IBM Spectrum Scale AFM/AFM DR

April 19th 2018
UK User Group Event

Venkat Puvvada (vpuvvada@in.ibm.com)

Motivation for AFM

- Data sharing across geographically distributed sites is common
  - While the bandwidth is decent, latency is high
  - Network is unreliable, subject to outages
- Infrastructure needs to be scalable to move data across the WAN
  - Mask latency and fluctuating performance of the network
- Applications desire local performance for remote data
  Move data closer to compute servers
- Traditional protocols for remote file serving are chatty and unsuitable
- Large files (VM images, virtual disks) are becoming predominant
- Existing caching systems are primitive

AFM caching basics

- Sites - two sides to a cache relationship
  - A single home cluster
    - Presents a fileset that can be cached (export with NFS or GPFS)
  - One or more cache clusters
    - Associates a local fileset with the home export
- AFM Fileset
  - Independent fileset with per-inode state in xattrs
  - Data is fetched into the fileset on access (or prefetched on command)
  - Data written to the fileset is copied back to home
- Gateway Node (designation)
  - Receives requests (GPFS RPCs) from other (application) nodes on VFS calls
  - Maintains an in-memory queue of pending operations
  - Moves data between the cache and home clusters
  - Monitors connectivity to home, switches to disconnected mode on outage, triggers recovery on failure

AFM caching basics

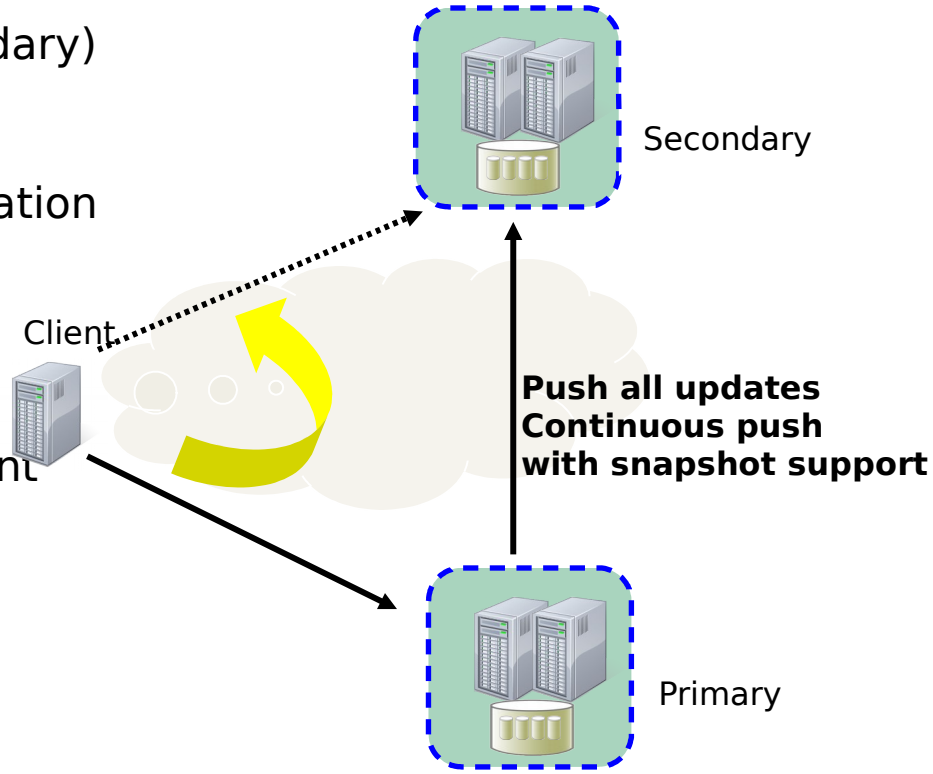
- Multiple caching modes
  - Read Only
  - Local Update
  - Single Writer
  - Independent Writer
- Cache behavior
  - Data updates to home are asynchronous
  - Writes can continue to cache when the network link to home is unavailable
  - Data is managed like in a cache but on stable storage in GPFS
- Transport Protocol
  - NFSv3 (also native GPFS in 4.1) over TCP/IP
- Multiple cache relationships per file system
  - Cache relationships are at a fileset level
  - The same file system can contain multiple homes, caches and uncached data

Disaster recovery with AFM: Goals

- Replicate data from primary to secondary
  - Relationship is Active-Passive (primary is read/write, secondary is read-only)
  - Allow data to be “trucked” out of band for initial setup
- Promote secondary to be a primary in case of primary failure
  - Will require reverting the secondary to a previously known consistent state (based on RPO) and could result in loss of data
- Tolerate failure of relationship with secondary with no interruption of operations at primary
  - due to network and/or secondary outage
  - Re-establish relationship once secondary is back from outage
- Launch a new primary from secondary in case of primary failure
- Populate a new secondary from primary in case of secondary failure
- Allow the relationship to be suspended and resumed
  - for administrative/maintenance purposes
- Replicate whole filesystem or parts of filesystem
  - Using fileset granularity allows primary (filesystem) to be replicated piece-wise (fileset level) to multiple secondary sites

AFM Replication for Disaster Recovery

- Primary-Secondary Relationship
  - Active-Passive (RW primary, RO secondary)
  - AFM Fileset in “push only” mode
- Continuous replication
  - Trucking and In-band initial data population
  - Only deltas are pushed
- Snapshots for consistent copy
  - Primary and secondary snapshots are coordinated
  - Snapshots are not application consistent
- Fileset granularity
- Failover and failback
  - Handle primary, secondary failures
- RPO and RTO from mins to hours
  - Depends on data rate change, link bw



Migration using AFM

- Migrate using either NFS or NSD protocol
- No real downtime required, switch applications immediately to the Spectrum Scale
- Replication back to the old system during the migration (Single Writer mode)
- Data modification at both sites during the migration (Independent writer mode)
- No replication back to old system (Local updates mode)
- Never use Independent Writer mode for migration unless both sites are changing the data from the same fileset
- Migrate the stub files, no recalls
- File level granularity, provide list of files to migrate
- Run policy to quickly check whether all files are migrated
- Disable AFM after the migration
- Convert to AFM DR filesets if DR is the requirement

AFM/AFM DR debugging tips

- Replication performance
  - Increase the flush threads (afmNumFlushThreads)
  - Increase async delay (afmAsyncDelay)
    - For write coalescing
    - To avoid conflicting locks between SMB and AFM
  - Increase write merge length to enable read ahead (afmMaxWriteMergeLen)
  - Network tuning as documented in KC
  - Limit number of parallel recoveries (afmMaxParallelRecoveries). Running recovery is expensive which involves creating the snapshot, policy scan, home readdir etc..
  - Tune timeout values on flaky networks to prevent fileset being move to Unmounted state (afmRevalOpWaitTimeout, afmSyncOpWaitTimeout, afmAsyncOpWaitTimeout)
  - Choose NSD protocol if network is reliable
  - Increase afmHardMemoryThreshold to avoid dropping the queue and running recovery

AFM/AFM DR debugging tips

- Application performance
  - Refresh intervals tuning in Independent Writer mode
  - Set afmDIO=2, for making synchronous operations not pushing the writes in IW mode

- Fileset is not replicating, not in sync with home/secondary
  - Requeued messages
    - No space at remote
    - Temporary protocol errors
    - Lock conflict between AFM and SMB
    - Renames operations (bug in AFM, pre 5.0.0 versions)
  - recovery+resync keeps failing
    - Try resync/failover or changeSecondary commands.
  - Rate of incoming changes are more than AFM could replicate
    - Throttle incoming requests (afmMaxThrottle)

AFM/ AFM DR recommendations

- 100 AFM enabled filesets per filesystem (not hard limt)
- 100 million files per fileset (not hard limit)
- Dedicated gateway nodes
- Limit number of filesets per gateway node based on workload.
- /var directories needs to be provisioned for bigger storage (based on number of filesets handled by gateway node * files in fileset * 255 bytes) space for internal usage during recovery/resync.
- Have enough network bandwidth between primary/cache and secondary/home sites to replicate incoming changes or migrate the data.
- Use of outband trucking for AFM or AFM DR is deprecated. Users can still copy the data to secondary/home site, but use inband option for conversion.
- Never use Independent Writer mode if cache is the only site updating it
- Never use Independent Writer mode for migration unless both sites the changing the data from the same fileset.
- Split prefetch list files, not more than one million files per invocation. Background prefetch reads could stop the replication back.

AFM/AFM DR enhancements (intended)

- Continuous replication after gateway node failure or queue drop, don't wait for the recovery/resync to complete
- Improve independent writer performance by doing asynchronous revalidation
- Allow users to choose fileset to gateway mapping
- Minimize use of local filesystem storage during the recovery and resync
- Externalize some of the undocumented configuration parameters used for tuning
- Small write performance improvements
- Dependent filesets support
- Support over a billion files per fileset
- Enhance AFM migration and prefetch
  - Support directory level prefetch
  - Provide list of prefetch failed files
  - Data movement statistics

Questions?