White Paper

BEST PRACTICES FOR REPOSITORY MIGRATION WITH EMC CLOUD TIERING APPLIANCE – CENTERA TO ATMOS

Abstract

This white paper describes best practices in using EMC Cloud Tiering Appliance to perform a repository migration from Centera to Atmos storage.

January 2012

EMC believes the information in this publication is accurate of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is”. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Part Number h10532

Table of Contents

Executive summary

The EMC® Cloud Tiering Appliance (CTA) is used to implement a tiered storage strategy through file level archiving, thereby facilitating significant storage savings. As archived data ages it may become necessary to migrate it to another storage tier to free up space on the archive tier. This migration process can be entirely automated using the Cloud Tiering Appliance. Provided that appropriate planning and setup have been done, a migration of archived data can be executed with no interruption to end user access.

Audience

This white paper is intended for use by EMC Cloud Tiering Appliance administrators as well as individuals involved in planning or performing a repository migration using CTA. This paper assumes the reader has prior technical experience with the components of NAS and CAS environments.

This document is supplemental to the published product documentation and is not intended as a replacement for said documentation. It is assumed that the reader is familiar with those documents including the *EMC Cloud Tiering Appliance and Cloud Tiering Appliance/VE Getting Started Guide, EMC Cloud Tiering Appliance and Cloud Tiering Appliance/VE Release Notes* and *CTA and CTA/VE Interoperability Matrix* before going through this document. If you have not reviewed those documents, please do so in addition to reading this paper in order to gain a complete understanding of the Cloud Tiering Appliance technology.

Introduction

Changing the location of network accessible data can have wide-ranging effects depending on many environment variables. Aspects of the environment that can be affected include:

- The physical path and cabling that are used to access the data set
- The drives and storage systems on which the data is stored
- The networking equipment used to interconnect clients and applications with network storage
- The security infrastructure used to prevent, detect, and report unauthorized access
- The services and policies being applied to the data set (quotas, antivirus, backup/replication, archiving)

Proper consideration should be given to each aspect of the storage environment to ensure the migration executes as smoothly as possible and the transition to a new storage location is transparent to end users and applications. This white paper will help administrators avoid such situations through comprehensive planning and the development of detailed processes for performing a migration.

Overview of the Cloud Tiering Appliance technology

As businesses increasingly rely on the 24x7 availability of NAS systems, administrators are required to dedicate ever more time and resources toward maintaining the storage environment while managing its growth. For storage administrators, the complexity of NAS environments as well as the near universal reliance on their continued, uninterrupted operation can cause concerns. Data migrations can become high risk events that take significant time to plan and execute. While successful completion of a migration may bring numerous benefits in addition to meeting growth requirements, the impact to end users is always the primary concern.

EMC Cloud Tiering Appliance allows administrators to minimize the disruption associated with typical CIFS and NFS migrations approaches, thus simplifying NAS management. CTA provides policy-based automated file migration capabilities to give administrators fine grained control over the dataset to be migrated and how the migration will be executed.

Flexible scheduling options for the migration within CTA allow an administrator to schedule a migration to run at off-peak hours to maximize available resources. Incremental migrations or ‘delta copies’ after the initial migration can further be used to reduce the amount of time required to bring the source and destination datasets to a fully synchronized state. This avoids the large downtime window traditionally associated with data migrations required to relocate client access and begin using the new storage location.

The CTA migration is fully out-of-band and there is no impact to end users by introducing CTA into the environment or by launching a migration. There is no requirement to remount or map NFS/CIFS clients to the CTA to perform the migration. Users continue to access the storage location through standard processes. The only action required by an administrator that will impact user access during the migration will be the final cutover window. This window can be minimized or eliminated entirely through the use of namespace technologies such as DFS, NIS, and LDAP to transparently direct users to the new storage location.

Migrating from Centera

This section describes the architecture and mechanism for archiving and recall functionality in environments utilizing CTA with Centera as an archiving destination.

Connecting to Centera

Centera systems consist of an array of clustered nodes with one of three possible roles. Nodes with the storage role are used to store archived file data. A single object archived to Centera may be saved across multiple cluster nodes and across multiple disks on each of those nodes to provide high levels of redundancy and recovery capabilities. Nodes with the access role provide Centera clients in an IP network, such as Cloud Tiering Appliance, with access to the objects saved to storage nodes.

Nodes can also be configured to perform both access and storage services. Each Centera system has a minimum of two access nodes and potentially many more.

Centera systems have one or more storage pools that are used to store archived file data. When a Cloud Tiering Appliance establishes a connection, it is in relation to a single specific storage pool serviced by the Centera system.

Prior to being allowed access to a storage pool, CTA will need to authenticate with the Centera through one of three methods to determine what access rights are applied. Anonymous authentication can be specified when defining a Centera connection on a Cloud Tiering Appliance. This method of authentication may fail as not all Centera systems will allow or support anonymous access depending upon the configuration and software version. Authentication can be performed by providing a specific username and password that will be stored on the CTA in an encrypted form. Alternately, a PEA (Pool Entry Authorization) file can be uploaded to Cloud Tiering Appliance to be used for authentication.

To allow Cloud Tiering Appliances to establish connections to Centera, you must provide a unique name for identification purposes, a connection string consisting of a list of IP addresses or hostnames identifying access nodes in a Centera cluster, and an authentication method (anonymous, username/password, PEA file). This can be done from the GUI or CLI (use **man rffm**) of a Cloud Tiering Appliance. The connection string is distributed to all CTA-HA that have registered with the CTA to provide HA recall services.

Each connection string is identified on the Cloud Tiering Appliance by a unique name. The unique name is placed into the stub data for files that are archived from NAS. When those files need to be recalled, a CTA will read the stub data and the unique name within will be used to identify the connection string and storage pool that will be accessed.

Note that a connection string will allow the Cloud Tiering Appliance to access one specific storage pool on a Centera. Therefore it may be possible or even required that multiple connection strings created on a single Cloud Tiering Appliance utilize the same set of IP addresses/hostnames but different authentication methods and different names in order to access multiple storage pools on a single Centera. In the event that a Cloud Tiering Appliance should archive data to multiple storage pools you will need to define multiple connection strings with unique names and different credentials for authentication to allow access to specific storage pools.

When defining a Centera in a Cloud Tiering Appliance, you should provide the IP addresses or hostnames of at least four access nodes as part of the connection string. If there are less than four access nodes in a cluster, then all access nodes should be specified. Each Centera cluster contains two or more nodes with the access role. In general, it is not necessary to specify more than four access nodes in the connection string.

NOTE: The total length of the connection string should not exceed 128 characters and thus when using name resolution, particular care should be taken to limit the length of the hostnames specified for access nodes.

When defining a Centera in Cloud Tiering Appliance, a connection string is specified in a comma separated value format.

An example using hostnames

Connection String: accessnode1,accessnode2,accessnode3,accessnode4

An example using IP addresses

Connection String: 10.10.0.1,10.10.0.2,10.10.0.3,10.10.0.4

For more details on how to define Centera connections in Cloud Tiering Appliance, refer to the *EMC Cloud Tiering Appliance and Cloud Tiering Appliance/VE Getting Started Guide*.

When using hostnames in the connection string to identify access nodes, the hostnames defined in DNS can be updated to force it to connect to different IP addresses. This may be necessary in the event of a disaster or if a cluster node is replaced and given a new IP. If IP addresses are explicitly used in the connection string then the Centera definition must be deleted and then added again with the new IP addresses. In order to delete the Centera definition, the Cloud Tiering Appliance administrator will need to delete all policies that reference the Centera and all schedules which reference those policies.

When a Cloud Tiering Appliance attempts to open a connection to a Centera, it will try to connect to the first IP address listed in the connection string (or the IP obtained through name resolution through DNS). If the access node is reachable at that IP address and the authentication is successful, then the CTA will query the access node for information about all other access nodes in the cluster regardless of which other access nodes were provided in the connection string. Communication between Centera and CTAs, such as archiving and recall activity, is then automatically load balanced between all access nodes that are discovered and can be reached on the network.

If the first access node from the connection string cannot be reached, the Cloud Tiering Appliance will attempt to connect to the second access node listed in the connection string. If the connection to the second access node fails, connections will be attempted to all other access nodes in turn. The connection process will only fail if all access nodes listed in the connection string are inaccessible or if the supplied credentials are not valid or authorized.

This connection process occurs each time the File Management service on a CTA or a callback daemon on a CTA or CTA-HA is started. As a result, it is important to include as many access nodes in the connection string as possible to ensure that CTA can connect to Centera even when one or more access nodes are offline. In general, specifying four access nodes is sufficient.

Effects of Centera replication

The content in a Centera storage pool can be replicated to a storage pool on another Centera at a remote site for disaster recovery purposes. Centera replication functions by copying newly created objects from a storage pool on a primary Centera to a storage pool on a secondary Centera.

Replication works asynchronously. For a period of time, new objects will exist only on the Centera where they were created. Thus in the event that a Centera fails, recall from the storage pool that is the replication target may fail for some newly archived files. To minimize this risk, Cloud Tiering Appliance should use the delayed stubbing feature for archiving tasks. This feature ensures that a minimum period of time has elapsed from when an object was written to Centera until the original file data is removed from NAS. More information on delayed stubbing is available in the *EMC Cloud Tiering Appliance and Cloud Tiering Appliance/VE Getting Started Guide*.

When replication is being used for a storage pool, the connection string supplied to Cloud Tiering Appliance must consist solely of access nodes from the source of that replication. Supplying access nodes from both Centera systems can cause the Cloud Tiering Appliance to connect to the storage pool that is the replication target and create new objects. These objects may not be replicated back to the primary Centera. This can cause data unavailability when a recall for that file is attempted from the replication source.

When Cloud Tiering Appliance connects to a Centera, it retrieves information about replication of the storage pool including the access nodes of the replication target. If all access nodes of a Centera configured for replication become inaccessible, CTAs that were connected to that Centera will use the information about the replication target to open a new connection. This new connection is used for read operations only to allow recall requests to be serviced by Cloud Tiering Appliances. Thus you do not need to include any access nodes from the replication target in the connection string.

However, if the File Management service on a Cloud Tiering Appliance or the callback daemons on a CTA or CTA-HA are started while the Centera access nodes provided in the connection string are not accessible, the information about the replication target cannot be retrieved and the connection process will fail. Therefore, it is important to avoid restarting the callback daemons when all access nodes of a Centera configured for replication are inaccessible.

In the event of an actual disaster where the primary site Centera will not be brought back online, Cloud Tiering Appliances will need to be manually reconfigured to use the replication target as the primary Centera either by editing the connection string on Cloud Tiering Appliance directly or when using hostnames by updating the DNS entries for the Centera access nodes.

Overview of the archiving process

When archiving to Centera, the CTA archiving threads create C-Clips and BLOBs on Centera.

For NFS files, a BLOB is created to hold the file data and it is attached to a C-Clip. The clip has keys that hold the original last modified timestamp of the source file, owner UID, and privileged GID, and mode bits.

For CIFS files, a BLOB is created to hold file data and it is attached to a C-Clip. The clip has keys that hold the original last modified timestamp of the source file. The NTFS DACL and alternate data streams are created as embedded BLOBs inside of the clip as long as they are less than 100 KB. When alternate data streams or the NTFS DACL exceed 100 KB, they are written into additional BLOBs and attached to the C-Clip.

Overview of the recall process

Data archived to Centera is always recalled by a CTA or CTA-HA. Stub files on primary storage contain the Centera Clip ID that uniquely identifies the object containing the archived file data.

In Celerra environments, blades will open up to 32 HTTP connections to CTA for each DHSM connection string that references it.   The HTTP connections for each DHSM string can be used to parallelize the recall of a single file.

For full recalls in NetApp environments, the appliance software applies BLOB slicing to read small sections of file data from multiple Centera access nodes concurrently to parallelize the recall of a single file. For pass-through recalls with NetApp ONTAP 7.3 and above, recall requests are sent to the CTA or CTA-HA for specific data sections concurrently based on client I/O.

Overview of the migration architecture

The following actions are performed when a repository migration task is launched:

1. The CTA database is queried for all files archived without retention to the source repository specified in the migration task.
2. The original path for the stub is checked to determine if the stub exists.
    a. If the stub cannot be found at the location in the CTA database (moved, renamed, deleted, etc.) no action will be taken.
    b. If a stub is found at the location but it does not have matching metadata with the record in the database, no action will be taken.
    c. If a non-stub is found at the location, no action will be taken.
3. If the matching stub file exists at the location in the database, the archived file is copied from the source Centera to the destination Atmos.
4. The CTA database is updated with the new destination path for the stub once the archived data is successfully migrated.
5. The stub file is updated with the new secondary path and metadata to the corresponding object on Atmos.
6. If there is no delay period specified in the migration, the file in the source Centera is removed.

With a delay period specified in the migration task, a background thread will query the CTA database on a daily basis to determine files where the current system time exceeds the update time recorded in the CTA database. When an entry is found meeting this requirement the original destination file on the source Centera is removed.

Migrating to Atmos

This section describes the architecture and mechanism for archiving and recall functionality in environments utilizing CTA with Atmos as an archiving destination.

Connecting to Atmos

Atmos systems consist of an array of clustered nodes that run multiple services responsible for storing and retrieving file data and metadata. Each node typically runs the storage service to manage the disks where user data is stored. Nodes that run the web service to process API access are called access nodes. These nodes provide Atmos clients in an IP network such as CTA with access to the objects saved to the Atmos system. Atmos nodes can be deployed as an access node while also performing storage services for objects. Each Atmos system has a minimum of two nodes and potentially many more.

Prior to being allowed access to storage resources, CTA will need to authenticate with the Atmos system. CTA can connect to the web services running on access nodes using HTTP or HTTPS to connect securely. An Atmos system is configured in CTA using a user ID and shared secret generated by the cluster. The user ID corresponds to a tenant or subtenant admin which has access to the cluster’s storage resources. The credentials configured for Atmos allow CTA to securely access the Atmos storage delegated for the specific tenant or subtenant.

To allow Cloud Tiering Appliance to establish connections to Atmos, you must provide a unique name to identify the system, a DNS hostname to resolve the access node IP addresses, a transport protocol (HTTP or HTTPS) and the authentication credentials. This can be done from the GUI or CLI (use man rffm) of a Cloud Tiering Appliance. The connection string is distributed to all CTA-HA that have registered with the CTA to provide HA recall services.

Each Atmos is identified on the Cloud Tiering Appliance with a unique name. This identifier is placed into the stub data for files that are archived from NAS to Atmos. When those files need to be recalled, a Cloud Tiering Appliance will read the stub data and the unique identifier within will identify the connection configuration required to access the storage resources.

When defining an EMC Atmos in a CTA, the DNS hostname provided should resolve to multiple access node IP addresses. When a Cloud Tiering Appliance needs to connect to Atmos, it will first resolve the hostname using a DNS lookup. It will cache all returned IP addresses and load balance requests across the available nodes. The DNS hostname configured in CTA should resolve to at least two access nodes on the

EMC Atmos and preferably more for further load balancing. In the event of an Atmos node failure, CTA can still connect to the system using the remaining available nodes.

In the event of a disaster with Atmos, the DNS hostname defined in CTA can be updated to point to the new cluster IP addresses. This may also be required when adding or removing nodes from the Atmos system. CTA will automatically detect changes to the DNS configuration to remove unresponsive nodes and add new nodes.

Effects of Atmos replication

The objects stored in an Atmos system can be replicated to a system at another location for disaster recovery purposes. Atmos replication uses policies to define how newly created objects and their respective metadata should be copied to other Atmos systems or geographic locations. These policies specify where the first object instance should be stored, the location for every subsequent replica and whether the replication is synchronous or asynchronous.

Atmos replication allows for granular definition of each replica including the replica type (synchronous or asynchronous). Synchronous replication provides bit-for-bit copies at each location that are identical at any point in time. It guarantees that all replicas are successfully written before acknowledgment is sent to the client. This type of replication can have performance ramifications based on the number of replicas and the latency between sites. Asynchronous replication does not guarantee every replica is up to date before acknowledging the client but does not suffer from the same performance effects as synchronous replication.

When asynchronous replication is used, newly created objects will exist only on the Atmos system where they were first written. If this Atmos system fails before the objects can be replicated to another location, recalls will fail from the replication target for newly archived files. To mitigate this risk, CTA provides the delayed stubbing feature for archiving tasks. This feature ensures that a minimum period of time has elapsed from when an object was written to Atmos until the original file data is removed from NAS. More information on delayed stubbing is available in the *EMC Cloud Tiering Appliance and Cloud Tiering Appliance/VE Getting Started Guide*.

When replication is being used for an Atmos system, ensure that the DNS hostname used by CTA to connect to Atmos is defined using only the IP addresses for the primary system. Configuring the hostname with IP addresses from Atmos systems at multiple locations can cause the CTA to connect to one or more replication target and create new objects. If the objects are not replicated according to the Atmos replica policy, data unavailability can occur when a recall for those objects is attempted using the original replication source.

Overview of the archiving process

When archiving to Atmos, the CTA archiving threads create objects on Atmos.

For NFS files, an object is created to hold the file data and it has a unique object ID (OID). The file metadata is stored on Atmos separately from the object containing the file data and is used only to recreate stubs created by CTA. The metadata stored by

Atmos includes among others the original file size, file path, timestamps and the original owner UID, GID and mode bits.

For CIFS files, an object is created to hold the file data and it has a unique object ID (OID). The file metadata is stored on Atmos separately from the object containing the file data and is used only to recreate stubs created by CTA. The metadata stored by Atmos includes, among others, the original file size, file path, timestamps and the original NTFS DACL including owner SID. If there are Alternate Data Streams (ADS) with the original file, they are stored as a new object on Atmos and the object is linked back to the object created for the file data.

Overview of the recall process

Data archived to Atmos is always recalled by a CTA or CTA-HA. Stub files on primary storage contain the Atmos Object ID (OID) that uniquely identifies the object containing the archived file data.

In Celerra environments, blades will open up to 32 HTTP connections to CTA for each DHSM connection string that references it. The HTTP connections for each DHSM string can be used to parallelize the recall of a single file.

For full recalls in NetApp environments, the appliance software reads small sections of file data for the requested object from multiple Atmos nodes concurrently to parallelize the recall of a single file. For pass-through recalls with NetApp ONTAP 7.3 and above, recall requests are sent to the CTA or CTA-HA for specific data sections concurrently based on client I/O.

Pre-Migration tasks

Stub scanner

In order to properly migrate archived data from Centera to Atmos, CTA relies heavily on the locations of stub files within its database. Once archived content in the source repository is identified from the database, CTA attempts to locate the corresponding stub file on primary storage using the original path. As described in the 'Overview of the migration architecture' section, CTA must find and validate the matching stub for the archived data or the data will not be migrated.

Stub scanner tasks are critical to updating the CTA database with the most recent location of stub files on primary storage. When stub files are renamed, moved or deleted after archiving, the stub scanner tasks update the original stub path to the latest path. The stub scanner inspects the stub content to determine the stub's matching offline content and updates the associated record in the CTA database with the stub's current location.

Stub scanner tasks should be run for every dataset containing files archived to Centera before running a repository migration. The tasks should be completed close to starting the repository migration task to ensure that the stub paths are as accurate as possible. This helps to ensure that the stubs are migrated and updated by the task and not incorrectly identified as orphans. In some cases it may be necessary to re-run the stub scanner tasks between each repository migration run to catch stubs that were modified or relocated and therefore not migrated by the previous task run.

It is important to inspect the stub scanner task log for each run to verify that it completed successfully without any errors. Failures to read stub data or properly update the CTA database with stub information can cause the repository migration task to not identify the stubs if their location has changed. The log file can be viewed from the GUI by selecting the **View Log** link within the stub scanner Task Summary page. These logs are also accessible from the CLI and are stored in the */var/log/rainfinity/filemanagement/fws/support/* directory. The log file is named fwst-xx.log where 'xx' represents the task ID.

Orphan management

Stub scanner tasks are critical to updating the CTA database with accurate information regarding stub files on primary storage. They are also important to identifying stub files that have been deleted and the subsequent orphaned data on Centera. As with stubs that have been relocated and cannot be found by the repository migration task, orphaned data on Centera is not migrated to Atmos.

Before running a repository migration task it is important to run orphan management on CTA to clean up orphaned data on Centera. This data can come from stub files that have been deleted intentionally, files that have been fully recalled or previous versions of archived data. As the stub files for this content on Centera no longer reside on primary storage, it will not be copied by the repository migration task to Atmos. Additionally an error message is printed in the migration task log when a stub cannot be found so having a high number of orphans can cause the log to grow significantly.

It may be necessary to run stub scanner tasks for each dataset to update the CTA database before running orphan management if they have not been run recently. Once those tasks have been run and completed successfully, orphan deletion policies and tasks can be created on CTA to begin cleaning up the Centera. Note that orphan deletion tasks won't delete data under retention or any files that have been detected by a stub scanner task within the last 30 days by default.

Orphan management is not mandatory before running a repository migration task but if there are a lot of orphans in the CTA database, it will improve the task efficiency. The database entries for the orphaned Centera content will not be affected by the migration to Atmos so orphan cleanup can still be done post-migration. This may be necessary if the data is still under retention but the retention period will expire soon or if the minimum "missing" period for orphans has not been exceed.

Launching a migration

When Cloud Tiering Appliance is used to migrate a Centera repository to Atmos, the migration process will generally involve the following set of actions:

1. Configure the CTA with the new Atmos destination.
2. Perform various safety checks (LAN/WAN bandwidth, security infrastructure, recall capability, etc.) to ensure the migration is safe.
3. Execute stub scanner task(s) for all datasets archived to the source Centera.
4. Determine appropriate scheduling options for the migration.
5. Create a new repository migration task in CTA and execute a simulation.
6. If the simulation executes successfully, launch the repository migration task.
7. Monitor the task logs on CTA to ensure the migration is successful.
8. Perform a comparison of the source and destination repositories for consistency.
9. If there are files that were not migrated, determine the root cause and re-run the repository migration task. It may be necessary to re-run stub scanner tasks as described in Step 4 to update the CTA database with the latest location of stubs.
10. Review logs on CTA to ensure the migration is successful and no further action is required.

Creating a new repository migration task is described in the Online Help within CTA and the steps described there should be carefully followed.

The steps required to monitor, verify and complete a migration are described in **Monitoring a migration**.

Monitoring a migration

Administrators have several options for monitoring the progress of a repository migration. Various statistics are available from the CTA GUI using the **View Summary** link for the repository migration task under the Schedule tab.

Once the migration begins executing, the statistics include the amount of data moved thus far and the number of files moved. If any files cannot be migrated, the number of failed files will also be reported.

It may be necessary to monitor the repository migration log file. The log file can be viewed from the GUI by selecting the **View Log** link within the Task Summary page. These logs are also accessible from the CLI and are stored in the *var/log/rainfinity/filemanagement/fws/support/* directory. The log file is named fwmig-xx.log where ‘xx’ represents the task ID. Simulation logs are available under */var/log/rainfinity/filemanagement/fws/simulation/.* The simulation log is named fwmigs-xx.log where ‘xx’ represents the simulation task ID.

The Task Summary page also provides a link to the detailed file list. This list contains the name of every file that was successfully migrated by the most recent run of the migration task. The logs are also accessible from the CLI and are stored in the */var/log/rainfinity/filemanagement/fws/support/* directory. The log file is named fwmigd-xx.log where ‘xx’ represents the task ID.

To view the history of the repository migration task, select the **History** link from the drop-down menu in the status column for the task. The history overview displays all migration runs for the task. For each run, the status of the migration is displayed, the amount of files and data migrated, and the start/finish times for the run.

Finalizing a migration

During the repository migration task, stubs will be updated transparently so end users should be unaware of the relocation of the backend data. There is no requirement to have a cut-over period when the Atmos is brought online and the Centera is decommissioned. As more data is migrated off the Centera to Atmos and stubs are updated, load will gradually increase to the Atmos system and recall load should decrease on the Centera.

As described previously, there may be data residing on Centera that cannot be migrated by CTA to Atmos and the corresponding stubs cannot be updated. In most cases, this data can be described as orphaned as there is no corresponding stub on primary storage. For example:

- Stubs that have been deleted
- Stubs that have been fully recalled due to user I/O
- Stubs that have been fully recalled, modified and re-archived (previous versions)
- Stubs under retention and/or archived data under retention

As mentioned previously, CTA will not migrate orphaned data from Centera to Atmos. This can cause a discrepancy in the amount of data archived to Centera by CTA and the subsequent amount of data that can be migrated to Atmos with the repository migration task. This discrepancy however should not include stubs that are not migrated because they have been renamed or moved but still remain on primary storage. These files can be properly migrated by re-running stub scanner tasks for their respective datasets before re-running the repository migration.

Once the final repository migration run is successfully completed and all discrepancies are understood, the Centera can begin to be decommissioned. If data was archived to Centera from Celerra or VNX, all filesystem DHSM connections for the Centera can be safely deleted. When deleting the DHSM connection(s) the option '*recall_policy=yes*' must be used. For example:

```
# fs_dhsm <fs> -delete <cid> -recall_policy yes
```

Where <filesystem> is the name of the primary filesystem and <cid> corresponds to the ID of the Centera DHSM connection. This option forces a full recall of any stub files that have not been updated and still use the connection to Centera. This prevents interrupting access to any remaining Centera stubs when the Centera is fully decommissioned. Make sure before deleting the DHSM connection(s) that the primary filesystem has sufficient free space to tolerate the recall activity.

Stub file formats

This section provides detail on the stub file formats used by Celerra/VNX and NetApp.

Celerra/VNX to EMC Centera stub file format

```
<OFFLINE_ATTRS
OFFLINE_PATH="http://ccd.interop.prv/fmroot/Centera4/D9DRHVCFKCL8Te49RSIOQU54DFRG412NA3DKG90H8I
6RJFDO"
      READ_METHOD="read_pass_through"
      INFO="<?xml version="1.0" encoding="UTF-8"?>
            <RFStubInfo Version="1"
            LastModifiedTime="128333808780000000"
            LastModifiedTimeUTC="Tuesday, 2007-09-04 12:01:18+0000"
            ArchiveTime="128377409480000000"
            ArchiveTimeUTC="Wednesday, 2007-10-24 23:09:08+0000"
            FileSize="471612"
            RetentionTimeInSeconds="0"
            SecondaryStorageProtocol="CIFS"
            SecondaryStorageClass="CAS"
            SecondaryDeviceType="CENTERA"
            CenteraName="Centera4"
            CenteraCAVersion="1"

CenteraCAValue="1a0000000000000dcdedc660f3fab44112448db50570402f76ba4788d25df89908e2aa38ffdb85
            SecurityHash="17bfbbf1cba908c085d8a92174198840" >
</RFStubInfo> "
            OFFLINE_MTIME="1188907278"
/>
<STANDARD_ATTRS
            HANDLE="4294967324-22-1191536213"
            ONLINE_CTIME="1198165790000006"
            UID="32768"
            GID="32773"
            ATIME="1198165815"
            MTIME="1188907278"
            CTIME="1198165790"
            CREATE_TIME="1191536213"
            DOS_ATTRS="4640"
            PARENT_INODE="13"
            FSIZE="471612"
            BLOCK_SIZE="8192"
            BLOCKS="16"
            BYTES_USED="8192"
            INODE="22"
            DEVICE="28"
            NLINK="1"
            MODE="0644"
            FILE_TYPE="File"
            OWNER="S-1-5-32-544"
   />
```

Celerra/VNX to Atmos stub file format

```
<OFFLINE_ATTRS
OFFLINE_PATH="http://acd.interop.prv/fmroot/atmos1/4c86b897a20a1b7804c86ba40e18f104ded0fa413
        READ_METHOD="read_pass_through"
        INFO="<?xml version=”1.0” encoding=”UTF-8”?>
               <RFStubInfo Version=”1”
               LastModifiedTime=”129291263840000000”
               LastModifiedTimeUTC=”Thursday, 2010-09-16 15:59:44+0000”
               ArchiveTime=”129518552670000000”
               ArchiveTimeUTC=”Monday, 2011-06-06 17:34:27+0000”
               FileSize=”126976”
               RetentionTimeInSeconds=”0”
               SecondaryStorageProtocol=”CIFS”
               SecondaryStorageClass=”CAS”
               SecondaryDeviceType=”ATMOS”
               AtmosName=”atmos1”
               AtmosOIDVersion=”1”
               AtmosOIDValue=”4c86b897a20a1b7804c86ba40e18f104ded0fa4135bc”
               SecurityHash=”e6bd69de0272c8f22bf21bc296dba566” >
         </RFStubInfo>"
        OFFLINE_MTIME="1284652784"
/>
<STANDARD_ATTRS
        HANDLE="4294967346-90-1285085288"
        ONLINE_CTIME="1307381668000348"
        UID="10005"
        GID="10001"
        ATIME="1305052876"
        MTIME="1284652784"
        CTIME="1307381668"
        CREATE_TIME="1285085287"
        DOS_ATTRS="4640"
        PARENT_INODE="85"
        FSIZE="126976"
        BLOCK_SIZE="8192"
        BLOCKS="16"
        BYTES_USED="8192"
        INODE="90"
        DEVICE="50"
        NLINK="1"
        MODE="0644"
        FILE_TYPE="File"
        OWNER="S-1-5-32-544"
/>
```

NetApp to EMC Centera stub file format

```
   <?xml version="1.0" encoding="UTF-8"?>
   <RFStubInfo Version="1"
               LastModifiedTime="128351760459191770"
               LastModifiedTimeUTC="Tuesday, 2007-09-25 06:40:45+0000"
               ArchiveTime="128426394070000000"
               ArchiveTimeUTC="Thursday, 2007-12-20 15:50:07+0000"
               FileSize="502170"
               RetentionTimeInSeconds="0"
               SecondaryStorageProtocol="CIFS"
               SecondaryStorageClass="CAS"
               SecondaryDeviceType="CENTERA"
               CenteraName="Centera4"
   CenteraCAVersion="1"
```

```
CenteraCAValue="1a0000000000008073cc4dd3ac63d52a1c2835f9e05b04027d23284383aa5d37918f7661a
fd55"
            SecurityHash="95d5876d76ecc9dfc7f720c15645058a"
>
</RFStubInfo>
```

NetApp to Atmos stub file format

```
<?xml version="1.0" encoding="UTF-8"?>
<RFStubInfo Version="1"
            LastModifiedTime="129443262457612470"
            LastModifiedTimeUTC="Friday, 2011-03-11 14:10:45+0000"
            ArchiveTime="129443378430000000"
            ArchiveTimeUTC="Friday, 2011-03-11 17:24:03+0000"
            FileSize="167880"
            RetentionTimeInSeconds="0"
            SecondaryStorageProtocol="CIFS"
            SecondaryStorageClass="CAS"
            SecondaryDeviceType="ATMOS"
            AtmosName="rfatmos2"
            AtmosOIDVersion="1"
            AtmosOIDValue="4c168a1ea106769004c1bc57a240e804d7a5ab53453d"
            SecurityHash="a71cba3b2ce61248fa532b50dab159b0"
 >
</RFStubInfo>
```

Conclusion

This document covers the fundamentals of performing a repository migration from Centera to Atmos with Cloud Tiering Appliance. Armed with the technical details on the Cloud Tiering Appliance architecture, storage administrators will be able to implement an effective migration strategy and relocate their production CTA environment from Centera to Atmos without impacting end-users.