Using CVMFS for User Analysis Code Distribution

Dave Dykstra, Fermilab

CernVM Workshop

3 June 2019

Background, motivation

- Fermilab used to have user-writable Network Attached Storage mounted on local worker nodes, and much User Analysis Code was run directly from there
  - Caused frequent meltdowns on the NAS server
  - Limited ability to run grid jobs elsewhere
- When that was disallowed, the recommendation was to download and unpack a tarball of code from high speed storage (dCache)
  - Caused a meltdown on many dCache servers as hundreds or thousands of jobs were reading the same large file
  - Tarballs ranged in size up to 3 GB
- Temporary solution was to put the tarballs in a dCache “resilient pool”, with 20 replicas of all the files
  - Works, but highly wasteful of resources
  - Especially wasteful when running on the grid, because the same tarball gets copied many times over the WAN

Considering CVMFS

- We knew CVMFS is extremely efficient for code distribution
- Software tends to have many files in common with previous versions
  - Files reused thanks to cvmfs deduplication
- Current standard publish + distribution delays are too long, but they could be reduced
- A proof-of-concept test using existing tarballs from dCache showed publish rate could be reasonably handled by one server
  - Showed bzip2 uncompress 5 times slower than gzip

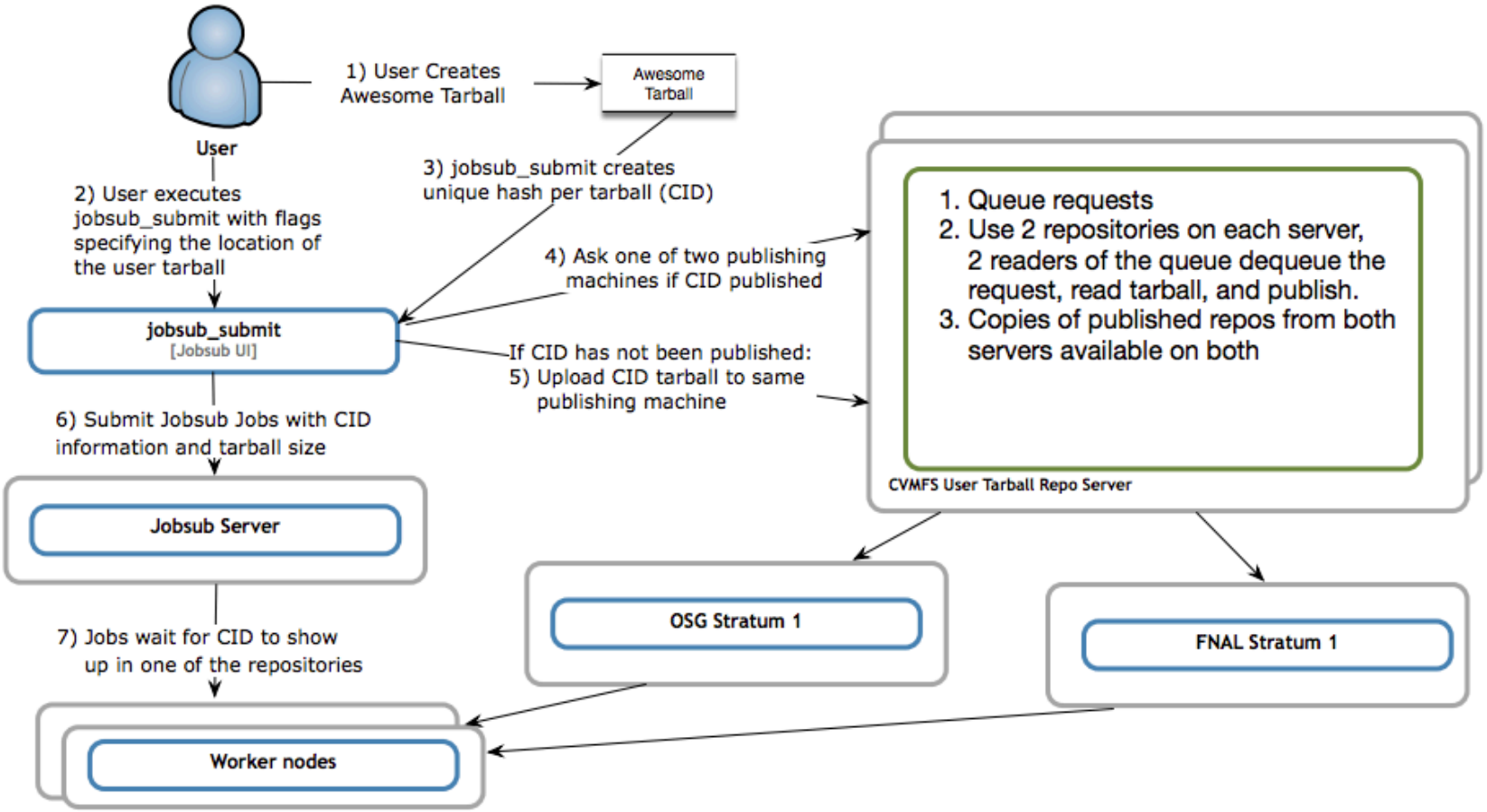
User code in CVMFS

- The primary CVMFS publication interface is not designed for use by large numbers of users
  - Expects small number of experts to maintain each repository
- We are building a system to publish user code tarballs in CVMFS
  - Integrating with our job submission system
  - Expect to begin production use by the end of June

System design

- Two publishing servers for redundancy
  - Provide web api
- Two repositories on each server so cleanups don't block publishing
- Tarballs given unique Code ID (CID) based on hash
- Job submission client directly uploads tarball, authenticated by X.509 proxy
- Tarballs are unpacked and published in CID directory
- Minimize distribution delays, less than 5 minutes
- Job wrapper waits (with a timeout) for CID to appear in any of the four repositories, passes the directory to job

Control flow



Server software packaging

- Most of the server software is packaged in a single rpm plus its dependencies
  - Intended to be able to deployed by multiple organizations
- Configuration is in a single simple file
  - Mainly just repository names needed
  - Some other standard system configuration needed such as grid-mapfile
- Creates repositories and replicas
- Provides https web apiDoes automatic cleanup

API

- /pubapi/publish?cid=XXX
  - Uses POST to upload tarball; easy to use with curl
  - Does queueing and publishing
  - Responds OK when queued or PRESENT if cid already existed, and updates a timestamp if present
  - CIDs assigned by client; api accepts any CID and may include slashes to group into subdirectories
    - Fermilab's client will create tarball first if given a directory
- /pubapi/exists?cid=XXX
  - Responds MISSING or PRESENT
- /pubapi/update?cid=XXX
  - Responds MISSING or PRESENT
  - Updates a timestamp if present

Repository cleanup

- Cleanups happen after a configurable number of days since last time a CID was used
- Timestamps for previously published CIDs are stored in any repository when CID reused
- Cleanups happen in one repository per hour starting at a configurable hour, followed by cvmfs garbage collection

Expedited updates

- TTL set to 15 seconds in repository config instead of usual 4 minutes
- Cache delay set to usual 61 seconds in Apache
- Stratum 1 checks for updates to these repositories twice per minute instead of usual once every 5 minutes
- Cvmfs client kernel cache flush takes the usual one minute
- Total delay for small updates should be less than 3 minutes

Status

- Still a work in progress, but code should be complete very soon
- Production servers (former worker nodes) are installed, and updates to FNAL stratum 1 are expedited
- Monitoring interface still to be completed
- User/Administrator documentation still to be written
- Going through new Fermilab process to be officially released as open source
- In future plan to use cvmfs_server tarball ingest, and the gateway feature for parallel publish so big publishes won't slow down small
- Current version at https://github.com/DrDaveD/cvmfs-user-pub
  but plan to move to https://github.com/cvmfs-contrib