FIFE Workshop 2017
CVMFS: the right way for software distribution, and maintenance tips

DAVE DYKSTRA

FIFE

FABRIC FOR FRONTIER EXPERIMENTS

CVMFS: the way to distribute grid software

- Directly mounted bluearc shared filesystems will be removed from Fermilab's GPGrid nodes in 2018
- All projects that want to use GPGrid and aren't yet using CVMFS will need to start using it for software or send all code with each job
- The projects that run grid jobs using daily builds should request a new separate CVMFS repository

Advantages of CVMFS

- Highly efficient use of caching for many readers
- Files that don't change between software releases are shared (de-duplication)
- Distributed to anywhere on the Open Science Grid as easily as GPGrid, and even beyond to the rest of the world
  - Content is securely verified
  - Cached at each site in web proxy caches
  - Cached on each worker node
- Can now be used as performant POSIX file catalog of data files in high speed storage (e.g. dCache)

Disadvantages of CVMFS

- It takes some time to publish files, depending on how many and how large files
- It takes additional time, typically up to an hour, to be available on all worker nodes
  - This time is expected to be reduced to about 15 minutes in the latter half of this year
- Files that are removed are not immediately removed from servers (repository servers and stratum 1s)
  - Optional daily garbage collection is however now quite robust

Two ways to publish to cvmfs

- Fermilab has our own cvmfs publishing machine, oasiscfs.fnal.gov (plus a backup)
- Projects that are large enough to be registered as an OSG VO have their own repository hosted here
- Smaller projects with just a few people share the fermilab.opensciencegrid.org repository
  - Only one project can publish at a time, so files are first stored in an intermediate area, then requests to publish are handled one at a time
  - A small update might need to wait quite a while for another to finish

News since my last FIFE workshop talk

- The cvmfs software has matured quite a bit with two major releases last year
  - For example, cvmfs-uptodate workaround no longer needed
- Number of repositories hosted at Fermilab has more than doubled to 21, plus artdaq added to fermilab repo
- osgstorage.org repositories have been created as POSIX interface to high speed data storage
  - All currently hosted at UNL, including one for nova
  - Most (not ligo) can be cached for other OSG sites with stashcache
    - Useful for partially reused data files, e.g. Genie flux files
  - The one for ligo limits reading of the data to jobs with authorized X.509 proxies

OSG CVMFS news of last 2 years

- One more egi.eu repository (snoplus) has been imported to OSG, plus a desy.de repository (ilc)
  - more can be added on request of an OSG VO
- Eight more OSG repositories have been exported to EGI, including fermilab, lsst, mu2e, nova, uboone, xenon, and singularity
- Additional stratum 1 at IHEP in Beijing for exported repositories
- Better monitoring for broken repository updates
- GOC operations has been quite stable

Maintaining a CVMFS repository

- Content of a repository is entirely the responsibility of the experiment
  - Assign management to a small number of knowledgable individuals
- The amount of space is not currently subjected to quotas, but the storage space is not limitless
  - Space is used on the repository servers and multiple stratum 1s
  - Only publish things that are used on worker nodes

CVMFS repository maintenance tips

- Set up .cvmfsdirtab with wildcards matching every software release directory for application and external packages
  - Avoids catalogs getting too large (keep under 200K files each)
  - Avoids loading info about files that will not be used
  - Avoids generating as much garbage when anything in a catalog changes
  - There's also an optional new feature to auto-generate catalogs
- To sync from another filesystem use cvmfs_rsync
  - Avoids subtle problem when removing old releases with catalogs
  - /grid/fermiapp/cvmfsfermilab/sbin/cvmfs_rsync
- Make sure all files are world readable

CVMFS repository maintenance tips

- Avoid data files that not similar sizes and access patterns as executable software
  - All jobs in a batch of jobs should generally access the same files
  - Typically the total read per job should be about a Gigabyte compressed or less
  - Larger amounts of data or randomly accessed different data should go into high bandwidth storage (dCache)
  - Tar files, etc, are better if they are unpacked
- Partially re-used data files can go in osgstorage.org repositories with data in dCache

CVMFS repository maintenance tips

- Also generally best to avoid source files in CVMFS
  - Does not affect client or squid performance, but it multiplies the number of small files on Stratum 1s which affect their performance
    - In the future, different implementations on Stratum 1s may mitigate this affect, but there are no specific plans to change this
  - Not a requirement, a best practice
  - Source required for compiling is fine, but avoid very rarely accessed source such as for debugging if possible
- If files are to be frequently removed, request to have garbage collection enabled

FIFE CVMFS Documentation

- https://cdcvs.fnal.gov/redmine/projects/fife/wiki/Introduction_to_FIFE_and_Component_Services#OASISCVMFS