Spectrum Scale: Active File Management

Feature Updates

User Group Meeting, Boulder, 17th April 2019

Venkat Puvvada (vpuvvada@in.ibm.com)

AFM recent improvements

- Fileset to gateway node mapping.
  - Users can choose gateway node for the fileset
  - Supported from 5.0.2 using afmHashVersion=5
  - Gateway node can assigned using the mmchfileset command to existing filesets
- Prefetch and migration
  - Performance improvements
  - Directory level prefetch
  - Better queuing statistics
  - Multiple options for controlling the prefetch behavior (some are not documented)
- Resync Performance
  - resync/changeSecondary/failover command performance improvements
  - Operations from the same directory are not queued serially for the better replication performance

AFM features (currently not documented)

- Mount wait timeout
  - AFM by default waits for 8 seconds for the remote mount to happen
  - Set mount wait timeout using command “mmfsadm afm mountWaitTimeout <time in seconds>”
  - In memory value, not persitent across daemon restarts

- Set ACLs on RO (read-only) mode fileset.
  - Useful during the migration.
  - Useful when the ACLs from the home needs to be ignored.
  - mmafmlocal mmputacl -i <acl file> <path>

AFM features (currently not documented) afmIOFlags

- Can be be set at the fileset level from 5.0.3 (ex. mmchfileset ... -p afmIOFlags=8)

- flags=0x1, 0x2, 0x4 – reserved

- flags=0x8
  - Home have migrated files. Set the file's state accordingly during the readdir/lookup.
  - Don't trigger read on migrated files, set cached bit automatically.

- flags=0x10
  - Don't pull ACLs from home system during the migration
  - Adhere to inheritance ACLs set at the cache

AFM features (currently not documented) afmIOFlags

- flags=0x20
  - Don’t compare mtime with nanoseconds granularity for the filetime changes
  - Can be used with AFM resync if data is already copied to home/secondary site using external tools like rsync.
- flags=0x40
  - Force ctime change from the home
  - Can be used to update the ACLs/EAs from home if the user forgot to enable AFM at home previously
- flags=0x80
  - Skip recovery during the resync
  - Improves the resync performance
  - Might leave extra files at home/secondary site

AFM features (currently not documented) afmIOFlags

- flags=0x100
  - Avoid in-memory message queue drop during the conflicts or unhandled errors
  - Can be used to debug the frequent queue drops where there are not enough traces

- flags=0x200
  - Async revalidation in caching modes like (RO, IW and LU)
  - Revalidation happens in the background, not in the application path
  - Provides better performance in the IW mode
  - Should not be used during the initial migration cut-over

AFM prefetch options (currently not documented, 5.0.2.x or 5.0.3.x)

- --prefetch-threads
  - Number of threads used for the queuing the prefetch
  - Default 4 and max 36

- --gateway
  - Specify the gateway node to run the prefetch (default it runs on the MDS)
  - --gateway local, run on the local gateway node
  -
- --readdir-only
  - Do the recursive readdir, namespace population
  - Used with –directory , --dir-list-file options
  - Can be used for namespace population.

AFM prefetch options (currently not documented, 5.0.2.x or 5.0.3.x)

- --dir-list-file
  - List file containing the directories
  - Recursively prefetch all the sub-directories

- --nosubdirs
  - Don't recurse on subdirs
  - Can be used with –dir-list-file option to skip subdirs

- --split-dirs
  - Merge the multiple entries from the different directories queue to the daemon
  - Improves the replication performance as VFS serializes the lookups on the same directory using the directory inode mutex.

AFM enhancements (intended)

- Dependent fileset support
  - Allow linking of dependent filesets under the AFM enabled independent fileset.
  - Dependent filesets share the independent fileset target and other config options
  - Allow migration to pull the data to the dependent filesets in the cache
  - User’s responsibility to create the dependent filesets at both sides before starting the replication or migration
- Use multiple gateway nodes for the AFM prefetch
  - Performance improvements
  - Directory level prefetch, directory list file prefetch
  - Better queuing statistics
  - Multiple options for controlling the prefetch behavior (some are not documented)
- Parallel IO
  - Use the user defined mapping to mount the target to the multiple NFS servers
  - Replicate the data using the multiple mounts, multiple streams
  - Dependency checking at the daemon level

AFM enhancements (intended)

- Continuous replication after gateway node failure or queue drop, don't wait for the recovery/resync to complete
- 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
- Support over a billion files per fileset
- AFM migration statistics, better estimation on time to completion.
- Generic backend support including replication to COS or caching.
- NFSv4 as backend protocol
- Immutability and IAM modes
- External tools support for replication like rsync etc..
- AFM DR with LTFS and TSM support