when NAS server hardware requires a technology refresh.

Business case

Computer storage exists in tiers or levels, the differentiation between tiers being cost, performance, availability, or redundancy, and so forth. For example, flash storage outperforms legacy storage consisting of NL-SAS or SATA, but flash costs more per gigabyte.

Like other storage types, NAS data also exists in tiers that do not all have the same value. Typically, as data ages, users access it less frequently. To optimize use of a typical tiered NAS storage environment, customers must ensure that less-valuable, less-frequently-accessed data does not consume high-speed, expensive resources. High-speed storage should be reserved for active, important data, while less-active data should reside on less expensive storage.

Any storage tiering process ideally works automatically to guard against adding administrative overhead.

Solution overview

CTA employs Hierarchical Storage Management (HSM), a longtime staple of the mainframe world. VNX includes Distributed Hierarchical Storage Management (DHSM) or FileMover, which moves a file from primary storage to lower-cost, secondary storage. Moving the file leaves behind a small stub pointer in the file's original location. This process of relocating data and stubbing describes archiving or tiering.

CTA acts as a policy engine by interacting with VNX and identifying files that fit predefined criteria. For these files, CTA initiates movement to a lower-tier repository (such as NAS, Centera, or cloud) and places a stub file in the original location. When a client reads the stub, CTA recalls the original file from the repository. To the user, the file appears to be in its original location on high-performance VNX storage. However, instead of the entire file, only an 8 KB stub file appears on the primary tier. This architecture provides more efficient use of the most expensive, highest-performing NAS storage.

CTA can also migrate data from one location to another, like moving data from a legacy Celera system to a new VNX. CTA's migration capability moves multiprotocol data—including stub files—from one location to another. CTA enables relocation of NAS data within a NAS server, across NAS servers, and across NAS servers from different vendors.

Introduction

Scope

This paper outlines CTA features, defines functionality, and presents sample challenges that CTA helps resolve. As a technical overview, this paper also describes how to manage CTA and implement solutions in a VNX NAS environment.

Audience

This paper is intended for users who have a basic understanding of VNX Unified Storage and/or a general grasp of NAS storage concepts.

Terminology

Archive repository – A Lower-tier storage than NAS, accessible to the CIFS or NFS clients. The repository is the target of a file archival process. In an archival operation, CTA moves data from the primary or source tier to the repository and leaves a stub file on the primary source tier. The stub points to the file in the repository. A repository tier can be a NAS share/export, an EMC Centera, an EMC Atmos cloud, or a public cloud such as Amazon S3 or Windows Azure.

CAS (Content-Addressed Storage) – A method of storing data by assigning it to a permanent location on a disk. Data is unable to be duplicated or modified after being stored, and is therefore easier to retrieve due to its fixed location.

DHSM – (Distributed Hierarchical Storage Management)– See "FileMover."

File archiving – A primary CTA function that scans a NAS server for files that meet defined criteria, and moves them to a lower-tier of storage. CTA replaces the file on the NAS server with a stub file that points to the real file on the archive repository.

File migration – The movement of files from one location to another, for example moving files across NAS servers.

FileMover – A services on the Data Mover that supports file archiving on the VNX/Celera. VNX Data Movers include FileMover software. FileMover or DHSM enables the file stubbing and recall of archived files. It also provides an API that CTA uses for both archiving and migration.

File tiering – See "File archiving."

File Policy (FPolicy) – A framework on NetApp Filers that enables stubbing and recall of archived files. FPolicy also provides an API that CTA uses for both archiving and migration. CTA uses the FPolicy interface to archive files from NetApp servers.

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

Orphan file – A file in an archive that does not have a stub file pointing to it. When a file is archived, a stub on the source NAS server points to the archived file. Deleting a stub does not automatically delete the archived file. Instead, the archived file becomes an orphan. To delete orphans, run an orphan delete job on the repository.

Policy – A group of one or more rules for migration; a rule and one or more repository destinations for archiving and tiering. A policy might contain rules, for example, that would send files that have not been accessed in one year to a company's private Atmos cloud server and files that have not been accessed in two years to a public Amazon S3 cloud.

Primary storage– The storage tier that CIFS and NFS clients mount on VNX.

Source tier– See "Primary storage."

Stub – A file that replaces the original file on VNX for file when the file is archived to secondary storage. The stub file contains all metadata associated with the archived file and the information VNX for file requires for accessing the archived data on secondary storage.

Archiving

Overview

CTA provides two primary functions: archiving/tiering and migration. Archiving moves inactive data to a lower tier of storage and leaves behind a stub file.

Hierarchical Storage Management

The basis for the CTA archiving function, HSM is straightforward, robust, and time-tested.

The existence of storage tiers in most customer environments makes HSM as important today as it was decades ago. Tiered storage systems emerged as the dominant archival mode of archiving over disk and tape storage. A solution that addresses how to move data to cheaper, external storage. HSM also frees users from having to remember where data is stored and manually run commands to recall data when it was needed.

HSM scans the data, finds files that had not been accessed for specified periods of time, and automatically moves them to a lower tier of storage. In place of the file, the system stores a small stub that contains an internal pointer to the file. The user sees the stub, which looks like the actual file. When trying to access the file, the system tells the user to wait while it automatically retrieves the data and restores it to the original location on the user's disk.

FileMover

FileMover is service for VNX/Celerra that allows HSM-style archiving. Every VNX/Celerra that functions as an archive source must have the FileMover API enabled and configured. At a basic level, FileMover intercepts client access to data and takes action before the client accesses the data. CTA is an external system that can direct FileMover. Using CTA, you can define a task that directs FileMover to perform a series of actions, for example:

- Scan a share for files that are more than 60 days old
- Move the files to the archive
- Replace the files with stubs
- Activate the CIFS offline bit on the stubs

Once stubs files are in place, FileMover monitors them. When a client tries to read or write to the file, FileMover intercepts that access request and recalls the archived data using the stub's information.

FPolicy

FPolicy is an API for NetApp that is similar to FileMover. CTA needs an API like FileMover or FPolicy to archive from a NAS system. Because these types of APIs are not available on other platforms such as Linux or Windows, CTA can only archive data from VNX, Celerra, and NetApp.

Archive Policies

A CTA policy for archiving/tiering consists one or more of rule(s) and destination(s). An example of a simple policy is: If this file has not been accessed in six months, send it to the Atmos cloud and replace it with a stub.

The one-rule, one-destination policy is common, and many CTA users use this type of policy on their data. However, CTA rules are flexible. CTA supports the creation of more complex rules that archive data to multiple tiers. For example, a policy consisting of multiple rules might look like this:

> If any file has not been accessed in more than one year and is larger than 1 MB in size, send it to Isilon and leave a stub. However, if the file is a PDF, don't archive it at all. Then when these PDF files have not been accessed for two years, move them from the Isilon to the Atmos cloud, and update the stub file to point to the new location.

Policy rules are based on attributes such as access time, modify time, inode change time, file size, file name, or directory name. The archive policy action is either "archive" or "don't archive." A single expression or combination of expressions defines the archive policy.

A policy does not contain a share name. You can define one policy to evaluate shares A, B, and C, and define a second policy to evaluate share D. You can also establish several different policies to evaluate a single share.

Providing data to CTA

Administrators usually direct a CTA archive policy to evaluate a file system, CIFS share, or NFS export. CTA scans the files and applies the policy rules to each file, one rule at a time. If there are multiple rules in the policy, CTA continues to apply the rules until a rule evaluates to "true." It then takes the action associated with the rule (such as "archive" or "don't archive") and moves on to the next file.

A CTA policy can also be directed at specific files. Instead of directing CTA to scan an entire CIFS share or NFS export, you might instead import a list of filenames; then CTA only scans and applies the archive policy to the files in that list. This feature is primarily for third-party vendors with software products that have their own scanning systems, but want to use CTA archive engine. The *Cloud Tiering Appliance Getting Started Guide* describes the file ingest feature and is available from EMC Online Support.

Scheduler

The CTA scheduler sets the job start time. For example, you can schedule a batch job to start at 2:00 a.m. on Saturday to scan share01 and evaluate the files with a policy for archiving.

Administrators usually schedule a job to run on a regular cadence: weekly, every other week, or monthly. The first archive job often archives the most data and can require a fair amount of time to run. Future jobs will move incremental amounts of data and will run faster.

Simulation

CTA can simulate archive jobs. You can schedule an archive job with a policy, but run it in simulation mode. In simulation mode, CTA scans the source share and applies the policy rules against each file, but does not take any archive action. Instead, CTA tracks the number of files and amount of data it would have archived, and displays a report at the end of the simulation. Simulation is a good way to test the effectiveness of a policy and to edit the policy rules before running a real archive job.

Recall

When a file has been archived to a repository and a stub appears on the source NAS share, the NAS client expects the stub to look and behave like the original file. File recall is the process by which the user clicks on the stub file and quickly accesses the original.

The stub file contains information needed to find the actual file. VNX sets the offline bit on the stub when the file is archived. When a user attempts to read a stub file, FileMover interacts with CTA to intercept the read request and begin recalling the file from the repository. If the repository is on a CIFS share or NFS export, FileMover recalls the file using CIFS or NFS. If the repository is CAS or cloud (such as Centera, Atmos, S3, or Azure), then VNX sends the recall request to CTA, which then executes the recall and passes the file to VNX.

After recalling the file, VNX performs one of these actions:

- Passthrough recall— Provides the file to the user, but leaves the stub in place,
- Partial recall— Recalls only as much of the file's contents as it needs to satisfy the client read request and saves the partial file on VNX
- Full recall— Writes the file back to its original location and deletes the stub

The `fs_dhsm` command on VNX sets the recall style and can be configured on each file system.

Recall using CTA-HA

If an archive or migration job batch job fails, there is no data loss. Simply correct the problem and rerun the job. For this reason, the complexity of a High Availability (HA) configuration for archival or migration is not necessary or justified.

However, recalls are mission-critical because they affect clients' ability to access their data. Therefore, configurations where CTA is in the recall path require an HA. These instances might include archiving from VNX/Celerra to Centera, Atmos, S3, Azure, or all archival from NetApp.

CTA-HA is a recall-only version of CTA. The HA configuration pairs the CTA-HA physical appliance or the CTA/VE-HA virtual appliance with one or more CTA or CTA/VE systems. If the source NAS server cannot perform the recall, either CTA or its CTA-HA partner can do so.

By creating a DNS hostname that maps to the IP addresses of both CTA and CTA-HA, and by configuring VNX to find CTA using that hostname, VNX can use both recall hosts in a round-robin fashion. This setup balances the recall load. If one recall host fails, the other can perform recalls until the failed host returns to service. This configuration also allows maintenance of one host while the other continues to perform recalls.

CTA-HA also performs keystore replication for encryption keys that are required when the encryption feature is enabled during archival to Atmos, S3, or Azure clouds. Without CTA-HA, encryption does not work because the key generator does not generate keys without successful key replication daemon running.

Archive requirements for the source NAS server

CTA can archive data from CIFS shares or NFS exports on VNX, Celerra, or NetApp NAS servers. FileMover for VNX/Celerra and FPolicy for NetApp both provide archiving services. To identify stub files, both FileMover and FPolicy read the offline bit on the stubs. The CIFS protocol supports offline bits, but NFS does not; VNX/Celerra handles offline bits internally for NFS-only archival.

CTA communicates with VNX and Celerra using the DHSM API. Before archiving data from a VNX/Celerra, the CTA must be configured with the details of the source array. CTA and FileMover automatically create the DHSM connections when needed. The

Cloud Tiering Appliance Getting Started Guide available on EMC Online Support provides the configuration procedure for CTA with VNX/Celerra.

Deleting the DHSM connection on a VNX/ Celerra Control Station can optionally trigger a recall of all stubbed data from the repository linked to VNX/Celerra file system using that connection. Before triggering a recall, ensure there is enough space available on the system for all of the recalled data.

CTA and CTA-HA must have "full control" Windows permissions for the CTA archive user. Further, local administrator and backup permissions are needed on the source CIFS server. If the source includes NFS exports, these exports must have root and read/write permission for CTA and CTA-HA IP addresses.

When archiving from CIFS shares, the source server must belong to a domain and the CTA configuration settings require a username from that domain. The user must be in the local administrators and backup operators group of the CIFS server associated with the source. CTA must be configured with the fully qualified domain name and IP address of the Domain Controller for Kerberos authentication.

Archive requirements for the target repository server

CTA can archive to three kinds of repositories:

- NAS (CIFS or NFS), such as VNX, Celerra, VNXe, Data Domain, Isilon, Windows, or NetApp
- CAS such as Centera
- Cloud such as Atmos, Amazon S3, or Windows Azure

Each repository has slightly different configuration requirements:

- The requirements for NAS repositories are similar to those for source servers.
    - A CIFS domain user must be in the local admin group and backup operator of the CIFS server.
    - NFS exports must have the CTA and CTA-HA's IP addresses added for root and read/write permissions.
- A Centera repository requires a PEA file, user profile, or "anonymous."
- A cloud repository requires a tenant user for Atmos, a bucket user for Amazon S3, or a container for Azure.

One repository can serve as an archive target for multiple CTAs, and one CTA can archive to multiple repositories.

A CTA repository migration job moves all archived data from one repository to another, and updates the stubs to point to the new location.

Only CTA or CTA-HA can access CTA repositories. The NAS share that serves as the repository is visible, but the layout of archived data is proprietary. Changes to the repository can render archived data unrecallable.

Compression and encryption

Compression and encryption are options when archiving to cloud repository tiers such as Atmos, Amazon S3, or Windows Azure. A policy option enables encryption and compression. Compression can be configured to be either fast or strong.

To enable encryption, the prerequisites are:

1. Configuration keystore replication between CTA and CTA-HA
2. Generation of an encryption key using CTA GUI

CTA stores the key in the keystore and replicates it to CTA-HA. Every archive task that uses a policy with encryption uses the key. If CTA generates and replicates a new key, it only applies the key to new archive tasks that use encryption. The old keys remain in the keystore and continue to apply for files encrypted using the old keys.

Keystore replication is sufficient to sustain normal outages. However, EMC recommends backing up the CTA configuration to preserve the keystore after generating a new key.

Multi-tiered archive

CTA supports multi-tiered archive, a feature used to specify how files of, for example, different ages can be stored to different types of repositories.

Consider the following example of a multi-tiered archive:

> Find NAS files that have not been accessed in six months, and archive them to my Atmos private cloud storage. Find NAS files that have not been accessed in one year, and send them to the Windows Azure public cloud. Also, if any archived files on the private cloud have not been accessed for one year, move the files from the private to the public cloud, and update the stub files to point to the new location.

By creating a multi-tiered policy type with several rules, each with a different repository, you can design archiving schemes to fit your needs. For the example above, the multi-tiered archiving policy would have these rules:

1. If atime › one year, archive to Windows Azure public cloud
2. If atime › six months, archive to Atmos private cloud

The order of the rules is important because they are applied sequentially. When the first rule evaluates as true, CTA takes the action this rule specifies, either “archive” or “don’t archive.” CTA does not apply the subsequent rules, and the policy moves on to the next file.

In this example, reversing the order of Rule 1 and Rule 2 would produce an unintended result. The six-month old rule would be applied first, and the one-year old rule would never be applied because any file older than one year is also older than six months. All data older than six months would be archived to the private Atmos cloud, and no files would be archived to the Amazon S3.

CTA database

When a file is archived to a repository, the stub on the source NAS tier points to the file location in the repository. However, the file in the repository has no pointer back to the source, meaning that repository files have no connection to the source. Archived files are not named the same way on the repository as they are on the source. The CTA database solves this problem. Each time a file is archived, an entry in the CTA database records the file's original location on the source and the file's location in the repository. The database includes entries for every archived file.

CTA does not use the database for recalls because the stub on the source includes the information required to locate the file in the repository. The database contains statistical data and orphan information. To protect the database, schedule regular CTA backups. If a CTA fails, import the most recent database backup into a new CTA.

Stub scanner jobs

For every scheduled archive job, CTA automatically schedules a monthly Stub Scan task. Stub Scan is a utility that reads the stubs in a share and compares them to the entries in CTA database. If stubs move to different locations or if orphans appear, Stub Scan will ensure that CTA database is kept current.

Because a stub on the source has the information necessary to recall a file from the repository, CTA does not need to query for stub and repository file locations in the CTA database. However, CTA can more efficiently manage the repository storage if information in the database is in sync with the system. You can run the Stub Scan task more frequently than the default of 30 days, but it is generally not necessary.

Orphans

If stub files are deleted from the source repository, the actual files in the repository become orphans and are not automatically deleted. They are not deleted automatically because many NDMP-based backup programs and checkpoints also back up stubs by default. Generally, storage administrators also back up stubs when backing up CTA-archived file systems. If a backup is restored, the stubs must point to something. If CTA had deleted archived files when the stubs were deleted, restoring the backup would restore stubs that point to nothing.

To delete orphan files and recover space on the repository, run the Delete Orphans task. Do not delete orphans until you are certain that doing so will not restore stubs that point to orphans. For example, if keep backups are kept for six months, then define the orphan deletion job to delete files that have been orphans for at least that long.

The CTA database and the Stub Scan jobs play important roles in managing orphan files. Every time the Stub Scan sees a stub, CTA records a "last seen" time in its database. If the stub is deleted, the stub scanner identifies the file in the repository

that was linked to the stub as an orphan. Because it records the “last seen” time in its database, CTA knows how long the file has been an orphan. CTA uses the orphan age to determine which orphans to delete.

If the CTA database is lost, the location and age of orphan files in the repository are also lost. CTA database backup is therefore an important process. With protected repositories and small stubs, a NAS server that employs CTA benefits greatly from faster backups and smaller backup windows of data on the primary tier.

Reporting

CTA generates reports on the files it archives or migrates. For archived files, the reports display the size, number of files archived, and breakdown by file types, but CTA does not give a detailed profile of the data in the file system.

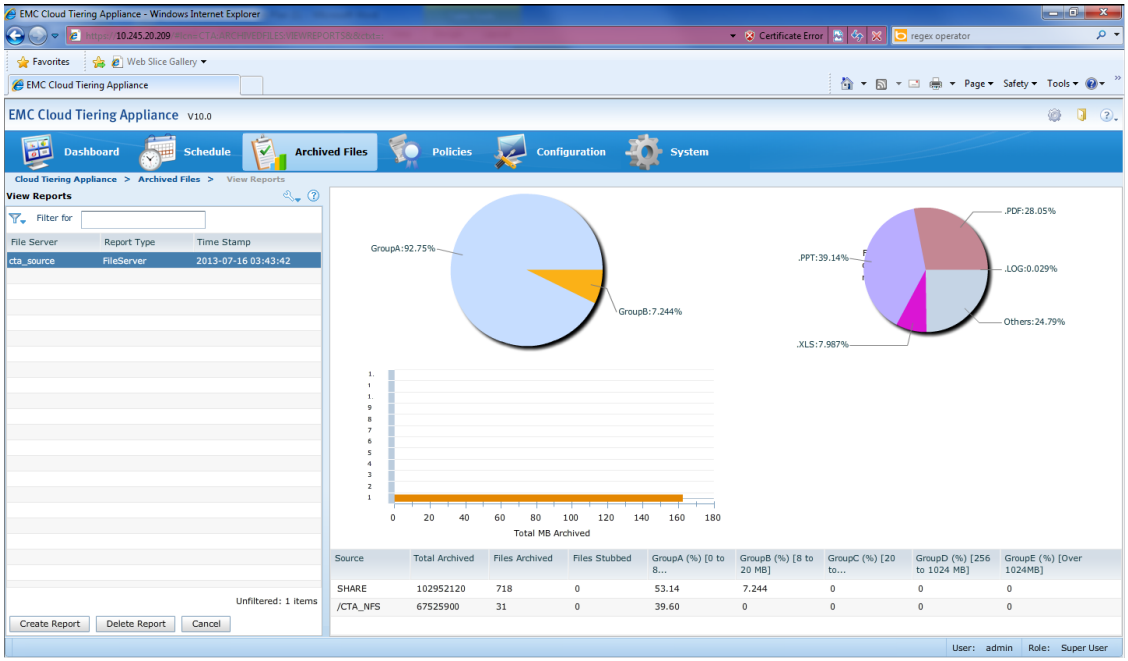

Figure 1 Sample Report

Archive simulations can help obtain information on file ages. For example, running multiple simulations that filter for access times of various ages yields an age profile for the files in the file system. Simulations do not actually move data as specified in the policy. Instead, they show the specific files that would have been archived had the actual archiving taken place. Running simulations is a useful way for you to verify that your archival policy filters files as intended.

Migration

Overview

CTA provides two primary functions: archiving /tiering and migration. Migration moves files from one location to another.

A migration is a batch job that moves data from a source CIFS share or NFS export on one system to a target. The target must be large enough to hold the data being moved, but it does not need to be the same size or have the same layout as the source. This is useful when replacing arrays. Administrators can use CTA to move data from the legacy array to the new array with minimal disruption to NAS clients. CTA can perform multi-protocol, incremental, stub-aware, cross-vendor migrations that can greatly reduce the effort and complexity of implementing new NAS technology.

CTA supports CIFS, NFS, or multi-protocol CIFS/NFS file systems. The supported source platforms for migration are VNX, Celerra, NetApp, and Windows. The supported target platforms are VNX, VNXe, Celerra, and Isilon. CTA can migrate data from a VNX, Celerra, or NetApp source to any supported target. However, when CTA migrates data from a Windows source server, only VNX or VNXe can be used as targets.

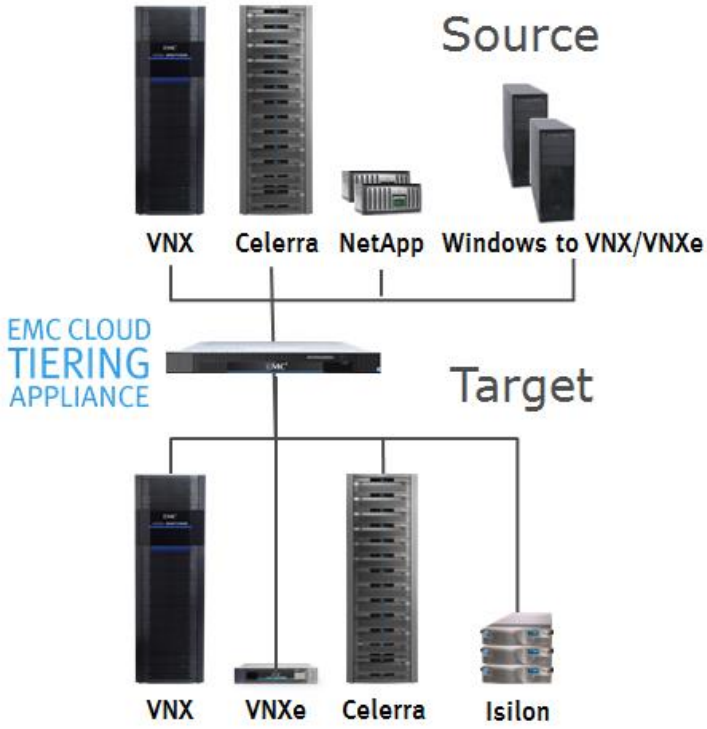

Figure 2 Supported platforms for migration

CTA needs the same kind of access permissions on the source and target for migration tasks as for archiving tasks. For a source CIFS server, a CIFS domain user and backup operator must be added to the local admin group of the CIFS server. For a

source NFS server, NFS exports must have root and read/write permissions for the CTA IPs.

Refer to the *Cloud Tiering Appliance Getting Started Guide* for more details on how to configure CTA for migration.

Migration source

CTA migration uses NDMP as its file transfer engine when migrating from VNX, Celerra, and NetApp. NDMP-style migrations are policy-based. So to use CTA for migration, you create a policy, similar the way you create archive policies.

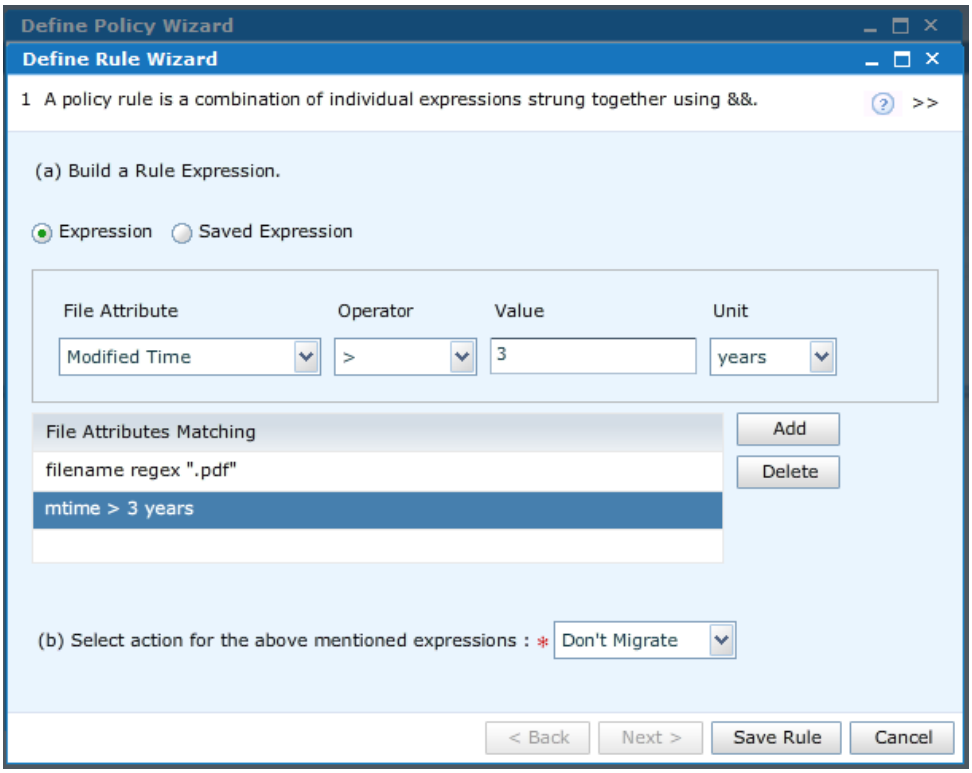

Figure 3 Creating a rule

For example, to migrate all the data in share1 to newshare1 but exclude PDF files that are more than three years old, create a rule to omit these files from the migration. Figure 3 shows this type of rule.

Copying everything without filtering requires a simple policy with a rule such as:

1. If size ›= 0 bytes, then migrate

CTA uses the FileMover to create snapshots and migrate files from VNX/Celerra using NDMPCOPY. CTA creates snapshots on the source file system by making API calls. Snapshots enable users to continue to access and write to the source share while the migration is taking place.

An NDMP user must exist on the source server for NDMP-style migrations. The server configuration on CTA requires the NDMP userid and password.

Note: File migration does not use CTA-HA.

CTA uses EMC CTA Migration Windows Service (EMWS) when migrating from Windows which is not policy-based. When migrating from Windows to VNX, CTA copies all the data. To perform migrations using a Windows source, install the EMWS copy agent on the Windows server.

Migration targets

Before starting a migration task, top-level FileSystems must already exist on target servers. CTA does not create them automatically. Migrating files to the root level requires that the directories be empty, except for system directories such as lost+found. However, files can be migrated to any empty directory in a non-empty share.

The protocol of the target (CIFS, NFS, or mixed protocol) must match the protocol of the source. For the NDMP-style migrations, an NDMP user must exist on the target server. This user does not have to be the same as the NDMP user on the source server. The server configuration on CTA requires the NDMP userid and password.

Migration process

Migrations run as scheduled batch jobs. Schedule the migration task on the CTA Schedule page or through the CLI, where you specify this information:

- A source share
- A policy for NDMP-style migrations (except for Windows-to-VNX EMCopy-style migrations, which doesn't accept policies)
- An empty target share or directory

When the job begins, CTA creates a checkpoint on the source. Then CTA copies all the data to the target, applying the policy where applicable. The migration can begin anywhere in the source share (for example, at the top level or in a subdirectory), and can move files to any empty directory on the target.

After the first pass completes, the target has a copy of the data from the snapshot on the source. However, if users have continued updating the source during migration, the source share will probably have changed from the snapshot. You can run a second pass to pick up those incremental changes. CTA will create another checkpoint, compare the second to the first, and pick up changes such as new files, deleted files, metadata changes, and so forth.

You can continue running passes as needed to pick up changes. For the final pass, put the source in offline or in read-only mode to ensure that the target is identical to the source. To complete the migration, transfer the client mounts to the target. The target has copied of all CIFS and/or NFS ACLs or permissions. After transferring the

clients, delete the source share and recover its storage space on the source NAS system.

If you wish to throttle CTA migration tasks, you can specify a maximum bandwidth rate when creating the task schedule. This way, the migration does not consume all the network bandwidth. You can also use the CTA SID translation tool to create a SID mapping of local to domain SIDs. CTA applies the mapping at migration time to ensure that the SIDs on the target will be correct. This guards against having permissions issues after the migration is complete.

CTA can run migration tasks in a continuous mode. The migration task has the option of running incremental migrations until reaching a files-moved threshold. For example, you might want to continue running incremental migrations until fewer than 1000 files are moved in one run. At that point, you would want the system to stop and notify you. Administrators typically perform incremental runs during scheduled off-hour periods over the course of several evenings, with the final locked cutover also performed during off hours.

CTA migration can also be asymmetrical, meaning that CTA can migrate between file systems of different sizes or disk layouts and across different platforms. CTA is also unique because it provides profile-filtering, multi-protocol migration (for example, from NetApp “multi” to VNX “multi”).

In addition, CTA can migrate stub files. If the source and target both support CTA stubs and are properly configured, CTA migrates stubs without recalling the archived data and the stubs continue to function after migration. If the source is VNX or Celerra and the target is Isilon or VNXe (which do not support stubs), CTA migrates the actual files and does not migrate stubs.

Other CTA interactions with VNX

CTA supports VNX File-Level Retention (FLR) on the repository. When archiving to a VNX, Celerra, Centera, or Data Domain, you can specify retention times on the archived data. Retention times on stub files can also be specified, and FileMover manages the stub retention on the source. CTA does not support FLR-enabled source file systems.

Historically, the access times during a migration could not be preserved. NDMP always set access times equal to modification times when moving the data. However, access times during a file migration can be preserved when VNX OE for File version 7.1 is the migration source. When using any other migration source, access times during migration are not preserved.

Deduplication on the repository of a VNX greatly enhances the efficiency of the repository storage. The use of de-duplication for the source is not recommended because stub files do not benefit from de-duplication because of compression issues. That is, file deduplication enables compression, and the files that are deduplicated and compressed are likely to be the same files that are archived. So when these files are archived, they must be re-inflated anyway.

Miscellaneous

Although archiving/tiering and migration are its primary functions, CTA can perform other tasks.

- "Delete stub with policy"— The only task type that deletes stubs, the repository file that the stubs point to, and the reference to stubs from CTA database. Use the "delete stub with policy task" with caution.
- "Multi-tier stub with policy" — Similar to the multi-tier archive, except that it only scans stubs. The task moves archived data that matches the policy from one repository to another, and updates the stubs to point to the new location. This is different from moving the entire repository because only files matching the policy rules are moved.

Summary

The Cloud Tiering Appliance (CTA) enables NAS data tiering, allowing administrators to move inactive data off of high-performance storage to less-expensive archival storage, thereby enabling more cost-effective use of file storage. CTA also enables relocation of NAS data to new CIFS shares or NFS exports, on the same or different servers, even from different vendors.

References

EMC Online Support

- *Cloud Tiering Appliance Getting Started Guide*