IBM Storage Protect

Configuring IBM Storage Protect for IBM Storage Scale Active File Management

---

Document version 1.6

Dominic Müller-Wicke
IBM Storage Defender Development Architect

Nils Haustein
IBM Client Engineering Storage EMEA

CONTENTS

1 Introduction

This paper describes scenarios and provides configuration guidance for IBM™ Storage Scale™ (previously known as IBM GPFS™[1]), Active File Management (AFM) in conjunction with data protection such as backup and hierarchical storage management (HSM). Because AFM can be configured in a variety of ways with data protection, each scenario is described.

HSM is implemented with IBM™ Storage Protect™ for Space Management (also known as UNIX HSM[2]). Backup is implemented with the tool "mmbackup," which is part of IBM Storage Scale. The tool mmbackup uses and requires the IBM Storage Protect backup-archive client to be installed on the nodes used.

Who should read this paper? The documentation was written for IBM Storage Scale and IBM Storage Protect administrators who plan to implement data protection in the AFM environment. This paper is also intended for administrators who have data protection for AFM already implemented, want to understand the challenges of this combination of technologies, and want to obtain recommendations how to deal with these challenges. In addition, the paper can help technical sales personnel to integrate the technologies into the customers' business processes.

How should this paper be read? You should identify and read the scenario that is closest to your implementation plans.

1.1 IBM Storage Scale Active File Management

Active file management (AFM) is a scalable, high-performance file-system caching layer that is integrated with the IBM Storage Scale cluster file system. AFM is based on a home-cache model. A single home provides the primary file storage that is exported. One or more caches provide a view into the exported home file system without storing the file data local. Upon file access in the cache, the data is fetched from home and stored in cache. Another way to get files transferred from home to cache is through pre-fetching. Pre-fetching can use the IBM Storage Scale policy engine to quickly identify files matching certain criteria.

[1] IBM General Parallel File System (GPFS)

[2] Note: Throughout this document, the term UNIX HSM is used to refer to IBM Storage Protect for Space Management.

When files are created or changed in cache, they can be replicated back to home. A file that was replicated back to home can be evicted in cache, whereby the user still sees the file in cache (the file is uncached), but the actual file content is stored in home. Eviction is triggered by the quota that is set on the AFM file set and can evict files based on size or last recent used criteria.

Cache must be an IBM Storage Scale independent file set. Home can be an IBM Storage Scale file system, an NFS export from any other file system, or file server (except for the disaster-recovery use case). The caching relationship between home and cache can be based on the NFS or native IBM Storage Scale protocol. In the latter case, home must be an IBM Storage Scale file system in a different IBM Storage Scale cluster. In this document, we assume the cache relation is based on the NFS protocol.

AFM masks wide area network latencies and outages by using IBM Storage Scale to cache massive data sets, allowing data access and modifications in cache even when remote home storage cluster is unavailable. The AFM implementation leverages the inherent scalability of IBM Storage Scale to provide a multinode, consistent cache of data located at a home cluster. By integrating with the file system, AFM provides a POSIX-compliant interface, making the cache completely transparent to applications. See the IBM Storage Scale documentation for more details on AFM.

The AFM relation is typically configured on the cache fileset in one specific mode (see section AFM Modes). The AFM mode determines where files can be processed (created, updated, and deleted) and how files are managed by AFM according to the file state (see section AFM file state diagram in section 1.1.2).

1.1.1 AFM Modes

The relation between cache and home can be based on different modes. The mode determines how AFM behaves. One of the following modes has to be set in cache when creating an AFM relation:

SW (Single Writer): The home export is the AFM target for one cache. Uncached files in cache are fetched from home upon access or with prefetching. Files can be modified in cache and are replicated to home asynchronously. Files that are deleted in cache are deleted in home. Files can be evicted in cache when present in home. Evicted files that are in status uncached can be seen in cache and are fetched upon access (or prefetching). Home is read-only in this mode from a user perspective (not from the AFM perspective).

IW (Independent Writer): Similar to SW mode, but home is the AFM target for one or more caches. All changes in the caches will be replicated to home asynchronously. Changes to the same data will be applied in the home fileset so they are replicated from the caches; there is no cross-cache locking. Potential conflicts must be resolved at the respective cache site.

LU (Local Update): The home export is the AFM source for one cache. Uncached files in cache are fetched from home upon access or with prefetching. Files can be changed in cache but are not replicated back to home as they become local. Also, files that are created in cache are not replicated to home. Eviction should be disabled for AFM LU mode. This AFM mode is typically used for data migration, whereby home is the legacy file system that is migrated to cache via fetching and prefetching files. File changes on cache are not applied in home.

RO (Read Only): The home export is the AFM source for one cache. Uncached files in cache are fetched from home upon access or with prefetching. Cache is read-only; files cannot be changed or deleted in cache. Files can be evicted in cache, which changes the file state to uncached. Uncached files can be seen in cache and are fetched upon access or prefetching.

DR (Disaster Recovery): AFM DR is a disaster recovery function to augment the overall business recovery solution. It asynchronously copies files from one fileset in one Storage Scale cluster to another fileset in another Storage Scale cluster and provides failover and failback capabilities. The AFM DR architecture is represented by two sites: primary and secondary. The primary site is a read-writeable fileset where the applications have read-write access to the files. The secondary site is another fileset that should be operated in read-only mode if the primary is online. All data created, modified, and deleted on the primary fileset is asynchronously replicated to the secondary fileset in the timely order the files have been changed. The primary is continuously available to the applications even in the event of communication failures with the secondary. AFM DR provides a Recovery Point Objective feature (RPO), allowing to recover the data in the secondary for a defined point in time after the primary has failed (failover). The RPO feature is based on “peer snapshots” that create snapshots on both primary and secondary filesets at the same recovery point. The RPO feature can be disabled eliminating the creation of peer snapshots. AFM DR also provides a failback mechanism allowing to recover the old primary system and copy all new and changed data back from the new primary (secondary).

Note: The usage of UNIX HSM in combination with AFM DR is supported when the AFM DR RPO feature is disabled (AFM fileset parameter *afmRPO=disabled*). Furthermore, it is not supported to create normal snapshots in a space managed file system.

AFM modes of cache filesets running in SW, IW, or RO can be changed to all other AFM modes. Filesets running in LU-mode can't be changed to other modes. In all cases AFM can be disabled on filesets and make them a pure independent fileset. More AFM modes may be available in the future.

1.1.2 AFM File States

A file in the AFM cache can have different states as shown in Figure 1. Please note that the file states may be different depending on the AFM modes. Figure 1 shows the caching file states.

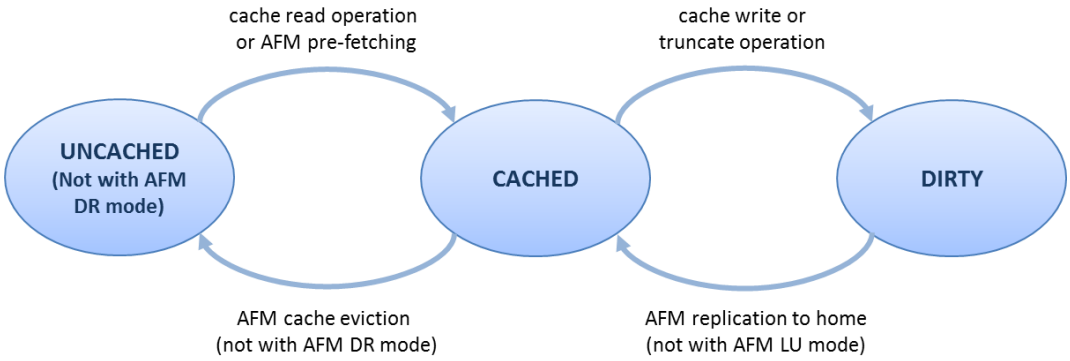

Figure 1: AFM file states at cache

UNCACHED: When an AFM relation is created between cache and home and there are files available in home, these files can be seen in cache without being present. This means that the file metadata is present in the cache, but the file content is still on home. Such files are in status uncached. In addition, the uncached status is achieved by evicting files from the cache. Note, with AFM DR mode files are never in uncached state on the primary.

CACHED: When an uncached file is accessed in cache for read or write it will be fetched from home. Fetching is the process of copying a file from home to cache. Files fetched

from home to cache are in status cached. Another way to fetch files from home to cache is by using the AFM prefetch command (mmafmctl prefetch). This command can leverage the policy engine to identify files quickly, according to certain criteria. Note, in AFM DR mode files on the primary are cached. They might be dirty as well if they have not been replicated.

DIRTY: When a cached file in the AFM cache is modified, that file is marked dirty, indicating that it is a candidate for replication back to home. The dirty status of the file is reset to cached, if the file has been replicated to home. When a file is deleted in cache, this delete operation will also be done in home. Modifying or deleting files in cache is not possible in AFM read-only mode. Note, in AFM LU mode dirty files are not replicated back to home, they become local. So the dirty state is also called local in AFM LU mode.

A cached file can be evicted from cache, which changes the file state to uncached and the file content is no longer stored in cache. This requires that the file is present in home and cached in cache. If a file is migrated in cache, it will not be evicted. Eviction is automatically controlled by AFM via the quota setting of the AFM cache fileset and not available in DR mode. In addition, the AFM evict command (mmafmctl evict) can be used.

1.2 IBM Storage Protect for Space Management

UNIX HSM provides tiering of files in an IBM Storage Scale file system from disk to an external pool. The external pool is represented by the IBM Storage Protect server, which can store files on tape.

In terms of UNIX HSM, a file can have three different states. A new or modified file has the state RESIDENT, which means that the file exists only in the file system. After the file was premigrated, it exists in a dual mode. One copy of the file resides in the file system while the other copy of the file is stored in the IBM Storage Protect. This file state is called PREMIGRATED. The third file state is called MIGRATED. This file state indicates that the file exists in the IBM Storage Protect server file space only. The copy of the file in the file system is replaced with a stub file, which typically doesn’t require any space in the data area of the file system.
HSM commands can be used to “premigrated” and “migrated” files. The environment might be configured to perform automatic threshold migration.

The following POSIX operations on files will cause UNIX HSM operations as described:
READ:

- A read access to a file requires the file data that will be read to be present, but doesn't change data. UNIX HSM provides different recall modes. All modes can be applied for the recall in terms of read access.
- No influence to RESIDENT files
- No influence to PREMIGRATED files
- File state MIGRATED change to PREMIGRATED. A recall is initiated because UNIX HSM allows multiple configurations for migrated files exceptions to this rule exist. See the UNIX HSM User Guide for more information.

WRITE:

- A write changes the content of the file. Therefore, the file content must be available at the time the write can be done. All write access to files requires a full recall of the file data. The recall will happen in normal recall mode. Different recall mode settings for the file will be disabled.
- No influence to RESIDENT files from HSM point of view
- File state PREMIGRATED change to RESIDENT. No recall.
- File state MIGRATED change to RESIDENT. A recall is initiated. (Because UNIX HSM allows multiple configurations for migrated files exceptions to this rule exist. See the UNIX HSM User Guide for more information)

TRUNCATE:

- A truncate of a file is comparable with a write where the new content of the file reduces the file size. Truncate cuts portions of the file at the end of the file. It is possible to truncate a file to size null. All truncate access to files requires a full recall of the file data, except truncate to null. The recall will happen in normal recall mode. Different recall mode settings for the file will be disabled.
- Same behaviors as described for case "WRITE." The only exception is truncate to size null. In this case, the file state will be changed to RESIDENT in all cases, but no recall is initiated at all.

Metadata changes or the rename of a file have no influence to the file migration state. The move of a file has no influence to the file migration state if the move doesn't cross IBM Storage Scale fileset boarders or IBM Storage Scale file system boarders. In term of

crossing the boarders described above, a recall is initiated and the file will be set to RESIDENT state.

AFM does whole-file caching by default. In general, reading more than three blocks of the file will cause AFM to cache the full file in the background for performance. To avoid reading large files when an application might be "peeking" into the file (for example, GUI viewers that read a few bytes to detect file mime type), certain file reads at the beginning of the file are handled individually. UNIX HSM provides the function to define different stub sizes for files. This means that a defined number of blocks will reside on disk while the rest of the data is migrated to the IBM Storage Protect server. To synchronize the AFM function to initiate the caching after reading three blocks of the file, the stub size for UNIX HSM migration should be set to the size of three blocks as well. This will reduce the amount of recalls that happen on home significantly.

Sequential read requests to an uncached file initiated from an application running on the cache cluster are pipelined. This means data blocks read sequentially from the application are directly passed from the home cluster to the application (via the AFM gateway node). This interlocks well with the UNIX HSM streaming recall function that essentially does the same on an HSM file system level. In cases where applications read the data in streams on the cache the UNIX HSM function streaming recall should be enabled for these files in the home cluster. Because a file in streaming recall mode can be read immediately after the recall has started, the combination of AFM streaming and UNIX HSM streaming recall will increase the overall read performance for uncached files in the cache.

Tip: UNIX HSM operates on file system level. In an AFM environment where a file system contains both cache and home fileset, consider using IBM Storage Protect EXCLUDE rules to exclude filesets from UNIX HSM processing. Follow this guideline especially for filesets where HSM is not applicable or required to prevent unexpected recall or migration behavior.

1.2.1 Important Environment Settings

To enable AFM support, the IBM Storage Protect processing option AFMSKIPUNCACHEDFILES YES must be set in the configuration file dsm.sys above all server stanza entries.

1.3 General Recommendations

Follow these general recommendations when using IBM Storage Scale AFM and UNIX HSM:

- Prevent cache eviction in combination with UNIX HSM on the same fileset. Both techniques have the same goal to reduce the space required in the fileset. The combination of both techniques unnecessary increase the complexity of the environment.

- IBM Storage Scale snapshots and UNIX HSM are not supported. This is because the deletion of a stub file that is reflected in a snapshot (the snapshot was generated before or after the file was migrated) will cause the recall of the file data. The file data will be stored in the snapshot so that it can be accessed later. Therefore, it is recommended to not use snapshots for an AFM fileset (in home or cache) and in the file system hosting the AFM fileset.
  The AFM DR RPO feature uses peer snapshots that create snapshots in the primary and the secondary AFM fileset to accommodate RPO. The usage of UNIX HSM in combination with AFM DR is supported when AFM DR RPO is disabled. When AFM DR RPO is disabled, then peer snapshots are not created.

- When using UNIX HSM on home or cache be aware that access (read or write) to multiple migrated files at the same time causes bulk recalls. Access to multiple files can be caused by users such as when they copy an entire directory or by AFM when changed files are replicated to home where the previous versions are migrated. You can avoid these issues by using the tape optimized recall process, which requires a list of files to be recalled before processing.

When running IBM Storage Scale AFM and IBM Storage Protect backup operations, prevent cache eviction in combination with IBM Storage Protect backup on the same fileset, if possible. Evicted (uncached) files will be skipped from backup processing. This can lead to errors in terms of the versioning of files on the IBM Storage Protect server.

1.4 General Processing Examples

Beginning with IBM Storage Protect version 7.1.3, the IBM Storage Protect for Space Management client and the backup-archive client skip files from processing that are AFM state dirty or AFM state uncached. IBM Storage Scale eviction will skip a file migrated with UNIX HSM. Use the following general processing guidelines for using the command-line interfaces from IBM Storage Protect and IBM Storage Scale for these cases:

The example below shows the messages that are issued if IBM Storage Protect for Space Management migration is performed on an uncached or dirty file:


```
> dsmmigrate file1
IBM Tivoli Storage Manager
Command Line Space Management Client Interface
  Client Version 7, Release 1, Level 3.0
  Client date/time: 07/20/15   10:56:47
(c) Copyright by IBM Corporation and other(s) 1990, 2015. All Rights
Reserved.

ANS9828W File: /gpfs1/file1 is skipped for migration: the file is either
AFM uncached or dirty.
```

The example below shows the messages that are issued if IBM Storage Protect backup is performed on an uncached or dirty file:

```
> dsmc incremental /gpfs1/file1
IBM Tivoli Storage Manager
Command Line Backup-Archive Client Interface
  Client Version 7, Release 1, Level 3.0
  Client date/time: 07/20/15   10:58:44
(c) Copyright by IBM Corporation and other(s) 1990, 2015. All Rights
Reserved.
. . .
Incremental backup of volume '/gpfs1/file1'
Normal File--> 15,539 /gpfs1/file1 [Skipped]

Total number of objects inspected:            1
Total number of objects backed up:            0
. . .
Total number of bytes transferred:            0  B
. . .
```

The example below shows the messages that are issued if IBM Storage Scale eviction is performed on a migrated file:

```
> mmafmctl gpfs1 evict -j sw2 --filter FILENAME=file1 --safe-limit=1
mmafmctl: [N] Unable to find any file to evict.
mmafmctl: Could not evict any of selected files.
```

Avoid evicting migrated files. While you can evict a file that is migrated but has resident blocks (this type of file might have a defined stub size or might be a file that is in “partial recall” mode and was partially accessed), in terms of an access to a migrated and evicted file, the recall operation from the IBM Storage Protect server takes place before the file data is fetched from AFM home.

2 User Scenario: Disaster Recovery

In this scenario, the cache fileset represents the primary site and is called primary fileset. While the primary fileset is available it allows read-write access to all files. The home fileset is the secondary site and it is called the secondary fileset. The secondary fileset obtains copies of all files asynchronously (see figure 2). In the case of a disaster at the primary site, the secondary site can be activated and become the new primary site using the failover process.

One or more primary filesets can be connected to a single secondary cluster whereby each primary fileset replicates to a distinct secondary fileset. For simplicity, we consider one primary fileset connected to one secondary fileset.

The primary is configured in PRIMARY mode; secondary is configured in SECONDARY mode and should be treated as read-only. Read-only mode on secondary is technically not enforced, it must be assured from an operational perspective because files create on the secondary are not replicated to the primary during normal operations. Files are managed (created, modified, read, and deleted) on the primary and asynchronously replicated to secondary, leveraging the AFM function. If the primary fails, secondary can be made the new primary lifting the read-only mode (failover). When the old primary is back online, it can be made primary again (failback). The normal failback process copies all changed files from the new primary to the old primary before the old primary becomes primary again.

In AFM DR Mode, there is no concept of uncached files in the primary, because the secondary is read-only. If a file on the primary has been changed, it becomes dirty and is replicated to the secondary during the next replication cycle. This is also true during failback, all files from the new primary are copied to the old primary and appear here in cached state.

AFM DR has a feature to assure a defined Recovery Point Objective (RPO) during failover and failback. The RPO feature is based on peer snapshots. A peer snapshot is initiated on the primary, takes a snapshot of the primary fileset and adds a snapshot marker to the replication queue. The replication queue contains operations that were executed on the primary fileset in order of their occurrence. These operations are sent to the secondary. When the snapshot marker arrives in the secondary it also takes a snapshot. Because the replication is in order, both snapshots have the same content, even though they have

been taken at distinct points in time. Peer snapshots allow to recover the file inventory at the secondary for a defined point in time, when the peer snapshot was taken.

The AFM DR RPO feature can be disabled (by setting primary fileset parameter afmRPO=disable). When the AFM DR RPO feature is disabled, then not peer snapshots are taken.

Note: The usage of UNIX HSM in combination with AFM DR is only supported when the RPO feature is disabled.
Background: if a file that is captured in a Storage Scale snapshots is in state migrated the deletion of this file causes a transparent recall into the snapshot. When many files are deleted this may cause recall storms in the primary. In addition, because the deletion is propagated to the secondary, it causes recall storms for migrated files in the secondary as well.
In addition, it is not supported to create normal Storage Scale snapshots in a space managed file system. See section General Recommendations to understand the limitations with UNIX HSM and IBM Storage Scale snapshots.

The subsequent sections elaborate on this user scenario and consider different cases with backup on secondary and primary.

2.1 Backup On Secondary And Primary

In this section, we elaborate on the AFM disaster-recovery use case with backup installed and configured on primary and secondary, as shown in Figure 2.

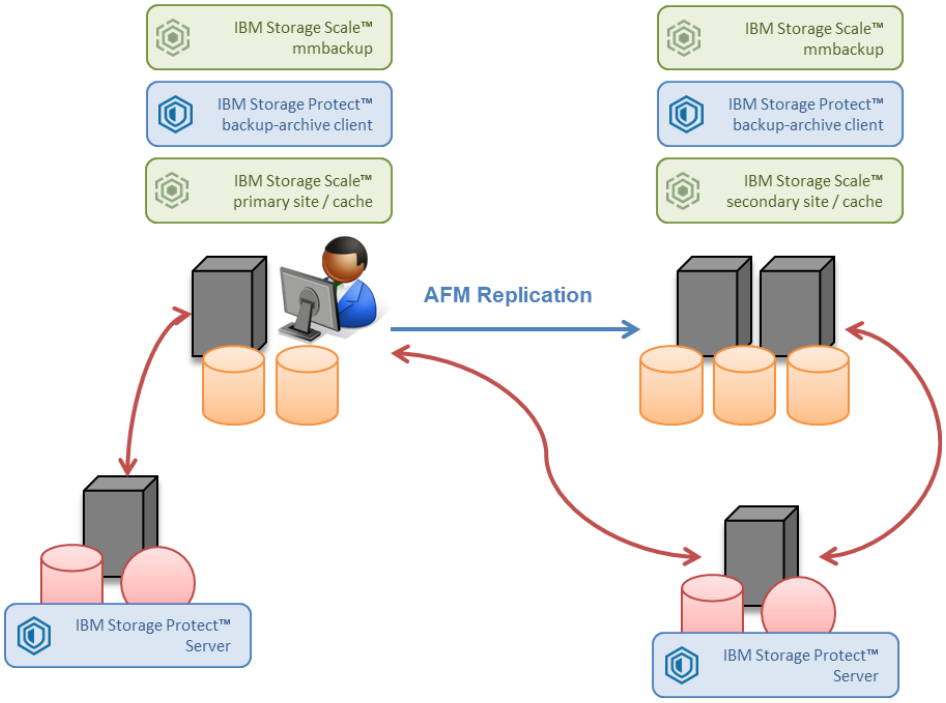

Figure 2: Disaster Recovery use case with backup on secondary and primary

The backup in IBM Storage Scale is accommodated with the mmbackup utility that uses the IBM Storage Protect backup-archive client. Backup of secondary and primary can be configured with the same or distinct Storage Protect servers. When using, the same Storage Protect server for backing up data from secondary and primary, the file system names in secondary and in primary must be distinct. This is required to prevent concurrent access to the same data from both AFM sites primary and secondary, which would lead to inconsistency in the backup.

The backup utility will not backup files in the primary are in state dirty[3].

2.1.1 Primary Operations

The following table summarizes the file operations on primary and consequences to secondary and primary. From an HSM perspective, the behavior is identical to the use case with HSM on secondary and primary (see section HSM on secondary and primary); we will not repeat this. Instead, we focus on the backup.

Operation	Primary	Secondary

[3] Storage Protect Backup Archive Client version 7.1.3 and later does skip uncached and dirty files from backup.

Operation	Primary	Secondary
CREATE file	Next backup will copy the file to the IBM Storage Protect server on primary. If the next backup cycle happens before the replication, the file will be skipped because it is dirty.	After replication, the next backup will copy the file to the IBM Storage Protect server on secondary.
READ file	No impact to backup processing.	No impact to backup processing.
WRITE file	After the write operation, the modified file will be replicated back to secondary asynchronously. Next backup will generate new version of the file in the IBM Storage Protect server on primary. If the next backup cycle happens before the replication, the file will be skipped because it is dirty.	After the write operation, the file will be replicated to secondary. Next backup will copy the file to the IBM Storage Protect server on secondary.
TRUNCATE file	Next backup will generate new version of the file in the IBM Storage Protect server on primary. If the next backup cycle happens before the replication, the file will be skipped because it is dirty.	Truncate operation will be promoted to secondary without transferring the file during next replication cycle. Next backup will generate new version of the file in the IBM Storage Protect server on secondary.
RENAME or MOVE file within fileset	Next backup will create a new file in the IBM Storage Protect server and expire the old file (name).	Move operation will be promoted to secondary without transferring the file during the next replication cycle. Next backup will generate new file in the IBM Storage Protect server and expire the old file (name)
MOVE file to other fileset	Next backup will expire the moved file in the IBM Storage Protect server on primary.	The next replication cycle will remove the file from the secondary fileset.

Operation	Primary	Secondary
		Next backup will expire the file in the IBM Storage Protect server on secondary
EVICTION	N/A, eviction is disabled with AFM DR	No Impact.
DELETE file	Next backup will have expired the file in the IBM Storage Protect server on primary	The next replication cycle will delete the file from the secondary fileset. Next backup will expire the file in the IBM Storage Protect server on secondary
PEER SNAPSHOTS	No impact, not supported with UNIX HSM	No impact, not supported with UNIX HSM
FAILOVER	Because primary fileset might not be available, no files are backed up.	No Impact
FAILBACK	If backup is performed during failover this may cause files to be skipped because they are dirty. Next backup cycle will backup these files.	No Impact

2.1.2 Challenges

Challenges with backup in the disaster-recovery user scenario:

- Backing up files on primary before replication will cause these files to be skipped because changed files that are not replicated are dirty[4]. This is particularly important to consider if the primary is disconnected from the secondary for a longer period.

- The backup operation on primary might interfere with the asynchronous replication causing race conditions on files.

[4] Storage Protect Backup Archive Client version 7.1.3 and later does skip uncached and dirty files from backup.

- A backup operation on primary and secondary might cause recalls of migrated files that have not been backed up. This might result in massive recalls delaying the backup operation. In addition, this might cause unwanted growth of the occupation of the secondary and primary file systems.

- If the primary fileset is down (failover condition) the backup operation on primary file system will skip the files in the primary fileset

- During failback files from the secondary are copied to the primary, either in-band using the apply-updates process or out-of-band using copy utilities. The backup operation on the primary may interfere with these processes, causing race conditions and skipped files.

- All operations that lead to new backup versions of files in the primary fileset will be asynchronously promoted to secondary and will cause the same backup operation at secondary. Because the secondary file system can contain several secondary filesets exported to several primaries, the amount of changed and newly created files can be high. It must be assured that the IBM Storage Protect backup environment that is used to protect the files in the secondary file system is configured to achieve the expected backup performance and scalability.

2.1.3 Recommendations

Recommendations for backup in the disaster-recovery user scenario:

- Configure the Backup client option AFMSKIPUNCACHEDFILES YES.

- Plan the backup schedules on primary and secondary to have least interference with the replication operation. AFM replication is done automatically in the background from primary to secondary per the asynchronous delay configuration parameter (afmAsyncDelay). If files change on primary, the files will be replicated to secondary. There are two considerations for scheduling backup on the primary and secondary:
    - Run backup when files on primary are rarely changed or created, for example, during the night.
    - Delay asynchronous replication before starting the backup operation adjusting the AFM parameter afmAsyncDelay on the primary fileset before and after the backup.

- If the primary is disconnected from the secondary it is recommended to suspend the backup operation on primary to prevent skipping to many dirty files. If the primary is reconnected to the secondary let the replication finish before resuming the backup operation.

- When using the same IBM Storage Protect server as destination for backup from the primary and secondary it is important that the file system names in the primary and secondary are distinct. Otherwise, all files from primary and secondary are stored in the same Storage Protect server file space causing confusion. Furthermore, when using the same Storage Protect server provision different node names for the primary and secondary system (proxy and target nodes).

- During failback suspend the backup operation on the primary until files have been copied from the secondary to the primary to avoid race conditions.

- To cope with the backup loads at the secondary site, certain filesets on the secondary file system could be excluded from backup by using exclude lists or perform fileset level backups.[5]

[5] Storage Scale mmbackup version 4.1.1 and later allow backup to single filesets. You can use this option for AFM home filesets to apply the same backup policy on both side's home and cache.

3 User Scenario: Branch Office Caching

In this scenario, one or more caches in branch offices work with files centrally stored in a home site (see figure 3). Home is the source for files and acts as a central location where files are stored, managed, and protected. Files in home can be seen in the caches without being present. Upon access, the file will be transferred from home to cache. A controllable alternative way for transferring files is by pre-fetching.

In this scenario, a cache is typically configured in SW or IW mode, rarely in RO mode. In SW or IW mode, the users at the cache site can create, modify, read, and delete files that are asynchronously replicated back to home. In IW mode, multiple caches can see and work with the same files exported by a single home. Care must be taken because there is no cross-cache locking. In RO mode, the cache can see and read all files stored at home, but files cannot be added, modified, or deleted at the cache site.

In the branch office scenario, it might be common that the caches are represented by smaller system providing less storage capacity. To cope with the data growth at the cache site, eviction might be used. Eviction files that are synced with home are stubbed in the cache, consuming almost no storage capacity. Eviction is triggered by the cache fileset quota and can be changed to evict files based on size or LRU.

The subsequent sections elaborate on this user scenario and consider different cases with HSM on home, HSM on home and cache and backup on home and cache.

3.1 HSM On Home

In this section, we elaborate on the branch-office use case with HSM installed and configured on home, as shown in Figure 3.

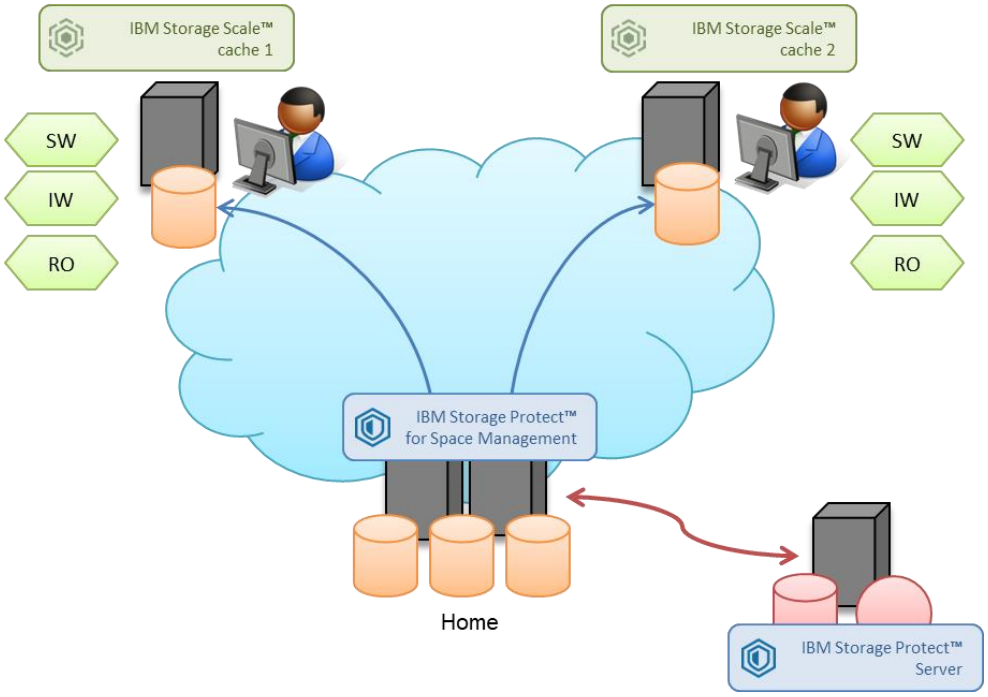

Figure 3: Branch office setup with HSM on home

AFM home is an IBM Storage Scale fileset contained in an IBM Storage Scale file system managed from UNIX HSM. The AFM cache fileset is configured in SW, IW, or RO mode and connects to the home fileset via NFS. In SW and RO mode, home exports a fileset for each cache, so there might be multiple filesets in home. In IW mode, there might be one fileset at home that is used by all caches.

Data stored in the home export can be seen in the cache fileset. Data created or modified in the cache fileset is replicated back to home asynchronously (SW or IW mode only). Pre-fetching might only be required if home has data before establishing the AFM relation to make this available to the caches.

UNIX HSM on home is configured to manage the file system that contains one or more home filesets for one or more caches. Files might be created on home however, care must be taken when modifying and deleting existing files on home, since this may conflict with modifications of the same files on cache. In addition, files will be replicated from cache to the corresponding home fileset.

3.1.1 Cache Operations

The following table summarizes the file operations on cache and consequences to home and cache regarding UNIX HSM.

Operation	Cache	Home
CREATE file	After the creation, the file will be replicated to the home fileset asynchronously.	After replication, the file becomes a candidate for migration in the home fileset.
READ file	If the file is uncached, it needs to be fetched from home. This may also occur after eviction.	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are read at the same time (entire directory) this may cause massive recalls on home.
WRITE file	If the file is uncached, it needs to be fetched from home. This might also occur after eviction. After write, the file will be replicated to the home fileset asynchronously. If the replication causes recalls on home, then the asynchronous replication might stall.	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are written at the same time this might cause massive recalls on home. After the write operation, during replication, if the file is migrated on home this operation will recall the file. The following write operation on home will change the file state to resident. When multiple files replicated from cache are in migrated state on home, this might cause massive recalls.
TRUNCATE file	No Impact, truncate operation on uncached file will not cause fetching. After truncate file size is adjusted in home during asynchronous replication (no data is replicated).	After truncate, during replication, if the file is not truncated to a size of 0 and it is migrated on home, this will cause recall on home. The following truncate operation on home will change the file state to resident. When multiple

Operation	Cache	Home
		truncated files replicated from cache are in migrated state on home, this might cause massive recalls.
RENAME or MOVE file within fileset	No Impact, uncached files will not be fetched.	No Impact, subsequent replication will apply move operation in home. UNIX HSM reconciliation will update the new name for migrated files.
MOVE file to other fileset	If the file is in uncached state, it will be fetched from home. Subsequent replication will remove the file from home.	If file needs to be fetched from home and is migrated on home, then the file is recalled during the fetching operation. When multiple files have been moved on cache (for example, a directory) and are migrated on home this will cause massive recalls. Subsequent replication will remove the files from the original location in home.
EVICTION	File will be set to AFM uncached in the cache fileset.	No Impact.
DELETE file	No Impact, subsequent asynchronous replication will delete file in home, no data is transferred.	No Impact, file will be deleted on home on next replication.

3.1.2 Challenges

- File modification in cache that requires the replication to the home can lead to the recall of the file on home in case it was migrated. Because many files may be replicated to home in each asynchronous replication cycle, massive recalls can occur on home. This can cause unwanted delays of the asynchronous replication. It might also cause the occupancy of the home fileset to grow unexpected.

- Evicted files on cache will be fetched from home on access. If the file was migrated on home, this can lead to the recall of the file on home in case it was migrated. If multiple evicted files are accessed at the same time, massive recalls can occur on home. This can cause unwanted delays of the file operation on cache. It might also cause the occupancy of the home fileset to grow unexpected.

- The recall behavior of UNIX HSM is significantly influenced by the storage medium where the migrated data is stored in the IBM Storage Protect server. If the data is stored in tape pools, the recall of each file might take longer depending on the used tape technology (2-5 minutes).

3.1.3 Recommendations

- Files that will be changed frequently in the cache fileset and require frequent replications to the home fileset must be excluded from the UNIX HSM migration to the IBM Storage Protect server to prevent unwanted recalls. The access time stamp of files can be used to identify and exclude those files in the migration policy.

- The policy driven threshold migration used to reduce the overly space needed for data in the file system by applying UNIX HSM migration functions must be configured to ensure that enough candidates can be found to free up space for the recalls required for the replication of changed data.

- The sizing of the home storage capacity should take into account recalls that might happen for different reasons explained above. Recalls cause the storage occupancy in the home fileset to grow and can cause race conditions with threshold migrations. If several cache filesets replicate data to home, that increases the storage capacity needs.

- When using eviction on cache it must be understood that access to files is promoted to home where files might be migrated. This might cause more recalls and overload the home system. Therefore, eviction should either be disabled or coordinated with migration on home, for example, by using the age of files.

- If after a catastrophic failure, the cache has to be rebuild, the policy engine can be used to identify all files migrated on home and the UNIX HSM tape optimized

recall function can be used to recall all these files prior to the pre-populating or pre-fetching them to cache.

3.2 HSM On Home And Cache

In this section, we elaborate on the branch-office use case with HSM installed and configured on home and on cache, as shown in Figure 4.

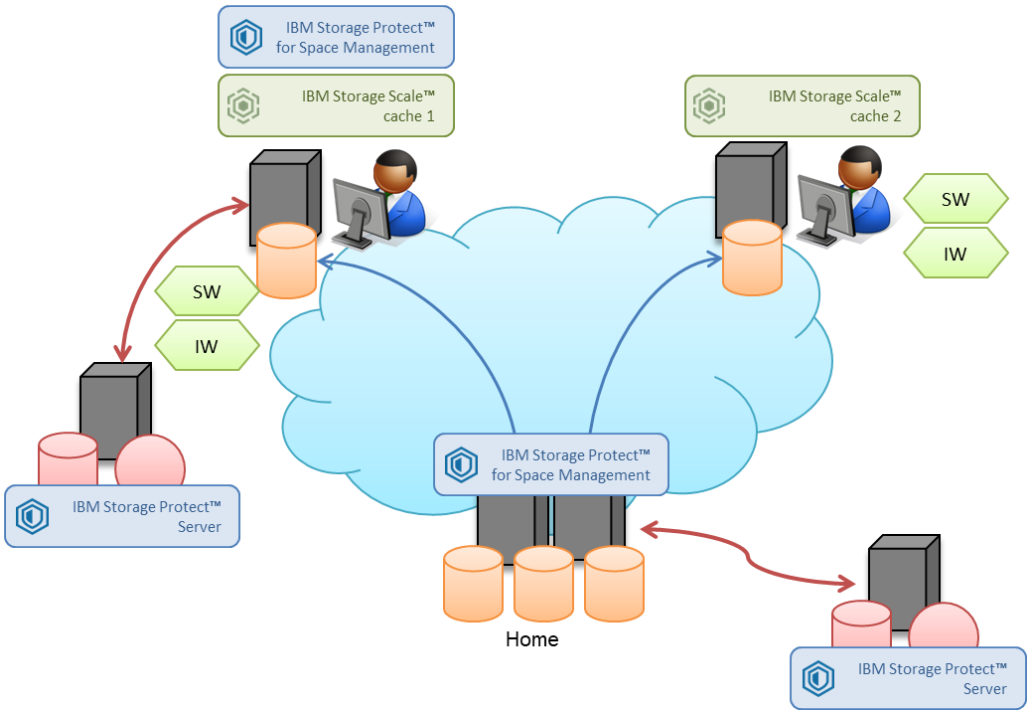

Figure 4: Branch office use case with HSM on home and cache

AFM home is an IBM Storage Scale fileset contained in an IBM Storage Scale file system managed from UNIX HSM. The AFM cache fileset is managed by another instance of UNIX HSM that works independently from the instance configured on the home fileset (and other cache filesets). The cache fileset can be configured in SW or IW mode. Configuring the cache in RO mode does not make sense in combination with HSM on the cache site. The AFM cache fileset connects to the home fileset via NFS. In SW mode, home exports a fileset for each cache, so there might be multiple filesets in home. In IW mode, there might be one fileset at home that is used by all caches.

Data stored in the home export can be seen in the cache fileset. Data created or modified in the cache fileset is replicated back to home asynchronously. Pre-fetching might only be

required if home has data before establishing the AFM relation, to make this available to the caches.

One instance of UNIX HSM is configured to manage one or more home filesets for one or more caches. Files might be created on home however, care must be taken when modifying and deleting existing files on home, since this may conflict with modifications of the same files on cache. In addition, files will be replicated from cache to the corresponding home fileset.

3.2.1 Cache Operations

The following table summarizes the file operations on cache and consequences to home and cache in regard to UNIX HSM.

Operation	Cache	Home
CREATE file	After the creation, the file will be replicated to the home fileset asynchronously.	After replication, the file becomes a candidate for migration in the home fileset.
READ file	If the file is migrated on cache, it will be recalled. When multiple files (entire directory) are read, this might cause massive recalls. If the file is uncached in cache (after eviction), it needs to be fetched from home.	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are read at the same time (entire directory) this might cause massive recalls on home.
WRITE file	If the file is migrated on cache, it will be recalled. If the file is uncached in cache (after eviction), it needs to be fetched from home. After the write operation, the file is marked dirty. The subsequent replication to home will clear this status. Eventually the file will become	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. After the write operation, during replication, if the file is migrated on home this operation will recall the file. The following write operation on home will change the file state

Operation	Cache	Home
	a candidate for migration in cache[6]. If the replication causes recalls on home, then the asynchronous replication may stall.	to resident. When multiple files replicated from cache are in migrated state on home, this might cause massive recalls.
TRUNCATE file	If the file is migrated on cache and the file is NOT truncated to a size of 0 it will be recalled. If the file is uncached, the truncate operation will NOT cause fetching. After truncate file size is adjusted in home during asynchronous replication (no data is replicated).	After truncate, during replication, if the file is not truncated to a size of 0 and it is migrated on home, this will cause recall on home. The following truncate operation on home will change the file state to resident. When multiple truncated files replicated from cache are in migrated state on home, this might cause massive recalls.
RENAME or MOVE file within fileset	No Impact. UNIX HSM reconciliation will update the new name for migrated files.	After replication from cache, UNIX HSM reconciliation will update the new name for migrated files.
MOVE file to other fileset	If the file is migrated, it will be recalled. When moving multiple files (entire directory), this will cause massive recalls. If the file is in uncached state, it will be fetched from home. Subsequent replication will remove the file from home.	If file needs to be fetched from home and is migrated on home, then the file is recalled during the fetching operation. When multiple files have been moved on cache (entire directory) and are migrated on home this will cause massive recalls.
EVICTION	If the file is migrated on cache, it will not be affected. After eviction AFM file state, will be set to uncached. There	No Impact.

[6] Storage Protect for Space Management version 7.1.3 and later does skip dirty files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.

Operation	Cache	Home
	is no impact to HSM processing[7].	
DELETE file	No Impact, subsequent reconciliation will expire the file at the IBM Storage Protect server. Asynchronous replication will delete file in home; no data is transferred.	No Impact, file will be deleted on home on next replication. UNIX HSM reconciliation will expire the file at the IBM Storage Protect server.

3.2.2 Challenges

Generally, the same challenges as with UNIX HSM on home. Above that, there are more challenges associated with UNIX HSM configured on cache.

- Migration of uncached and dirty files will generate warning messages. Uncached and dirty files won't be migrated[8].

- In general, eviction of files in the cache is not supported in combination with UNIX HSM. The usage of eviction might lead to unexpected behavior in terms of file operations like read, write, truncate, or move. Eviction of migrated files on cache will generate warning messages and those files are not evicted. Only whole files are being evicted.

- File operations like read, write, truncate or move on migrated files on cache will cause recalls on cache. If file operations causing recalls are done for multiple files, massive recalls can occur on cache. This might cause file operations to be delayed and the occupancy of the cache fileset to grow unexpected.

- File write operations on cache might lead to recalls on both sites cache and home. If the file is migrated on cache, it will be recalled on cache. If the file is migrated on home, the next replication will cause a recall on home. This might cause file operations, replication to be delayed, and the occupancy of the file systems on cache and home to grow unexpected.

[7] Storage Protect for Space Management version 7.1.3 and later does skip uncached files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.
[8] Storage Protect for Space Management version 7.1.3 and later does skip uncached files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.

- The recall behavior of UNIX HSM is significantly influenced by the storage medium where the migrated data is stored in the IBM Storage Protect server. If the data is stored in tape pools, the recall of each file may take longer depending on the used tape technology (2-5 minutes).

3.2.3 Recommendations

Generally, the same recommendations as described above for UNIX HSM activity on home. In addition to that, there are recommendations associated with UNIX HSM configured on cache.

- Cache eviction must be disabled on the cache fileset for this scenario, because it may conflict with UNIX HSM.

- To prevent migrating of uncached files[9] the migration policy should omit uncached files. The following example defines uncached files that could be used to prevent the migration of uncached files.

```
/* define macro to identify uncached files */

define( is_uncached,
( NOT RegEx(misc_attributes,['[DXazu]'])
  AND NOT RegEx(mode,['[bcps]']) ) )
```

- A file is marked dirty if it has content changes on the cache fileset site that were not yet replicated to home. To prevent migrating files on cache that have not been replicated the migration policy applied to the caches fileset must omit dirty files[10]. Find below an example defining dirty files:

```
/* define macro to identify dirty files */

define( is_dirty,
( NOT RegEx(misc_attributes,['[DXaz]'])
  AND RegEx(misc_attributes,['[w]'])
```

[9] Storage Protect for Space Management version 7.1.3 and later does skip uncached files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.
[10] Storage Protect for Space Management version 7.1.3 and later does skip dirty files from migration. Nevertheless, the used migration policy should sort out dirty files to reduce the processing overhead.

```
  AND NOT RegEx(mode,['[bcps]']) ) )
```

- Files that will be changed frequently in the cache fileset should not be migrated. The policy engine can be configured to identify files that do not change any further, for example, by using the access time stamp. The following example shows the command used to migrate files accessed 5 days ago, or longer:

```
/* define macro for modification time */

define( acc_age,
( DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) ) )

/* migration based on modification age*/

RULE 'modAge' MIGRATE
 FROM POOL 'system'
 TO POOL 'silver'
  WHERE (acc_age > 5)
```

- Files that will be changed frequently in the cache fileset and require frequent replications to the home fileset should not be migrated on home to prevent unwanted recalls. The access time stamp of files can be used to identify and exclude those files in the migration policy (see example above).

- The threshold migration must be configured on home and cache to ensure that sufficient candidates can be found to free up space for the recalls required for the replication of changed data.

- The sizing of the home and cache storage capacity should consider recalls that might happen for different reasons explained above. Recalls cause the storage occupancy in the home fileset to grow and can cause race conditions with threshold migrations.

- If after a catastrophic failure, the cache has to be rebuild from home, the policy engine can be used to identify all files migrated on home that belong to the failed cache and the UNIX HSM tape optimized recall function can be used to recall all these files prior the pre-fetching them to cache. The home fileset must be excluded from threshold migration until all files have been pre-fetched.

3.3 Backup on Home and Cache

In this section, we elaborate on the branch-office use case with HSM and backup installed and configured on home and on cache, as shown in Figure 5.

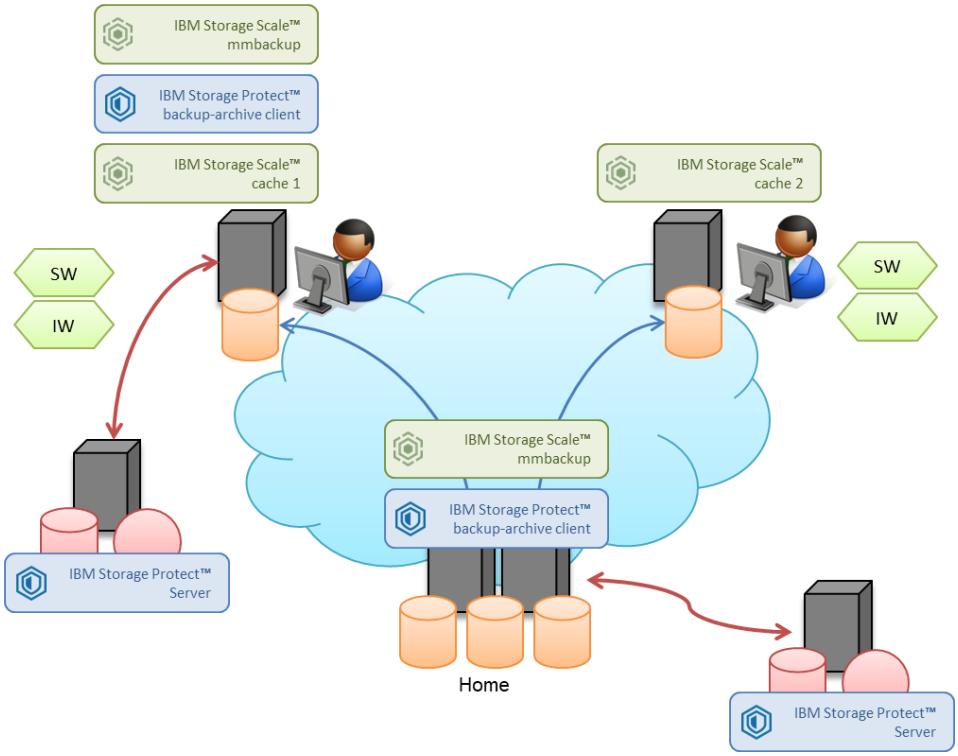

Figure 5: Branch office use case with backup and HSM on home and cache

The setup of AFM solution corresponds to the setup described in section HSM on home. In addition to UNIX HSM being installed and configured on the file systems hosting the cache filesets and on the home filesets, managing file migration and recall independently between cache and home, the backup of these file systems is enable.

The backup in IBM Storage Scale is accommodated with the mmbackup tool that uses the IBM Storage Protect backup client. Back up of home and cache can be configured independent. The IBM Storage Protect server file space must be different for the home fileset and the cache fileset. To prevent concurrent access to the same data from both AFM sites cache and home. This would lead to inconsistency in the backup.

3.3.1 Cache Operations

The following table summarizes the file operations on cache and consequences to home and cache. From an HSM perspective, the behavior is identical to the use case with HSM on home and cache; we will not repeat this. Instead, we focus on the backup.

The following table summarizes the file operations on cache and consequences to home and cache. From an HSM perspective, the behavior is identical to the use case with HSM on home and cache; we will not repeat this. Instead, we focus on the backup.

Operation	Cache	Home
CREATE file	Next backup will generate a backup copy in the IBM Storage Protect server.	After replication, the next backup will copy the file to the IBM Storage Protect server.
READ file	No impact to backup processing.	No impact to backup processing.
WRITE file	After the write operation, the modified file will be replicated back to home asynchronously. Next backup will generate new version of the file in the IBM Storage Protect server on cache.	After the write operation, the file will be replicated back to home. Next backup will generate new version of the file in the IBM Storage Protect server on home.
TRUNCATE file	Next backup will generate new version of the file in the IBM Storage Protect server.	Truncate operation will be promoted to home without transferring the file during next replication cycle. Next backup will generate new version of the file in the IBM Storage Protect server.
RENAME or MOVE file within fileset	Next backup will generate a new file in the IBM Storage Protect server and expire the old file (name).	Move operation will be promoted to home without transferring the file during the next replication cycle. Next backup will generate new file in the IBM Storage Protect server and expire the old file (name)
MOVE file to other fileset	Next backup will expire the moved file.	Next backup will expire the moved file.
EVICTION	Eviction should be disabled.	No Impact.

Operation	Cache	Home
DELETE file	Next backup the file will be expired in the IBM Storage Protect server.	During the replication after the delete operation, the file will be deleted in home. Next backup will expire the file in the IBM Storage Protect server.

3.3.2 Challenges

Generally, the same challenges as with HSM on home and cache apply here. However, there are more challenges associated with backup configured on cache and cache.

- Uncached and dirty files will not be backed up. This might cause inconsistency of the backed-up data in the cache if files that are uncached have not been backed up.

- The backup operation on cache and home might cause recalls of migrated files that have not been backed up. This might result in massive recalls delaying the backup operation. In addition, this might cause unwanted growth of the occupation of the home and cache file systems.

- The backup operation on cache might interfere with the asynchronous replication to home, causing missed backup files and race conditions.

- The replication operation to home might interfere with the backup operation on home, causing missed backup files and race conditions.

- All operations that lead to new backup versions of files in the cache fileset will be asynchronously promoted to home and will cause the same backup operation at home. Because the home file system can contain several home filesets that are exported to several caches, the amount of changed and newly created files can be high. Ensure that the IBM Storage Protect backup environment that is used to protect the files in the home file system is configured to achieve the expected backup performance and scalability.

- Backup of file that is evicted on cache causes the file to be replicated from home to cache. If the files are migrated on home this might cause massive recalls on home.

3.3.3 Recommendations

Generally, the same recommendation as with HSM on home and cache apply. In addition, that there are challenges associated with backup configured on cache and home.

- Do not use eviction and backup for the same fileset. If eviction is used on the cache site, ensure that backup happens before eviction. Because evicted (uncached) files will be skipped from backup, this situation can lead to inconsistencies in terms of backup versions in the IBM Storage Protect server.

- When using HSM and backup for the same IBM Storage Scale file system and fileset, use the same IBM Storage Protect server. This allows you to prevent files from being migrated before they have been backed up, using the management class parameter: MIGREQUIRESBkup=yes.

- Do not use the same instance of IBM Storage Protect server as the destination for UNIX HSM on cache and home. This defeats the idea of disaster protection because if this single IBM Storage Protect server instance fails, both copies of migrated files might get lost. However, when only using backup on cache and home, the same IBM Storage Protect server instance can be used. In this case, however, it is important that the file space of the AFM cache fileset is different than the home fileset. This can be archived by selecting a different name for the root directory of the fileset on cache and home.

- The backup schedules on home and cache should take into consideration the replication cycles. AFM replication is done automatically in the background from cache to home and if files change on cache, the files are replicated to home. There are two options to accommodate this:
    - Run backup when files on cache are rarely changed or created, for example during the night.
    - Delay asynchronous replication before starting the backup operation by adjusting the AFM parameter afmAsyncDelay on the cache fileset before and after the backup.

- When the cache must be re-created (after a failure), pre-fetch all files back from home to cache before starting the backup. This also assures that subsequent

backups do not cause fetching of files from home to cache and perhaps recalls on home.

- When pre-fetching files from home the policy engine can be used on home to identify all files that have been migrated. The UNIX HSM tape optimized recall method should be used to recall all these files on home before pre-fetching them to cache.

- To cope with the backup loads at the home site, certain filesets at the home file system could be excluded from backup using exclude lists or perform fileset level backups.

- Eviction is disabled for all cache filesets, even if UNIX HSM is not configured because backup of files on cache can cause uncached (evicted) files to be fetched from home. In addition, if these files are migrated on home, it can cause massive recalls and the backup operation to be delayed. In addition, this can cause unwanted growth of occupancy in the home file system.

4 User Scenario: System Migration

In this scenario, AFM replication is used to perform a system migration from a legacy file system to an IBM Storage Scale file system. The cache fileset is in an IBM Storage Scale file system. The corresponding home can be an IBM Storage Scale fileset or a different storage device that is accessible via NFS mount. Each cache fileset corresponds to one home fileset or one NFS-mounted storage device. The cache does all the work.

The cache fileset is typically configured in LU-mode. In LU-mode, files from home are visible in cache but not yet present (uncached state). Files are fetched from home on access or by pre-fetching. Any changes done to files on cache are not replicated back to home. Therefore, the AFM relation for each file is broken when the file is modified. Newly created files on cache are also not replicated from home.

System migration is done by connecting a cache in LU-mode to an NFS export provided by home. Files can be immediately being seen in cache without being present there. Thus, users could immediately start working on cache whereby uncached files are fetched from home on access. In most cases a subset of files, such as the most active ones, are copied from home to cache using AFM pre-fetching, before users and applications use the new system (cache). Once the required files are available on cache the users and application can use the cache while more files are fetched from home. If all files are fetched from home to cache the legacy filer (home) can be discontinued.

It might be common that a cache is represented by larger system providing more storage capacity. One or more cache filesets might be used to consolidate multiple legacy storage devices (each represented by NFS export) or legacy IBM Storage Scale file systems.

NOTE: Eviction shall not be used with the system-migration use case because it defeats the purpose of data migration.

4.1 HSM On Home

In this section, we elaborate on the system-migration use case with HSM installed and configured on home, as shown in Figure 6.

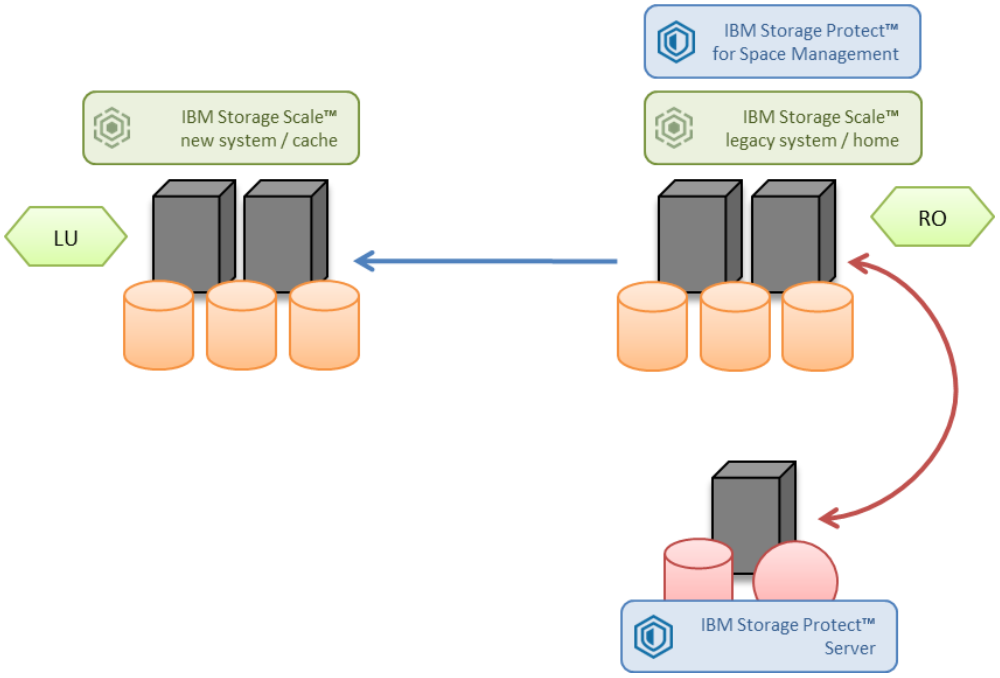

Figure 6: System migration setup with HSM on home

Home is an IBM Storage Scale fileset contained in an IBM Storage Scale file system that is managed from UNIX HSM. The AFM cache fileset is running in LU-mode. File content changes in the cache fileset are not being replicated to the home fileset.

Because UNIX HSM was used on the legacy system in the past, you can assume that many files that were in migration state were migrated and must be recalled for the system migration. Threshold migration should be enabled on home to prevent an out-of-space condition in terms of the expected file recalls for the system migration.

4.1.1 Cache Operations

Operation	Cache	Home
PRE-FETCHING	File data will be created in cache	Pre-fetching of a file will lead to a recall of the file in the home fileset if it was in migration migrated state. This will increase the threshold of the file system. No impact if the file is in migration state premigrated.

Operation	Cache	Home
CREATE file	File will be created in cache, but no replication to home will occur in LU mode.	No impact to HSM processing
READ file	If the file is in uncached state, it needs to be fetched from home.	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are read at the same time (entire directory) this might cause massive recalls on home.
WRITE file	If the file is uncached, it needs to be fetched from home.	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are written at the same time this might cause massive recalls on home.
TRUNCATE file	The file will be truncated to the new size.	No impact to HSM processing
RENAME or MOVE file within fileset	No Impact, uncached files won't be fetched.	No impact to HSM processing
MOVE file to other fileset	If the file is in uncached state, it will be fetched from home.	If file needs to be fetched from home and is migrated on home, then the file is recalled during the fetching operation. When multiple files have been moved on cache (directory) and are migrated on home this will cause massive recalls.
EVICTION	N/A	Eviction must be disabled
DELETE file	File will be deleted from cache.	No impact to HSM processing

4.1.2 Challenges

- After the AFM relation between cache and home has been established, files can be accessed in the cache fileset. Any access will lead to a fetching of the file data to the cache fileset. This will lead to random recalls of files in the home fileset, if files are migrated on home. Because no access optimization (ordering of recalls) will be applied by default for the recall of the files, a significant performance impact can be assumed.

- The recall behavior of UNIX HSM is significantly influenced by the storage medium where the migrated data is stored in the IBM Storage Protect server. If the data is stored in tape pools, the recall of each file may take longer depending on the used tape technology (2 - 5 minutes).

- Because the home fileset or legacy system uses the UNIX HSM function for a long time, it can be assumed that a huge number of files is in migration state migrated on the home fileset. Therefore, pre-fetching can lead to a mass recall of files on the home fileset.

4.1.3 Recommendations

- To prevent random mass recalls of migrated files caused by the AFM pre-fetching or user access, the following steps should be performed:

    - Generate the list of all migrated files on the legacy system (preferred processing on IBM Storage Scale: usage of the policy engine). Files in premigrated must not be listed. Those files can be read without recall.
    - Initiate a tape optimized recall processing in prepare mode to get the input file list separated in subfile lists ordered by media type and number. Use the created collection file to identify the space required for each media type or number to be recalled to the file system.
    - Initiate a tape optimized recall of a sub portion of all migrated files to file state premigrated. Ensure that the defined sub portion of files will fit into the free space available in the home fileset.
    - Use the same file list to initiate the pre-fetching of files to the cache fileset.
    - After the replication of the data has finished re-migrate the files on the legacy system (Since the files were recalled to premigrated state only the

re-migration is a local deletion process and doesn't require any data transfer)

- The sizing of the home storage capacity must take into account that massive number of files that are recalled. This might require that you temporarily add storage capacity to the home system. Alternatively, the recall and pre-fetch process must be executed in a way to prevent home from running full.

- The threshold migration must be configured to ensure that enough candidates can be found to free up space for the recalls required for the pre-fetching of file to the cache.

4.2 HSM On Home And Cache

In this section, we elaborate on the system-migration use case with HSM installed and configured on home and cache, as shown in Figure 7.

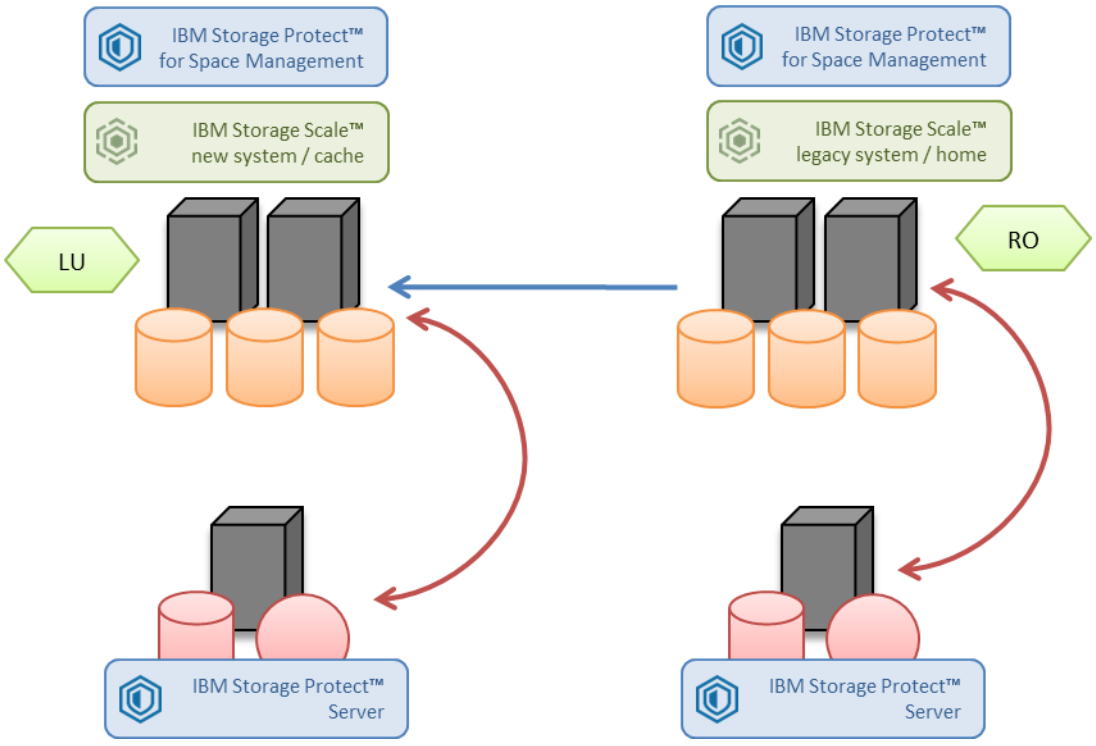

Figure 7: System migration setup with HSM on home

Home is an IBM Storage Scale fileset contained in an IBM Storage Scale file system that is managed from UNIX HSM. The AFM cache fileset is running in LU-mode. File content changes in the cache fileset are not being replicated to the home fileset.

Because UNIX HSM was used on the legacy system in the past, it can be assumed that many files that are in migration state were migrated and must be recalled for the system migration. Threshold migration will be enabled on home to prevent out of space condition in terms of the expected file recalls for the system migration.

4.2.1 Cache Operations

Operation	Cache	Home
PRE-FETCHING	File data will be created in cache	Pre-fetching of a file will lead to a recall of that file in the home fileset if it was in migration state migrated. This will increase the threshold of the file system. No impact if the file is in migration state premigrated.
CREATE file	File will be created in cache, but no replication to home will occur in LU mode. File becomes candidate for migration.	No impact to HSM processing
READ file	If the file is uncached in cache state, it needs to be fetched from home. Read operation might take a while if the file must be replicated from home and is migrated on home. When multiple files are read at the same time, this might cause recall-storms on home and time-outs of the read operations. If the file is migrated on cache, it	If file needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are read at the same time (entire directory) this might cause massive recalls on home.

Operation	Cache	Home
	will be recalled. When multiple files (entire directory) are read, this might cause massive recalls.	
WRITE file	If the file is uncached in cache, it needs to be fetched from home. After write, the file will become candidate for migration in cache. If the file is migrated on cache, it will be recalled.	If file in in uncached state in the cache and needs to be fetched and is migrated in home, then the file will be recalled during fetching operation. When multiple files are written at the same time this might cause massive recalls on home.
TRUNCATE file	If the file is migrated on cache and the file is NOT truncated to a size of 0 it will be recalled. If the file is uncached, the truncate operation will *not* cause fetching.	No impact to HSM processing
RENAME or MOVE file within fileset	No Impact, uncached files won't be fetched. UNIX HSM reconciliation will update the new name for migrated files.	No impact to HSM processing
MOVE file to other fileset	If the file is migrated, it will be recalled from the IBM Storage Protect server. When moving multiple files (entire directory), this will cause massive recalls. If the file is in uncached state, it will be fetched from home.	If file needs to be fetched from home and is migrated on home, then the file is recalled during the fetching operation. When multiple files have been moved on cache (directory) and are migrated on home this will cause massive recalls.
EVICTION	Eviction must be disabled	Eviction must be disabled
DELETE file	File will be deleted from cache. Subsequent UNIX HSM reconciliation will remove the file from IBM Storage Protect server.	No impact to HSM processing

Operation	Cache	Home
	No replication to home in LU mode.	

4.2.2 Challenges

Generally, the same challenges as with UNIX HSM on home (see HSM on home). In addition, there are more challenges associated with UNIX HSM configured on cache:

- Migration of uncached files will generate warning messages. Uncached files won't be migrated[11].
- File operations like read, write, truncate, or move on migrated files on cache will cause recalls on cache. If file operations causing recalls are done for multiple files, massive recalls can occur on cache. This might cause file operations to be delayed and the occupancy of the cache fileset to grow unexpected.
- The recall behavior of UNIX HSM is significantly influenced by the storage medium where the migrated data is stored in the IBM Storage Protect server. If the data is stored in tape pools, the recall of each file may take longer depending on the used tape technology (2-5 minutes).

4.2.3 Recommendations

Generally, the same recommendations as described above for HSM activity on home (see section HSM on home). In addition to that, there are recommendations associated with HSM configured on cache.

- To prevent migrating uncached files[12] (which should not really occur in this scenario) the migration policy should omit uncached files. Find below an example defining uncached files:

```
/* define macro to identify uncached files */
```

[11] Storage Protect for Space Management version 7.1.3 and later does skip uncached files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.
[12] Storage Protect for Space Management version 7.1.3 and later does skip uncached files from migration. Nevertheless, the used migration policy should sort out uncached files to reduce the processing overhead.

```
define( is_uncached,
( NOT RegEx(misc_attributes,['[DXazu]'])
  AND NOT RegEx(mode,['[bcps]']) ) )
```

- Files that will be changed frequently in the cache fileset should not be migrated. The policy engine can be configured to identify files that do not change anymore, for example, by using the access time stamp. Find below an example to migrate files accessed 5 days ago, or longer:

```
/* define macro for modification time */

define( acc_age,
( DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) ) )

/* migration based on modification age*/

RULE 'modAge' MIGRATE
 FROM POOL 'system'
 TO POOL 'silver'
  WHERE (acc_age > 5)
```

4.3 Backup On Home And Cache

In this section, we elaborate on the system-migration use case with backup installed and configured on home and on cache, as shown in Figure 8.

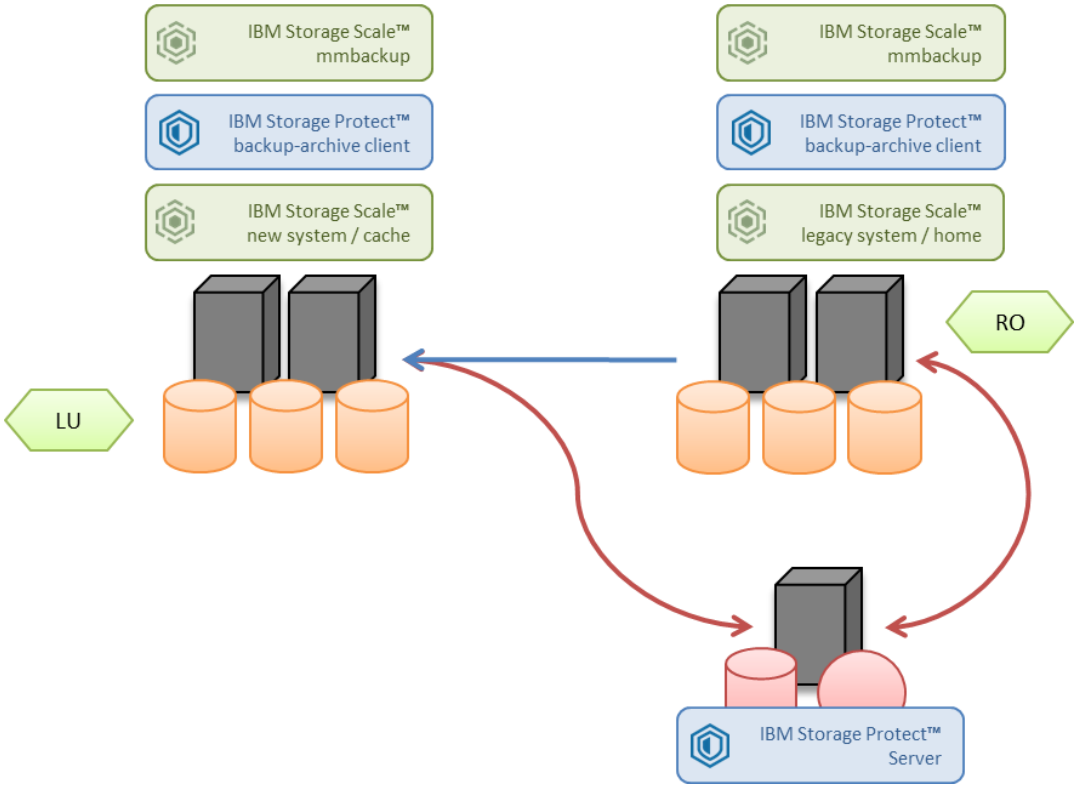

Figure 8: Backup on home and cache

The backup in IBM Storage Scale is accommodated with the mmbackup tool that uses the IBM Storage Protect backup client. Backup of home and cache can be configured independent. The IBM Storage Protect server file space must be different for the home fileset and the cache fileset. To prevent concurrent access to the same data from both AFM sites cache and home. This would lead to inconsistency in the backup.

4.3.1 Cache Operations

The following table summarizes the file operations on cache and consequences to home and cache with the focus on backup operations

Operation	Cache	Home
CREATE file	Next backup will generate a backup copy in the IBM Storage Protect server.	No impact to backup processing.
READ file	No impact to backup processing.	No impact to backup processing.
WRITE file	After the write operation, the modified file won't be replicated	No impact to backup processing.

Operation	Cache	Home
	to home. Next backup will generate new version of the file in the IBM Storage Protect server on cache.	
TRUNCATE file	Next backup will generate new version of the file in the IBM Storage Protect server.	No impact to backup processing.
RENAME or MOVE file within fileset	Next backup will generate a new file in the IBM Storage Protect server and expire the old file (name).	No impact to backup processing.
MOVE file to other fileset	Next backup will expire the moved file.	No impact to backup processing.
EVICTION	Eviction should be disabled.	No impact to backup processing.
DELETE file	Next backup will expire the deleted file.	No impact to backup processing.

4.3.2 Challenges

Challenges for backup on home and cache:

- Uncached files won't be backed up. This can cause inconsistency of the backed-up data in the cache if files that are uncached have not been backed up.

4.3.3 Recommendations

Recommendations for the usage of backup in the system-migration user scenario:
In addition, some recommendations are added in terms UNIX HSM is used as well on the same fileset.

- Ensure that files on cache that are in status uncached are backed up prior. Fetch all files from home to cache before activating backup (and HSM). In general, do not have uncached files in cache. Fetch all files from home to cache before activating backup (and HSM).

- To cope with the backup loads, certain filesets could be excluded from backup, using exclude lists or perform fileset level backups.

- When using HSM and backup for the same IBM Storage Scale file system and fileset, use the same IBM Storage Protect server. This allows to prevent files to be migrated before they have been backed up, using the management class parameter: MIGREQUIRESBkup=yes.

- Do not use the same instance of IBM Storage Protect server as destination for UNIX HSM on cache and home. This defeats the idea of disaster protection because if this single IBM Storage Protect server instance fails both copies of migrated files might get lost. However, when only using backup on cache and home the same IBM Storage Protect server instance might be used. In this case, however, it is important that the file space of the AFM cache fileset is different than the home fileset. This can be achieved by selecting a different name for the root directory of the fileset on cache and home.

- The backup schedules on home and cache should take into consideration the replication cycles. AFM replication is done automatically in the background from cache to home and if files change on cache, the files are replicated to home. There are two options to accommodate this:
- Run backup when files on cache are rarely changed or created, for example, during the night
- Delay asynchronous replication before starting the backup operation adjusting the AFM parameter afmAsyncDelay on the cache fileset before and after the backup.

- When the cache must be created (after a failure), pre-fetch all files back from home to cache before starting the backup. This also assures that subsequent backups do not cause fetching of files from home to cache and perhaps recalls on home.

- When pre-fetching files from home the policy engine can be used on home to identify all files that have been migrated. The UNIX HSM tape optimized recall method should be used to recall all these files on home before pre-fetching them to cache.

APPENDIX

References

IBM Storage Protect Documentation:

https://www.ibm.com/docs/en/storage-protect

IBM Storage Scale Documentation

https://www.ibm.com/docs/en/storage-scale

Integrating IBM Storage Protect with IBM Storage Scale:

https://www.ibm.com/support/pages/node/6116050

Petascale Data Protection with IBM Storage Protect:

https://www.ibm.com/support/pages/node/6115894

IBM Storage Protect for Space Management configuration guide

https://www.ibm.com/support/pages/node/6116842

Data Storage Management (XDSM) API:

http://pubs.opengroup.org/onlinepubs/9657099/

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it..

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

icensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee..

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS

IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.