Active File Management (AFM)

Spectrum Scale User Group 2017, Manchester

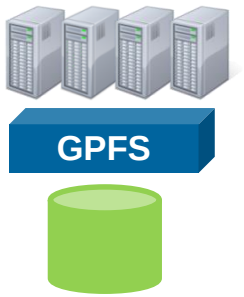
Nils Haustein · haustein@de.ibm.com

Thanks to Achim Christ for providing this material

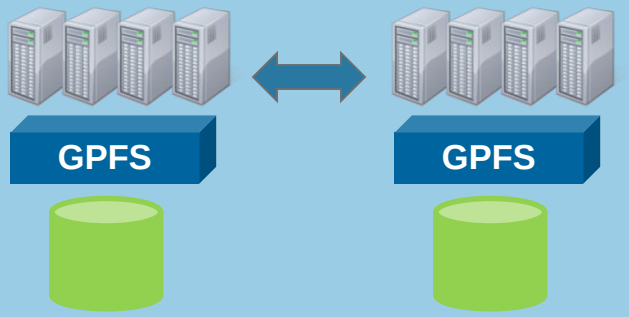
Table of contents

AFM Overview and Concepts

Spectrum Scale evolution



GPFS introduced concurrent file system access from multiple nodes

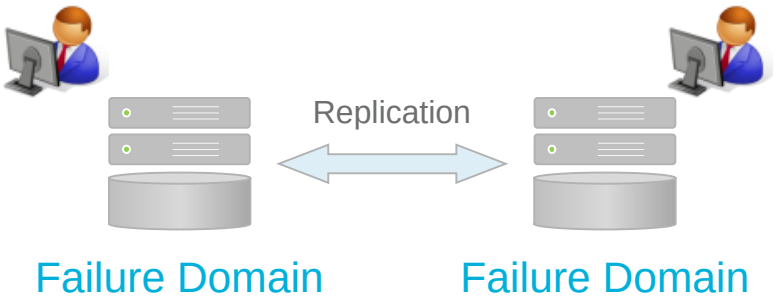


Multi-cluster expands the global namespace by connecting multiple sites

Active File Management takes global namespace truly global by automatically managing asynchronous replication of data

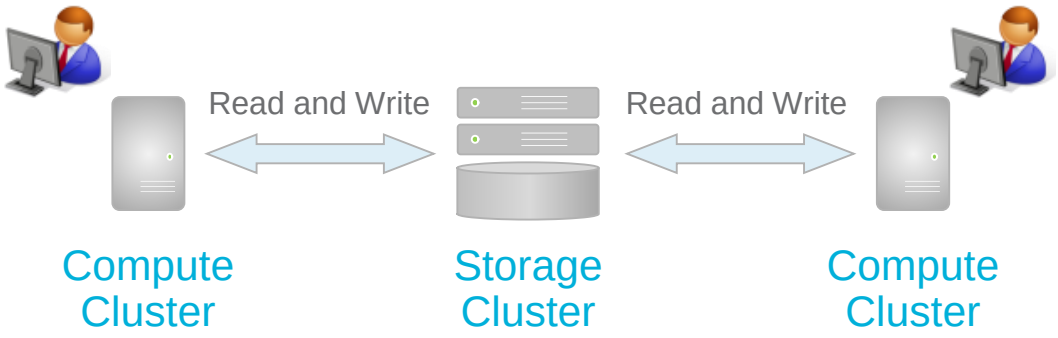
1993 2005 2012

Spectrum Scale “Stretched” Cluster



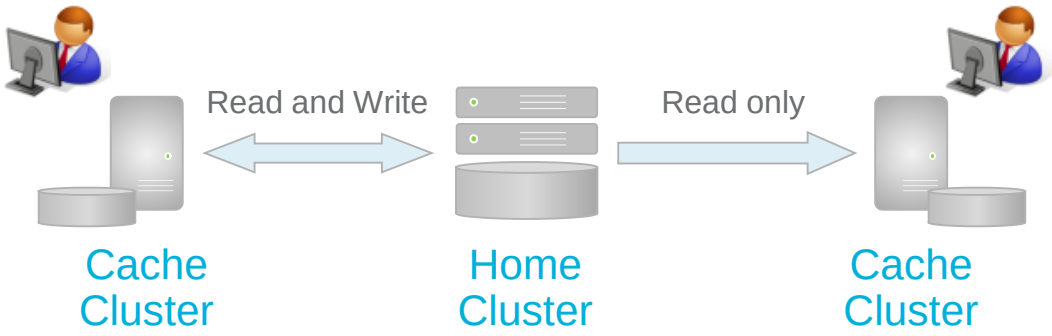
- Single Spectrum Scale cluster spans sites (failure domains)
  - Single administrative domain
  - Synchronous operation, consistent locking
- Synchronous replication between sites based on NSD Failure Groups
  - Distinct Failure Groups indicate resources which could fail simultaneously
  - Data can be replicated based on different Failure Groups
- Supports idea of high-availability
  - Failure of individual Failure Group compensated by GPFS
  - Details and recovery steps: Whitepaper | Wiki

Spectrum Scale Multi-cluster



- Independent Spectrum Scale clusters
  - Separate administrative domains
  - Synchronous operation, consistent locking
- Single storage cluster owns NSDs, one or many remote clusters mount file system(s)
  - Cross-cluster mount: Knowledge Center
  - Facilitates parallelism to optimize performance
- Unavailability of storage cluster affects all remote clusters (single data copy)
- Supports idea of multi-tenancy (clusters can be authorized for individual file systems)

Spectrum Scale Active File Management (AFM)



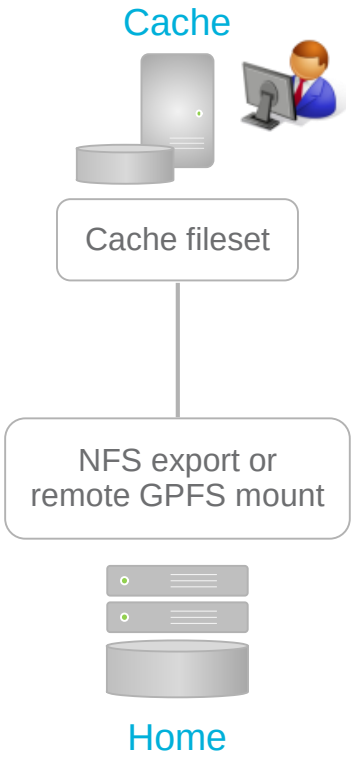
- Geographically dispersed systems share their files over WAN
  - Low bandwidth, expensive lines, temporarily unavailable, etc.

- Requirements
  - All users & applications should “see” their files stored somewhere else
  - Files should only be transferred when accessed and then cached locally

- Solution
  - Represent namespace (home) in remote system (cache) without transferring files
  - Transfer files when required and cache them, users work on local copy
  - Use parallelism to minimize transfer times
  - Asynchronous operation, no consistent locking

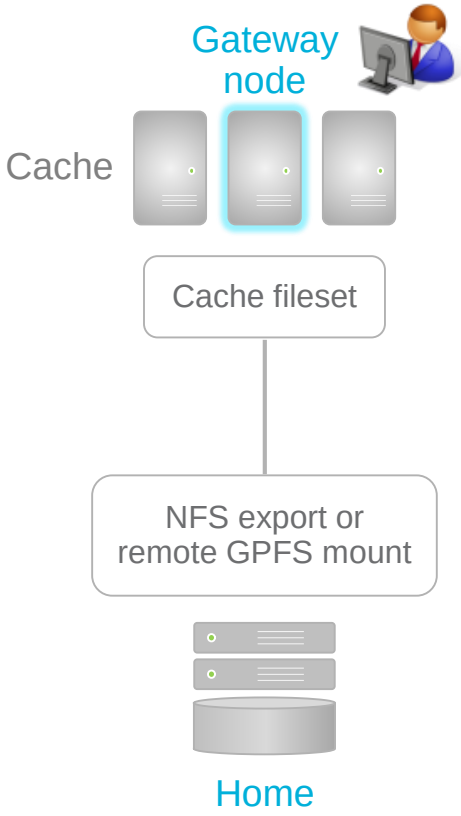
Active File Management (AFM) architecture

- AFM uses home-cache model
  - Single home provides primary storage of data which is exported
  - Exported data is cached in local GPFS file system
- Home can be NFS export or remotely mounted GPFS cluster
  - Only GPFS-based home file systems support ACLs, EA, and sparse files (irrespectively of NFS or GPFS protocol)
- GPFS cache presents home export in a fileset
  - One cache fileset can cache one home export
  - One cache server can cache multiple home exports (one fileset each)
- Different modes supported, can be combined per fileset
- AFM supported on AIX and Linux, gateway nodes must be Linux



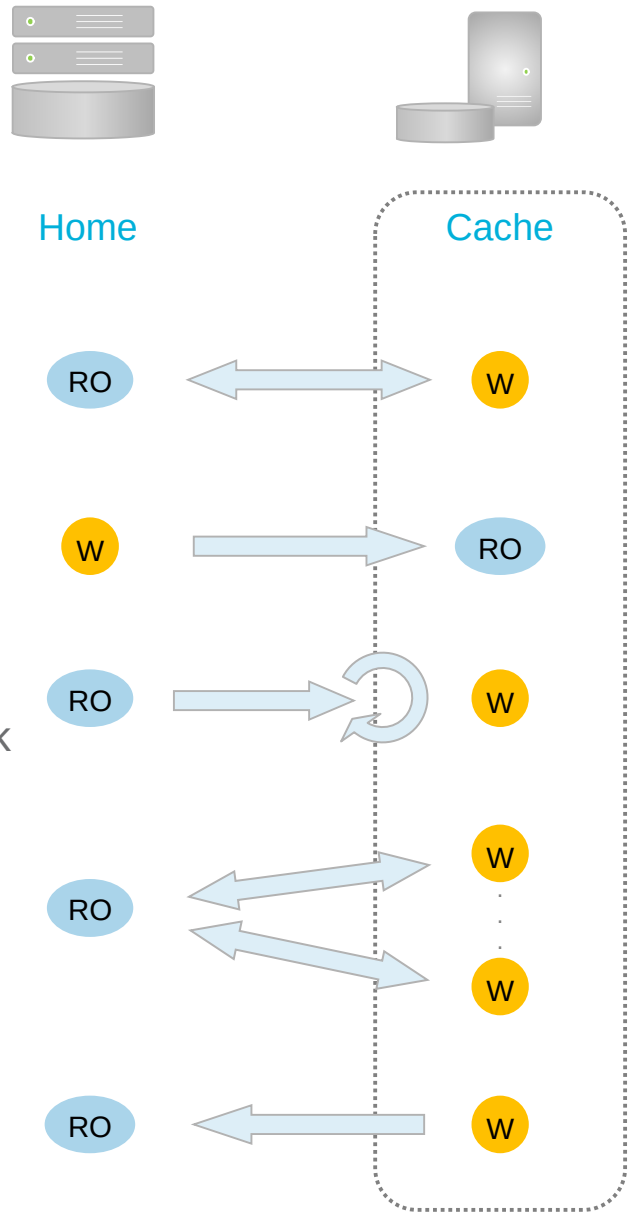
AFM gateway nodes

- Gateway node on cache manages communication with home
  - At least one GPFS cache node must be assigned as gateway node
  - Gateway node must be network connected to home server
  - Multiple gateway nodes can be used for redundancy and parallel I/O
  - Each cache fileset has gateway node (metadata server) in cache cluster
- Gateway nodes are setup during AFM configuration
  - Node role, recommendation to use dedicated resources
  - Command `mmchnode --gateway`
- Gateway nodes must be Linux, require server license



AFM cache modes

- Single Writer
  - Only cache can write data. Home can't change. Other peer caches have to be setup in RO mode
- Read Only
  - Cache can only read data, no data change allowed
- Local Updates
  - Data is cached from home like in SW mode
  - Once data is changed on cache it is not replicated back to home (stays local)
- Independent Writer
  - Multiple caches pointing to the same home
  - No file locking or write ordering from cache to home
- Primary / Secondary
  - Similar to SW, but no caching
  - All files created on primary are replicated to secondary



Basic AFM Operations (cache site)

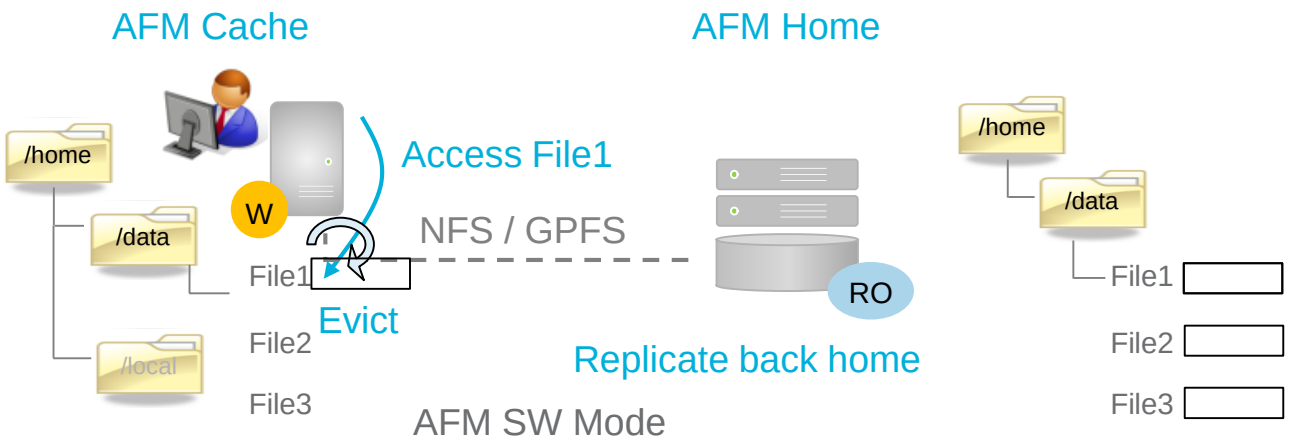
AFM Operation	Description
On-Demand fetching	Upon access of uncached file it is copied (fetched) from home to cache synchronously
Pre-fetching	Command driven fetching of files from home to cache
Replication	Changed files on cache are asynchronously replicated to home
Eviction	Cached files that have a copy in home are stubbed in cache
Peer Snapshot	Snapshot created on cache is propagated to home in the order of file changes and creates snapshot on home

- AFM operations depend on AFM modes
- The actual AFM mode is determined by the use case

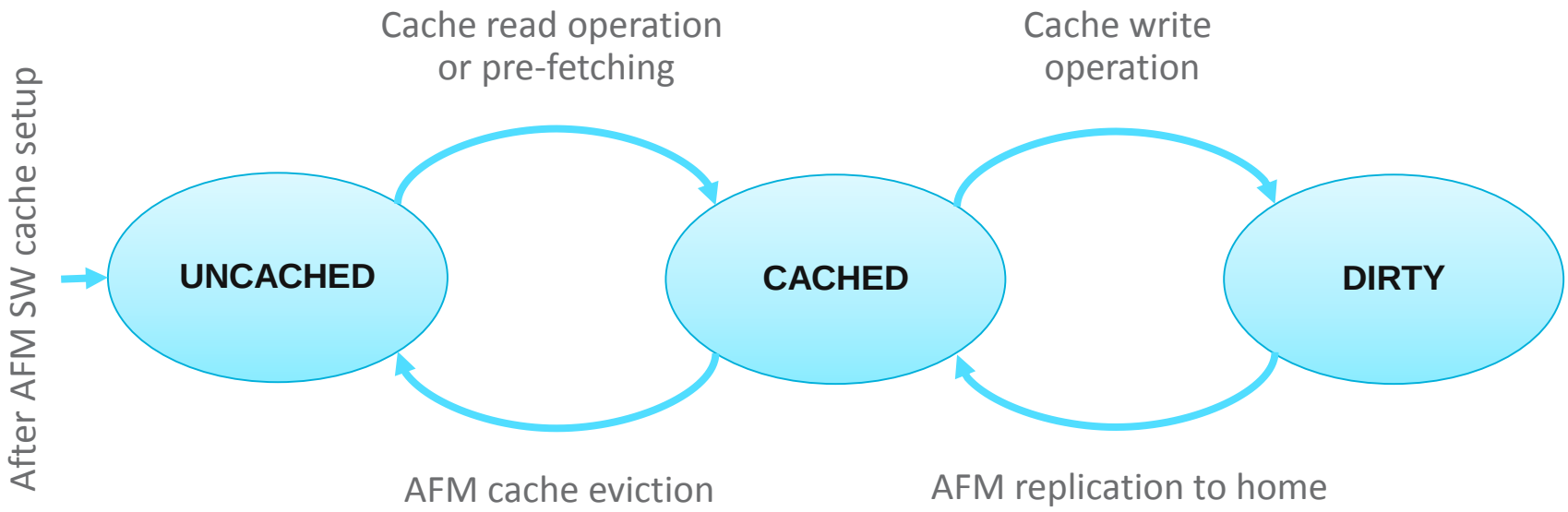
Use Case 1: Branch office

AFM caching use case (single writer)

- In AFM SW mode files are created and reside in Home:
  - File stubs (inodes) are created in Cache upon AFM SW setup
- Data is being copied from Home (fetched) to Cache upon file access (or pre-fetch operation)
- Changed files on Cache are replicated back to Home
  - New or changed files on home are not cached
- When file comes to rest it can be evicted from Cache based on thresholds
  - File remains visible in Cache but does not consume space
- Cache must be GPFS based, Home can be NFS server (no complex ACLs are transferred)



AFM file states in SW cache



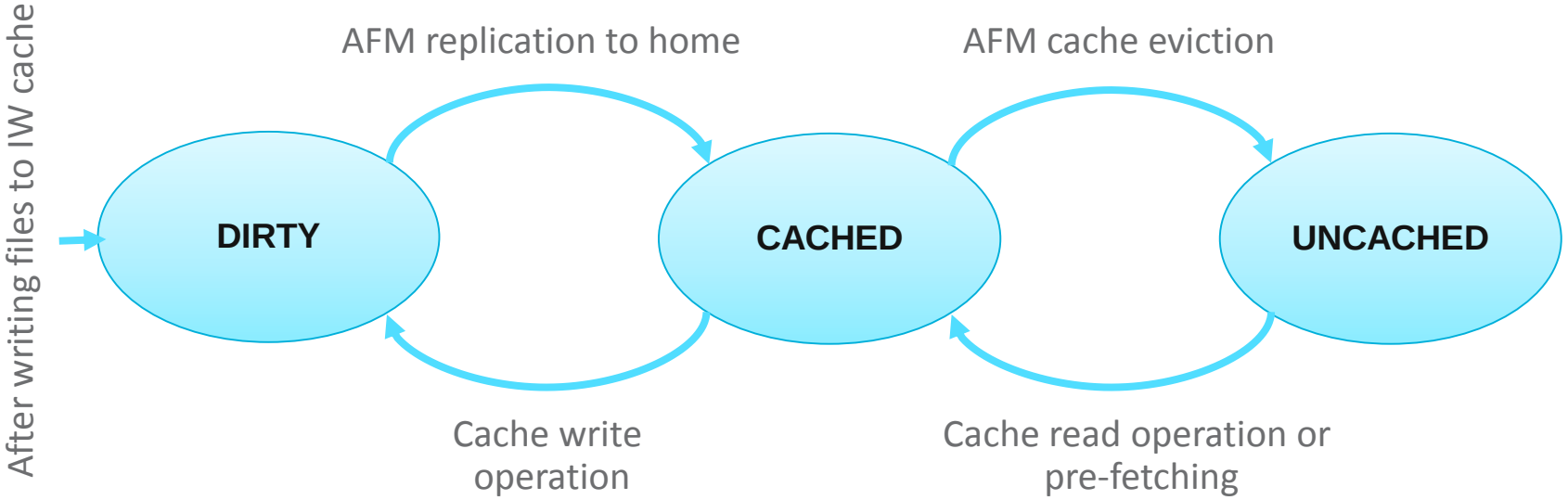
Files are created in Home, transferred to Cache and replicated back to Home upon modification.
Files are not modified in Home!

Use Case 2: Data ingest

AFM data ingest use case (independent writer)

- In AFM IW mode files are created in Cache and replicated to Home
  - File can also be created in Home
- New or changed files in Cache are automatically replicated to Home
  - New files in Home are cached in Cache
- Not a real DR solution because there is no notion of primary and secondary
- Multiple Caches can be connected to one Home, no conflict handling (last writer wins)
- Cache eviction frees up storage on Cache system for new data
- Cache and Home must be GPFS based

AFM file states in IW cache

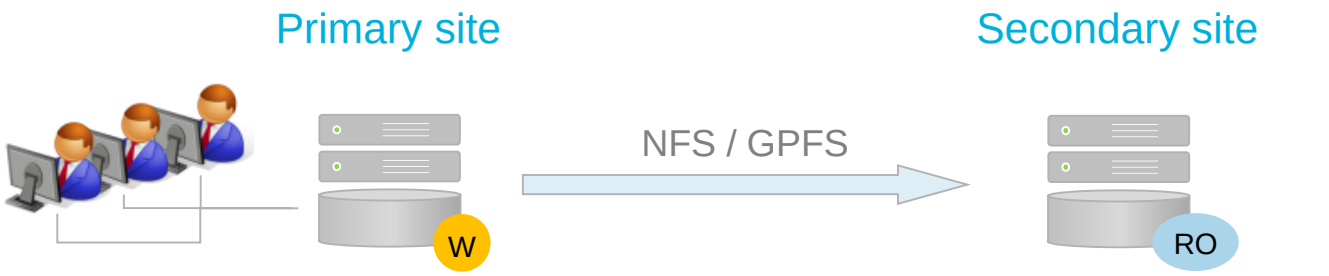


Files are created in Cache and replicated to Home.
Files can also be created in Home and are Cached in cache (no conflict handling – last writer wins)

Use Case 3: Disaster protection

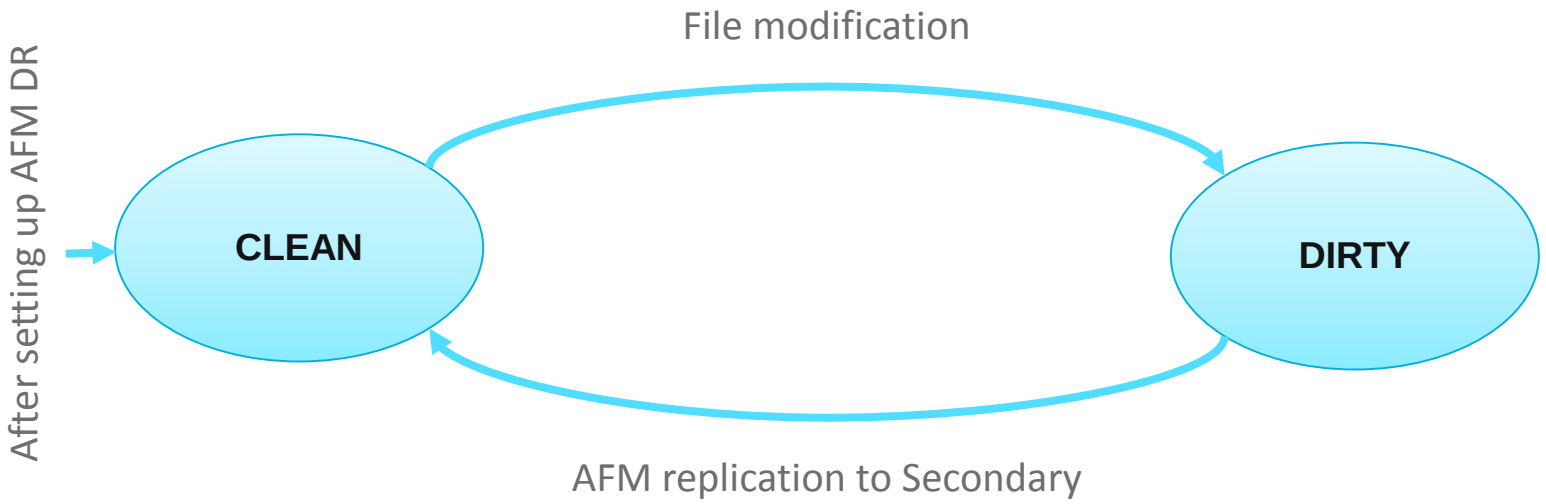
AFM Disaster Recovery (primary, secondary)

- In AFM DR mode files are created in Primary and replicated to Secondary
  - Secondary is read-only
- Automated peer-snapshots are created on Primary and propagated to Secondary
- Simple and efficient DR processes
  - When Primary is down then Secondary can be made new primary during failover
  - When old Primary is back online it can be made primary again during failback
    - During failback files delta is copied from new to old Primary
- When Secondary fails a new Secondary can be defined
- Primary and Secondary must be GPFS based



AFM DR Mode

AFM file states in DR primary

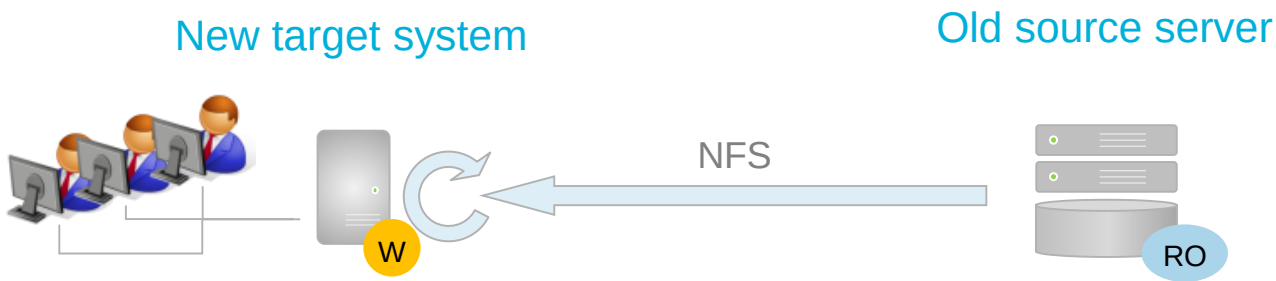


Files are created in Primary and replicated to Secondary.
Upon disaster Secondary is promoted to Primary (failover).
When old primary is back it can be made primary again (failback)

Use Case 4: Migration

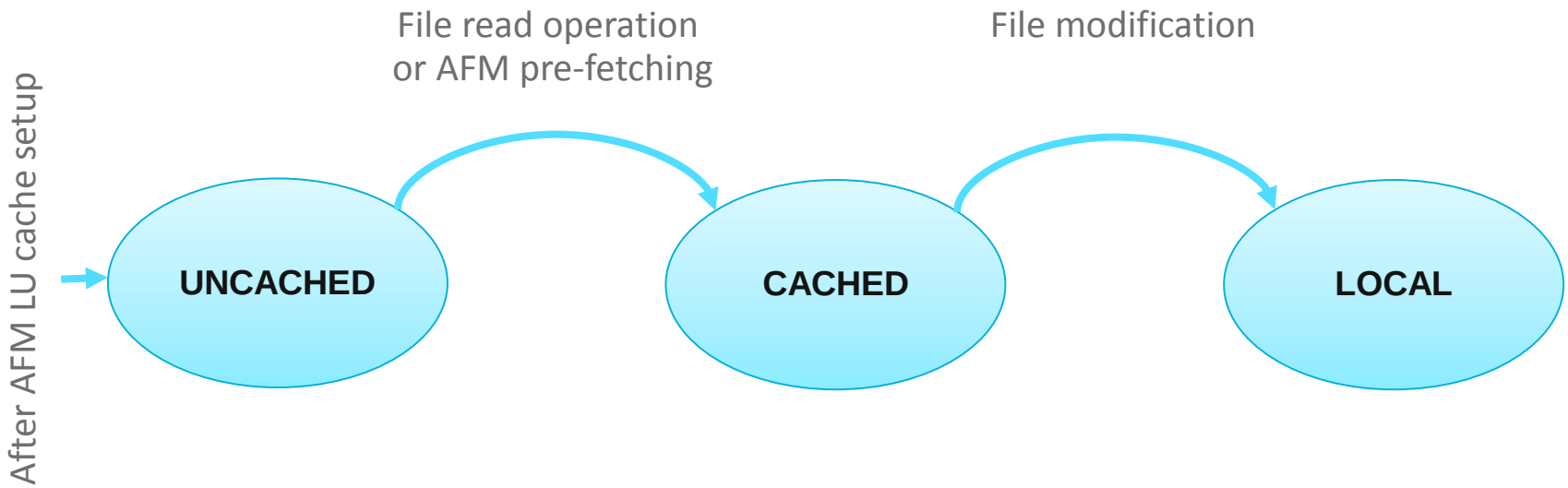
AFM migration use case (local update)

- In AFM LU mode files reside in old NFS server
- After establishing AFM LU relationship Cache “sees“ all files from old NFS server
  - Cache is configured in LU mode
  - Home provides NFS export(s)
  - Files can be pre-fetched (migrated) based on results of policy scans
- Switch over when sufficient files are pre-fetched
  - Un-cached files accessed on cache are transferred from home
  - Files changed on cache are not replicated back
- Cache must be GPFS based, Home can be NFS server (no complex ACLs are transferred)



AFM LU Mode

AFM file states in LU cache



Files reside in Home (old NFS server) and are transferred
to Cache upon access or with pre-fetch operation.
When files are modified on Cache there is no replication back to Home

Hints and tips

Setting up AFM caching

- On the Home cluster (GPFS based)
  - Home can be fileset or directory in GPFS file system
  - Create NFS export
  - Set home export path (`mmafmconfig ExportPath`)
- For a non-GPFS based Home create the NFS export
- On the Cache cluster
  - Define one or more Gateway nodes (`mmchnode -gateway`)
  - Create cache fileset

```
mmcrfileset fs fset
            -p afmTarget=nfsnode:/export-dir
            -p afmMode=sw --inode-space=new
```

  - Link fileset (`mmlinkfileset fs fset -J path`)
  - List fileset to see the files

Monitoring AFM

- Check AFM status: `mmafmctl fs getstate -j fset`

```
FsetName  Target          CacheState  GatewayNode  QueueLength   QueuenumExec
--------  ------          ---------   ---------    ---------     -----------
fset11    nfs://home/dir  Active      DRhost1      0             845
```

- Check AFM parameter: `mmlsfileset fs fset --afm`

```
…
Status                                  Linked
Path                                    /fs/fset
afm-associated                          Yes
Target                                  nfs://home-server/export
Mode                                    single-writer
File Lookup Refresh Interval            30 (default)
Dir Lookup Refresh Interval             60 (default)
Async Delay                             15 (default)
Prefetch Threshold                      0 (default)
Eviction Enabled                        yes (default)
…
```


Controlling AFM

- Using the command mmafmctl

```
mmafmctl Device {resync | cleanup | expire | unexpire} -j FsetName
  or
mmafmctl Device {getstate | flushPending | resumeRequeued}
                [-j FsetName]
  or
mmafmctl Device prefetch -j FilesetName
                [[--inode-file PolicyListFile] |
                 [--list-file ListFile]]
                 [-s LocalWorkDirectory]]
  or
mmafmctl Device evict -j FilesetName
                [--safe-limit SafeLimit] [--order {LRU | SIZE}]
                [--log-file LogFile] [--filter Attribute=Value...]
```

AFM parameters

- Set using `mmchconfig`, `mmcrfileset`, `mmchfileset`
  - `mmchconfig` parameters are global defaults
  - Fileset level setting override defaults

- AFM tuning options are dynamic

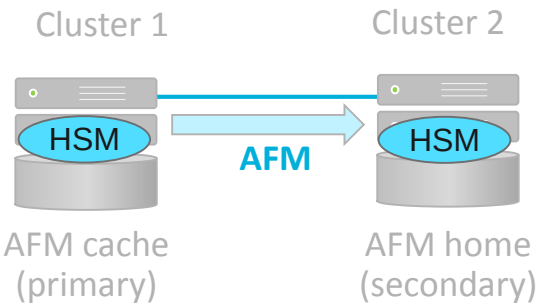
- Fileset AFM options (`mmchfileset -p afmAttribute=Value`)
  - `afmAllowEviction`
  - `afmAsyncDelay`
  - `afmDirLookupRefreshInterval`
  - `afmDirOpenRefreshInterval`
  - `afmExpirationTimeout`
  - `afmFileLookupRefreshInterval`
  - `afmFileOpenRefreshInterval`
  - `afmMode`
  - `afmShowHomeSnapshot`

AFM performance monitoring

- GPFS has various sensors for AFM related data
- To gather AFM related performance data use command: `mmperfmon query metrics`
- To get a list of AFM related performance data: `mmperfmon query –list=metrics`
- Examples

```
# mmperfmon query
gpfs_afm_bytes_read,gpfs_afm_bytes_written,gpfs_afm_bytes_pending
```


General considerations with AFM and TSM HSM



- Files changed on Cache are dirty until replication finished
  - Dirty files cannot be migrated, HSM 7.1.3+ ignores them
- Un-cached files are not present in Cache and need to be fetched from Home
  - Un-cached files cannot be migrated, HSM 7.1.3+ ignores them
- AFM Eviction conflicts with TSM HSM stubs - Do not use eviction with TSM HSM !!
- Potential recall storms on Cache when many migrated files are accessed
- Potential recall storms for on Home during replication, access and pre-fetching
- Recalls require additional storage capacity in file systems
- Snapshots and TSM HSM are conflicting
  - Triggers recalls when migrated files that are in a snapshot are deleted
- More Information: Configuration guidance for Spectrum Scale with Spectrum Protect..

Links

- GPFS Knowledge Center
  https://www.ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html
- GPFS Redbook
  https://www.redbooks.ibm.com/abstracts/sg248254.html?Open
- GPFS Wiki
  http://www.ibm.com/developerworks/wikis/display/hpccentral/General+Parallel+File+System+(GPFS)
- GPFS FAQ
  - http://www.ibm.com/support/knowledgecenter/SSFKCN/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

Disclaimer

- This information is classified for **IBM and Spectrum Scale User Group Internal use** and shall not be used or published outside this scope.
- This information is provided on an "AS IS" basis without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Some jurisdictions do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Important notes:

- IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.
- IBM makes no warranties, express or implied, regarding non-IBM products and services, and any implied warranties of merchantability and fitness for a particular purpose. IBM makes no representations or warranties with respect to non-IBM products. Warranty, service and support for non-IBM products is provided directly to you by the third party, not IBM.
- All part numbers referenced in this publication are product part numbers and not service part numbers. Other part numbers in addition to those listed in this document may be required to support a specific device or function.
- When referring to storage capacity, GB stands for one billion bytes; accessible capacity may be less. Maximum internal hard disk drive capacities assume the replacement of any standard hard disk drives and the population of all hard disk drive bays with the largest currently supported drives available from IBM.

IBM Information and Trademarks

- The following terms are trademarks or registered trademarks of the IBM Corporation in the United States or other countries or both: the e-business logo, IBM, system x, system p, System Storage, GPFS
- Microsoft Windows is a trademark or registered trademark of Microsoft Corporation.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.