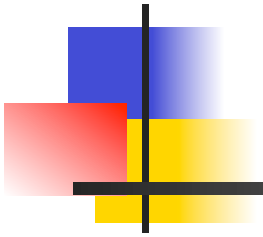
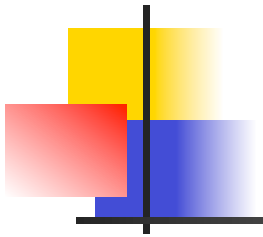


GPFS for Advanced users

Active File Management

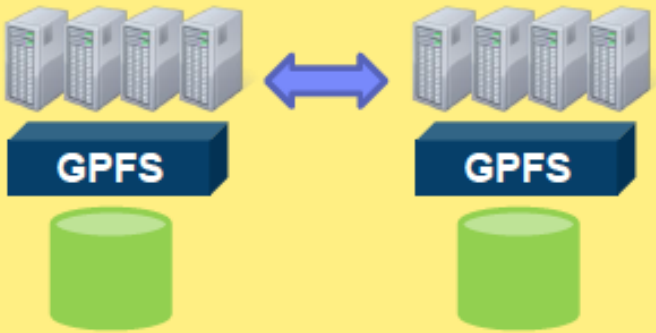


Motivation

- 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

Evolution of the global namespace: GPFS Active File Management (AFM)

GPFS introduced concurrent file system access from multiple nodes.



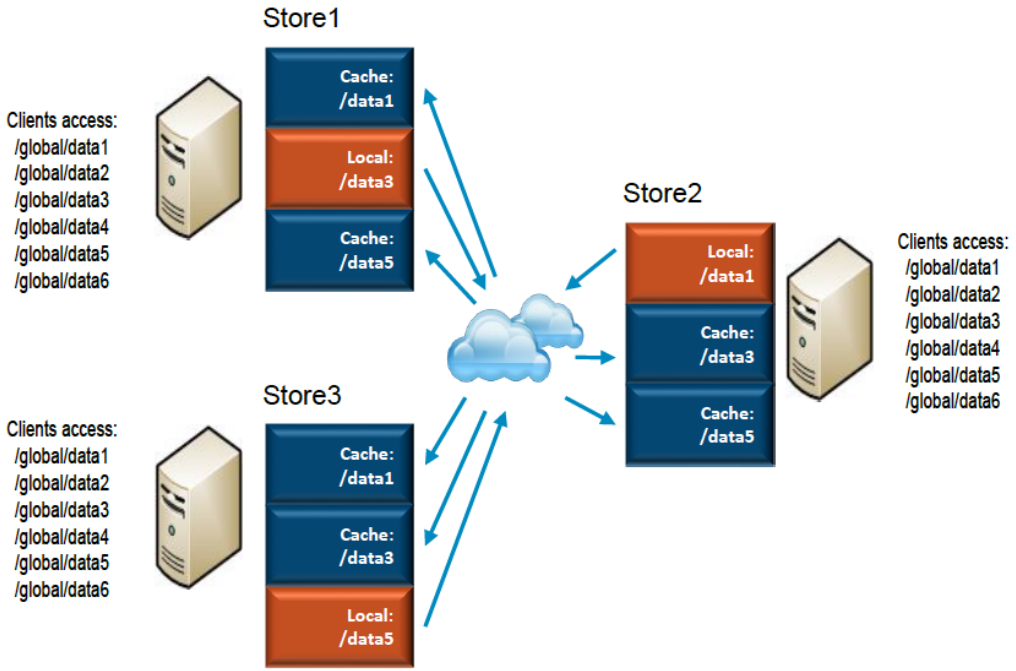
Multi-cluster expands the global namespace by connecting multiple sites

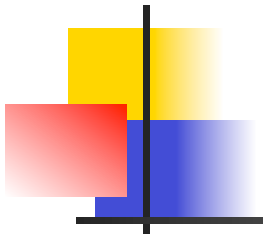
AFM takes global namespace truly global by automatically managing asynchronous replication of data

1993 2005 2011

Active File Management

- Enables sharing data across unreliable or high latency networks
- location and flow of file data between GPFS clusters can be automated.
- Relationships between GPFS clusters using AFM are defined at the fileset level.
  - A fileset in a file system can be created as a “cache” that provides a view to a file system in another GPFS cluster called the “home.” File data is moved into a cache fileset on demand.



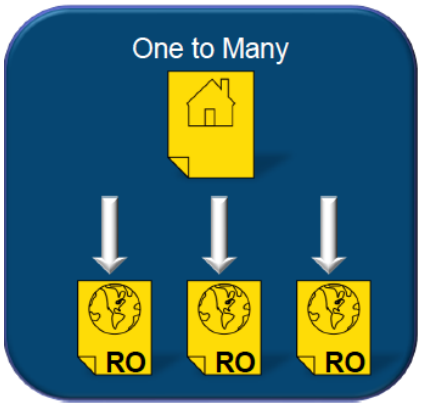


Active File Management Caching Basics

- Cache basics
  - Asynchronous updates
  - Writes can continue when the WAN is unavailable
  - TCP/IP for communication between sites (NFS or GPFS protocol)
- Two sides
  - Home - where the information lives
  - Cache
    - Data written to the cache is copied back to home as quickly as possible
    - Data is copied to the cache when requested
- Multiple caching
  - Read-Only
  - Single Writer
  - Independent writers (Cache-Wins)
  - Local updates

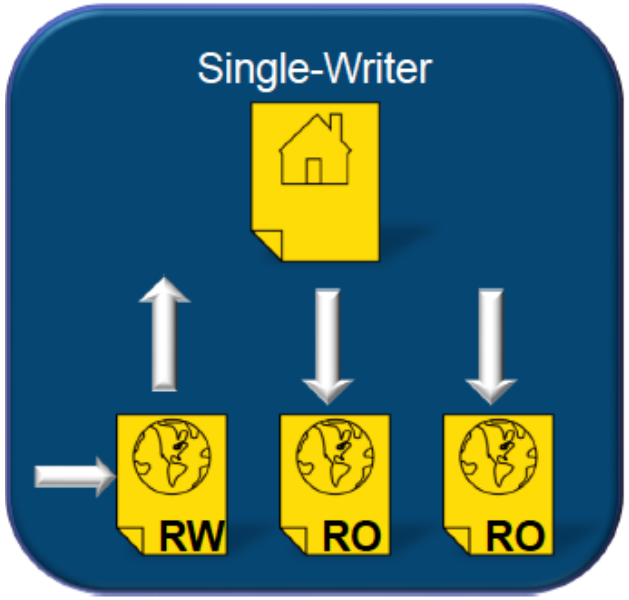
AFM Mode: Read-Only caching

- Read-only caching mode
  - Data exists on the home fileset and one or more cache sites
- Data is moved to the cache on-demand.
  - File Metadata caching: Listing the contents of a directory moves the file metadata information into the cache
  - Data – Opening a file copies the data in the cache
  - Getting data to the cache
    - On-demand when opened
    - Pre-fetch using a GPFS policy
    - Pre-fetch using a list of files
- Caching behavior
  - One to Many
  - Auto cleaning of cache
  - Cascading caches



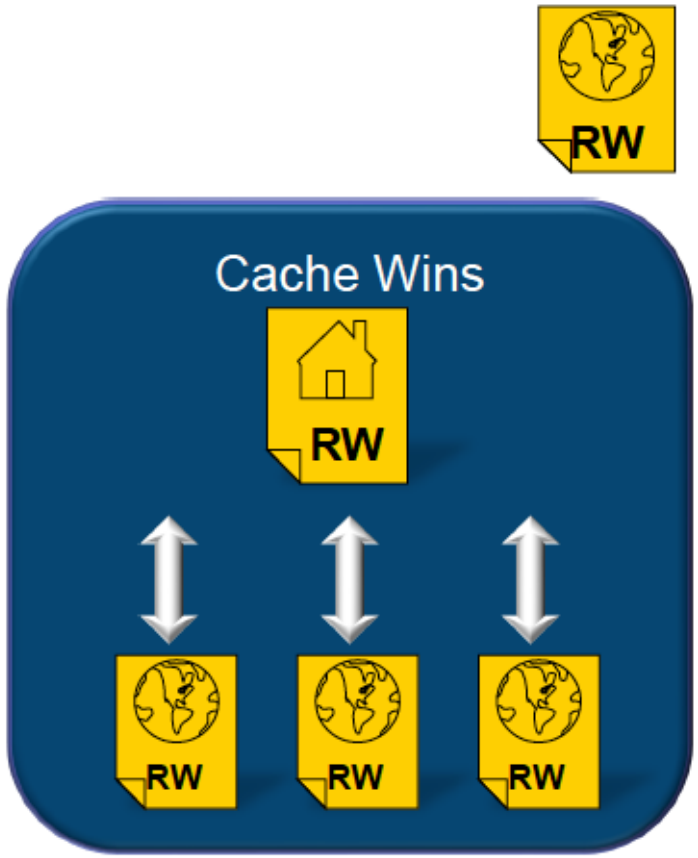
AFM Mode: Single-Writer

- Data written to a cache
- Asynchronous replication back to home
- Can have multiple read-only caches



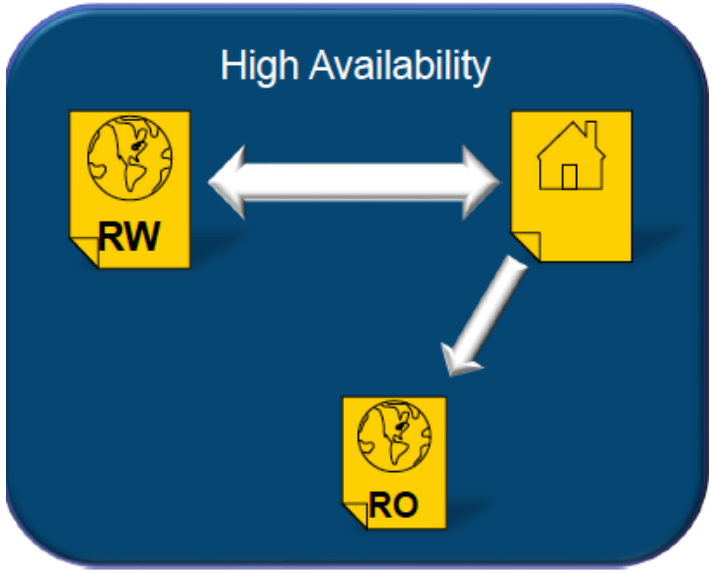
AFM Mode: Independent writers

- Multiple cache nodes
- All nodes can write data
- Conflict resolution
  - Default: The last writer wins



Asynchronous replication

- Asynchronous Replication in HA Pair
  - Cache site does the writing
  - Home site is failover
- Cache Fails
  - New cache can be defined
- Home Fails
  - New Home can be defined



Communication between AFM clusters

- Communication is done using NFSv3
  - Already tested with NFSv4 (in GPFS v.4.1)
  - Architecture is designed to support future protocols
- GPFS has it's own NFSv3 client
  - Automatic recovery in case of a communication failure
  - Parallel data transfers (even for a single file)
  - Transfers extended attributes and ACL's
- Additional Benefits
  - Standard protocol can leverage standard WAN accelerators
  - Any NFSv3 server can be a "Home"
  - Can be used as migration method from any NFS to GPFS (or between GPFS)

AFM Configuration example

- Setting up the Home cluster
  - NFS v3 server
    - recommended to use the GPFS cNFS
    - Should have “Cluster IP”
  - Define gateway nodes (for both home and cache)
    - Cache data is transferred between the GPFS clusters through gateway nodes

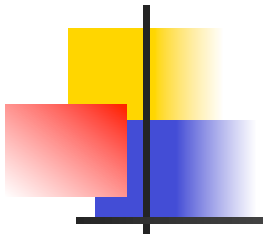
    mmchnode --gateway –N node1
  - Setting up a cache relationship
    - best practice to define the NFS mount points at fileset junction points
    - On the home:

    mmcrfileset master1 master_t1

    mmlinkfileset master1 master_t1 –J /gpfs/master1/master_t1

    #cat /etc/exports

    /gpfs/master1/master_t1 *(rw,no_root_squash,sync,fsid=92496)



AFM Configuration example (2)

- On the home (cont.):
  - Enable the exported path at home suitable for AFM:

```
mmafmconfig enable ExportPath
```

  - Start daemons

```
mmstartup -a
```

  - Start nfs services

```
/etc/init.d/nfs start
```

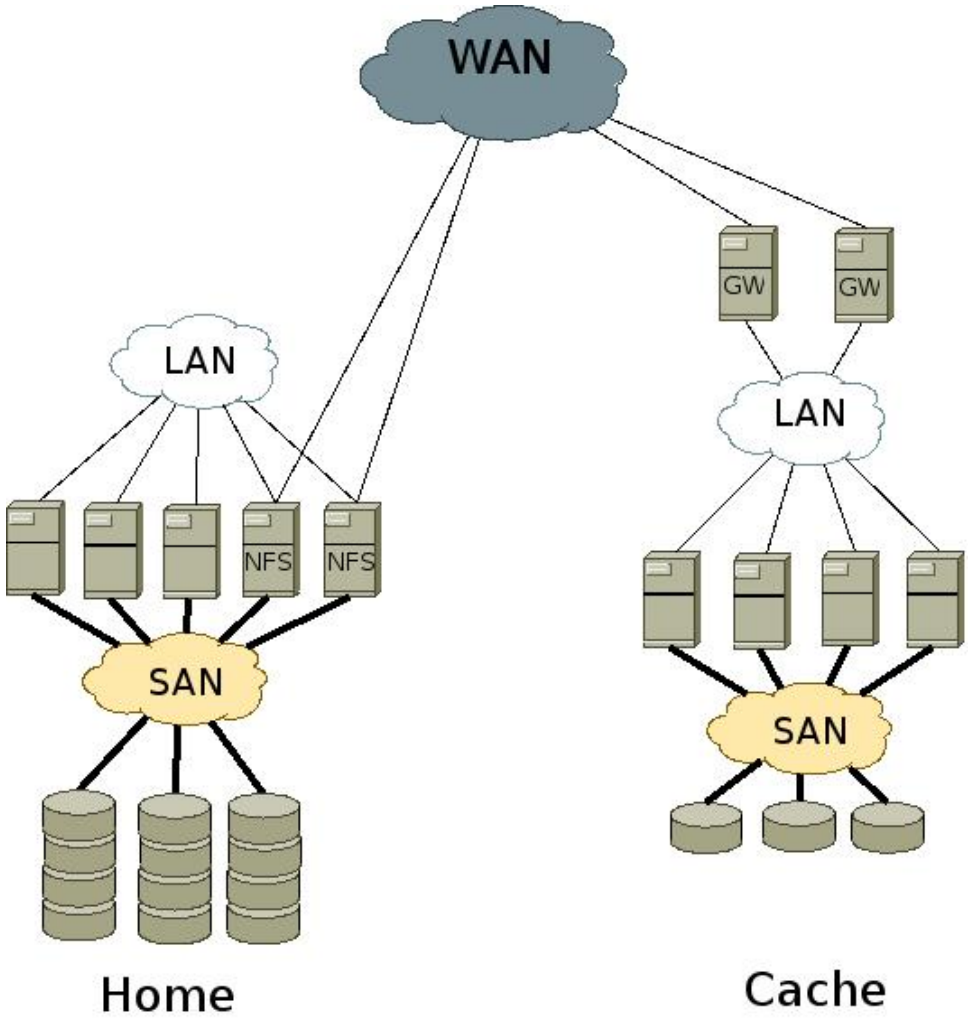
- On the cache:
  - create an independent fileset using –p parameter:

```
mmcrfileset c1fs master_t1 -p afmtarget=serv01:/gpfs_data/afm_home
    -p afmmode=ro --inode-space=new
mmlinkfileset c1fs master_t1 -J /c1fs/master_t1
```

  - Once the fileset is linked you are ready to start caching data

Example

- Transfer home → cache can happen in parallel within a node called a *gateway* or across multiple *gateway* nodes.

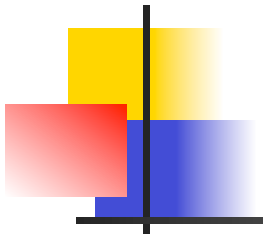


AFM management

- Commands: mmafmctl, mmafmlocal
- mmafmctl command can be used to control caching behavior, check the state of the cache and prefetch data.

mmafmctl Usage: mmafmctl Device {resync | expire | unexpire} -j FilesetName or mmafmctl Device {getstate | flushPending | resumeRequeued} [-j FilesetName] or mmafmctl Device failover -j FilesetName --new-target NewAfmTarget [-s LocalWorkDirectory] 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 ...]

Fileset Name	Fileset Target	Fileset State	Gateway Node	Queue State	Queue Length	Queue numExec
master_tl_ro	node1:/gpfs/tl	Active	node1	Active	0	1
master_tl	node1:/gpfs/tl	Active	node1	Active	0	348



Cache cleaning

- To enable cache cleaning enable a fileset soft quota for the cache fileset. You can enable quotas by using the –Q option to the mmcrfs or mmchfs commands. Cleaning starts when fileset usage reaches the soft quota limit.