

Rapid Turnaround User code in Cvmfs

Dave Dykstra

1 November 2018

Summary

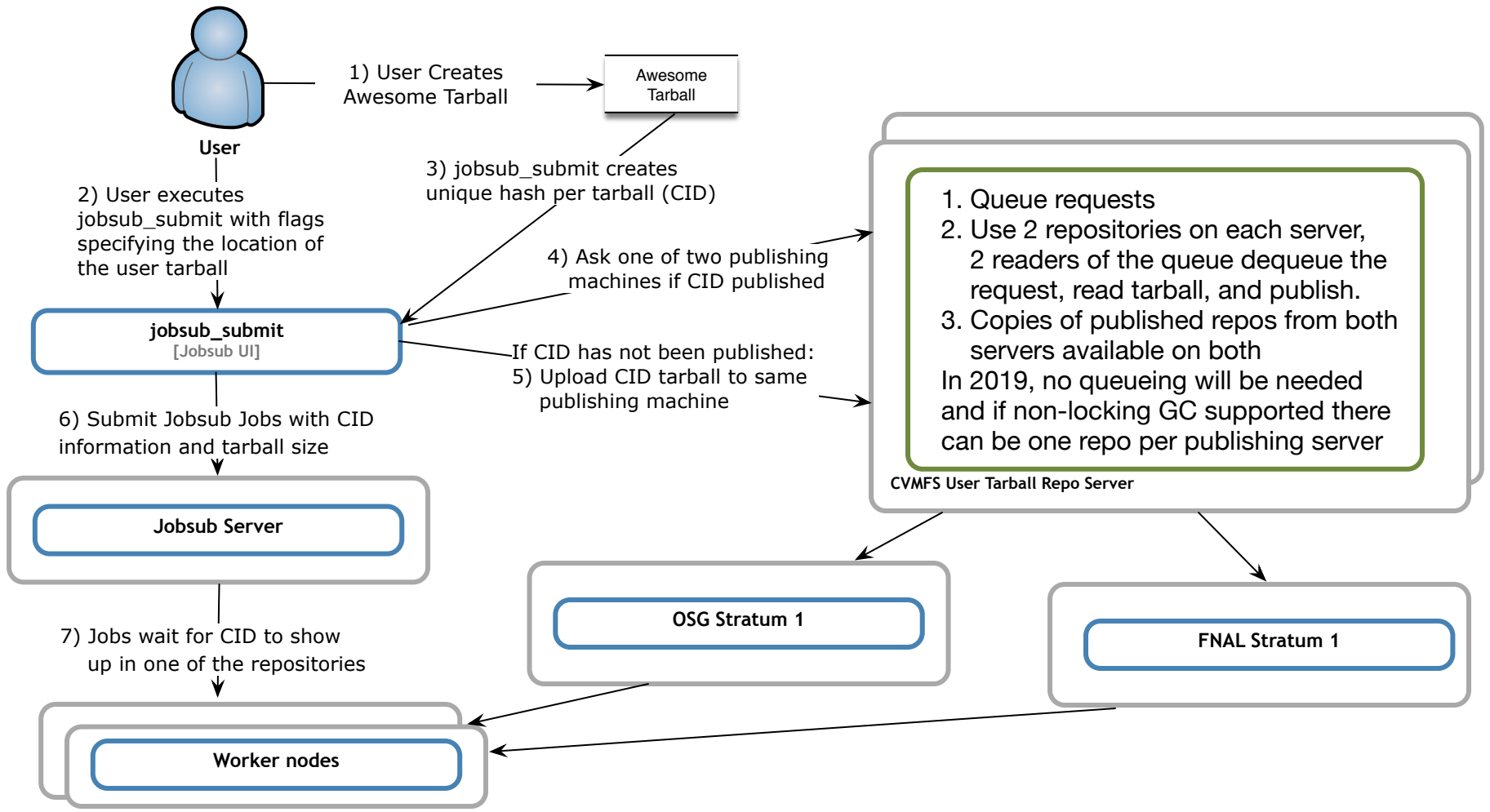
- The high level design of a system to quickly distribute FIFE user code in Cvmfs has been accepted
- A project plan needs to be finalized to estimate when it will be implemented
- Testing has been done to determine system requirements

User code publish testing results

- Test setup: poll 8 dCache dirs for user code tarballs, publish to 2 repos
- Current usage extremely low
 - Average of less than 8 tarballs per day
 - A little over 1 GB total per day, average size 150MB
 - Minimum seen 3 KB, maximum 1.7 GB
 - Never seen 2 publishes happening simultaneously
- Average time to complete publish 72 seconds
 - Range 9 seconds to 209 seconds (maximum was 834MB)
 - Includes 15-30 seconds to poll each dCache directory
 - Longest times can be cut significantly by switching to gzip instead of bzip2, with 1.7GB cutting from 5 minutes to 35 seconds
- Garbage collection run daily, publish disk usage very low (45GB)

Accepted design summary

- Jobsub client sends new user code tarballs to one of a pair of dedicated cvmfs publishing servers, for reliability
- Configure publishing servers to accept authentication from all known FIFE DNs
- Publish quickly as possible, set delay limits low as possible
- Jobsub job wrapper waits for code to show up in cvmfs if not there yet, up to some maximum time
- Clean up using same algorithm now used to clean up dCache



Transfer delay can be small

- Configure FNAL stratum 1 to bypass OSG stratum 1 for these repos, have separate cron check every minute
- OSG stratum 1 already checks all repositories in parallel about every 20 seconds
- BNL stratum 1 might be willing to speed these repos up too if we asked
- Expiration for squid caching already set to 61 seconds
- Configure client expiration on publishing server, can set very low such as 30 seconds
- Reasonable maximum: 5 minutes, average much less

Implementation plan summary

- Dave make an rpm with everything needed on the publishing servers, reading from a config file
- Ed's group make servers available
- Tony's group (Hyunwoo)
 - make puppet to install config file, rpm, and grid-mapfile
 - export repositories to OSG
 - tune stratum 1 to frequently update these repositories
- Dennis update JobSub
 - Client to contact publishing servers, upload tarballs
 - Job wrapper to wait for code to become available
- Shreyas adapt dCache tarball cleanup algorithm to Cvmfs