



April 2017

Flipping and Flapping

Some things we learned this year.

Vic Cornell

Senior Systems
Engineer, DDN

Luke Raimbach

Application support
Engineer, DDN

AFM - “Poor Man’s” DR ?

- ▶ **AFM IW or SW cache can used to push data to DR file systems over NFS**
 - All work happens in the cache – home can be relatively “dumb and slow”
- ▶ **Scheduled snapshots at home**
 - Simple crontab or SS GUI configured snapshots – protect your data without the overhead of managing snapshots on “Live”
- ▶ **PRO: Much cheaper and easier to then AFM DR**
 - Advanced / Data Management Edition is not required.
- ▶ **CON: Consistency is NOT guaranteed (no RPO / RTO)**
 - Some files may not arrive in for the expected snapshot (captured next time)

Which AFM Mode?

▶ Single Writer

- Does not check consistency with AFM Home
- Might be deprecated in future releases?
- Possibly receives less developer attention?
- Can get itself in a pickle.

▶ Independent Writer

- Introduces some extra overheads by referencing AFM Home periodically
- Feels like this mode receives more developer attention!
- Overhead Workaround: Tune some parameters beyond a sensible limit (e.g. snapshot retention period), so it behaves like Single Writer:
 - `afmDirLookupRefreshInterval` \geq `$snapshotRetentionPeriod`
 - `afmDirOpenRefreshInterval` \geq `$snapshotRetentionPeriod`
 - `afmFileLookupRefreshInterval` \geq `$snapshotRetentionPeriod`
 - `afmFileOpenRefreshInterval` \geq `$snapshotRetentionPeriod`

Surprise RPCs!

- ▶ Clients writing to an Active AFM cache send an RPC to the AFM Gateway node for each operation;
- ▶ AFM Gateway node adds the operation to the AFM queue for that fileset, then responds to the client RPC;
- ▶ Client waits for AFM RPC completion before retuning to the application.
- ▶ 1.4x to 4x slowdown over native FS work!
- ▶ Overhead is visible when the cache is “Active” on an AFM gateway node – i.e. is mounted and an AFM queue is being processed.

Diagram for Surprise RPCs?

- ▶ For each IO operation modifying data in the cache fileset that any client node in the cluster hosting the cache undertakes, that client must send (and the gateway must acknowledge) an RPC to place the operation inside the AFM queue, before the client daemon returns to the application.

Impact:

- ▶ An OpenFOAM workload which we were running inside a SW cache fileset for F1 customer.
- ▶ Results showed a 2x slowdown
- ▶ Didn't show up on benchmarks with large sequential writes - assumption is that the relative number of RPCs to the gateway is relatively small.
- ▶ The OpenFOAM workload involved a huge number of file creates, small writes, and file deletes => large number of RPCs. The AFM gateway then becomes a bottleneck when a large number of client nodes want RPC responses in a short time!

Simple Work Around

- Disable the cache during local workload
 - mmadmctl device stop -j fileset
 - Run workload (no AFM RPCs to “stopped” cache)
 - mmadmctl device restart -j fileset
- Could be activated by scheduler/cron etc.

Simple Work Around

- Pros
 - Improves performance during “production” hours.
 - No real changes to existing setup.
- Cons
 - There has to be a “non - production” time slot.
 - Compresses all transfers into smaller window – might need faster link.
 - mmadmctl device restart –j fileset runs a recovery which can impose a significant overhead. (got to love those python scripts).

Complex Workaround – Flipping

▶ “Cache Reversal”.

- Run the workload on the “home” filesystem.
- DR filesystem contains a number of read-only AFM caches which point back to home.
- Data are enumerated on the “Home” and then candidate files are pre-fetched to cache in DR filesystem.

Complex Workaround - “Cache Reversal”.

- A prefetch job in the cache cluster first calls a node in the home cluster to run an ILM job collecting a list of files.
- This list is fed into the AFM prefetch function using “mmafmctl”
- Files are “pulled” from “Home” (production) to cache in DR.
- A snapshot is then taken of the cache.
- In this case as we are creating an archive, the “Home” cluster runs another ILM job deleting anything over 30 days old. (“Check home” is disabled on the cache and we have snapshots)

Complex Workaround - “Cache Reversal”.

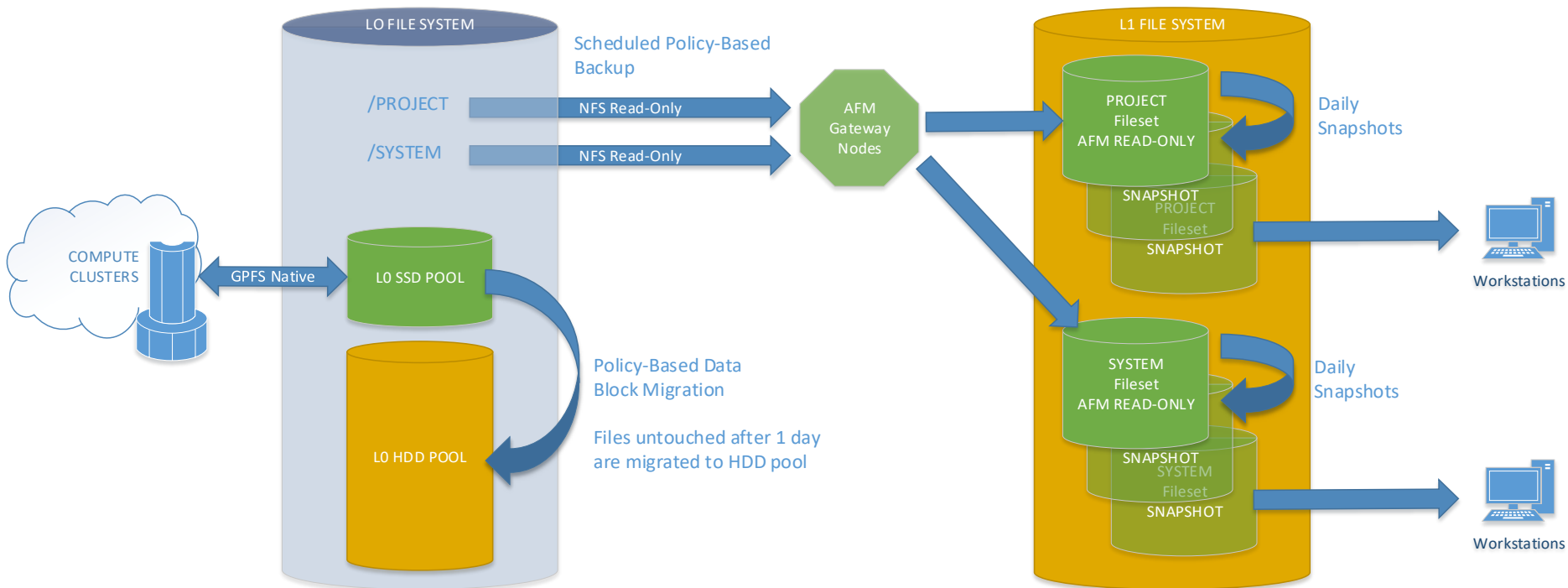
▶ Pros

- DR can be run all day – max use of link.
- Can have more flexible fileset structure at home as NFS export points can be moved or changed.

▶ Cons

- Overhead of ILM on production if metadata subsystem is weak.
- Ugly
- Scripts to support.
- Relies on internal logic – not externally developed logic

Final Workflow Architecture



Flapping

▶ Metanodes

- “There is one metanode per open file. The metanode is responsible for maintaining file metadata integrity. In almost all cases, the node that has had the file open for the longest continuous period of time is the metanode. All nodes accessing a file can read and write data directly, but updates to metadata are written only by the metanode. The metanode for each file is independent of that for any other file and can move to any node to meet application requirements.”

▶ Multicluster Networks

- Where there are multiple Multicluster client networks there is no requirement for them to “see” each other.

Flapping

- ▶ **Multicluster setups with shared file access have a high probability of “MetaNode Flapping”**
 - *“MetaNode role transfer occurs when the same files from a filesystem are accessed from two or more “client” clusters via a MultiCluster relationship.”*
- ▶ **Metanode flapping can be exacerbated by using the File System for Message Passing**
 - Control machines and users workstations watching directories and files for changes increase the frequency of role transfers.
- ▶ **Each MetaNode Role Transfer “Flap” forces all metadata updates to be flushed to disk.**

Proposed Solution

- ▶ **Implement an AFM Buffer**
 - If your work load is an ingest/analyse one, you can buffer writes from ingest nodes into a dedicated AFM Caching File System.
 - De-stage AFM caches to main File System over NFS
 - Decoupling the ingest and the analysis processes means no MetaNode transfer can happen for directories collecting new data
- ▶ **Merge Protocol Node & NSD Server Cluster**
 - User workstations or other monitoring tools will no longer “steal” the MetaNode role.