Dynamic Installation of CVMFS using Glideins

Namratha Urs, Marco Mambelli

HTCondor Week 2022, Madison, WI

May 24, 2022

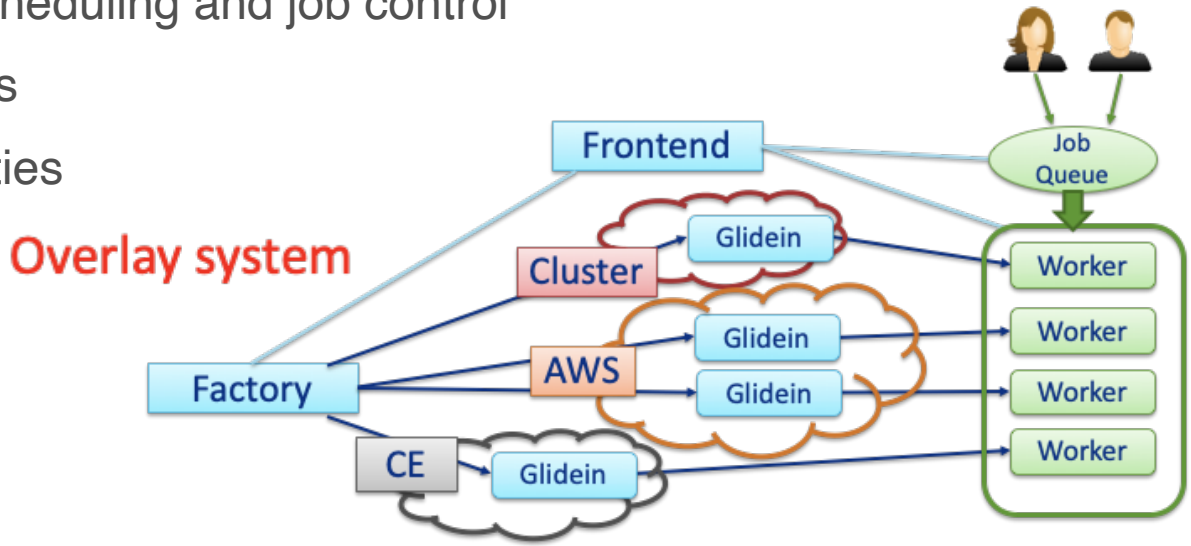
High Throughput Computing (HTC) for High Energy Physics

- Many computing resources used over long periods of time to accomplish a computational task
- Growing needs
  - Scalability
  - Accessibility
  - Simplified management
- Trends of increased heterogeneity
  - Multiple organizations
  - Different systems
  - Less standard infrastructure
  - Different authentications

Increasing complexity ⇒ More difficult for scientists!

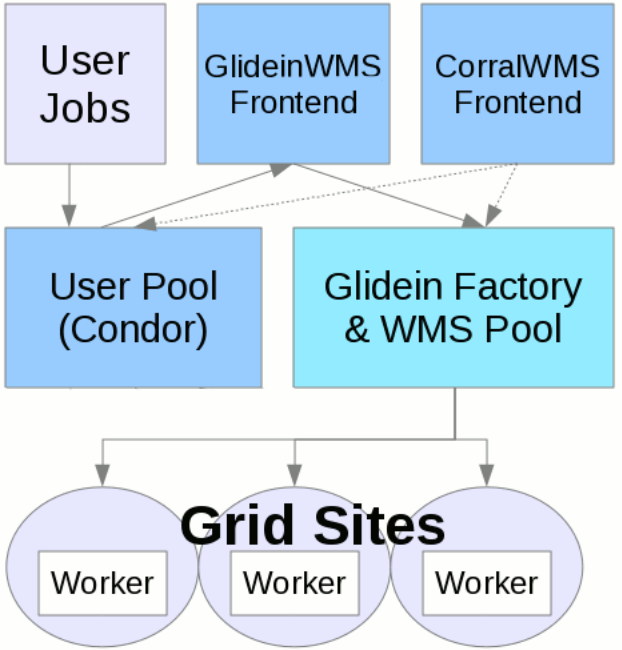
GlideinWMS

- Simplifies resource provisioning for distributed high throughput computing
- Provides reliable and uniform virtual clusters
- Glideins submitted to heterogenous resources not tested by VO
- Leverages HTCondor for scheduling and job control
  - Provides HTCondor pools
  - Uses HTCondor capabilities



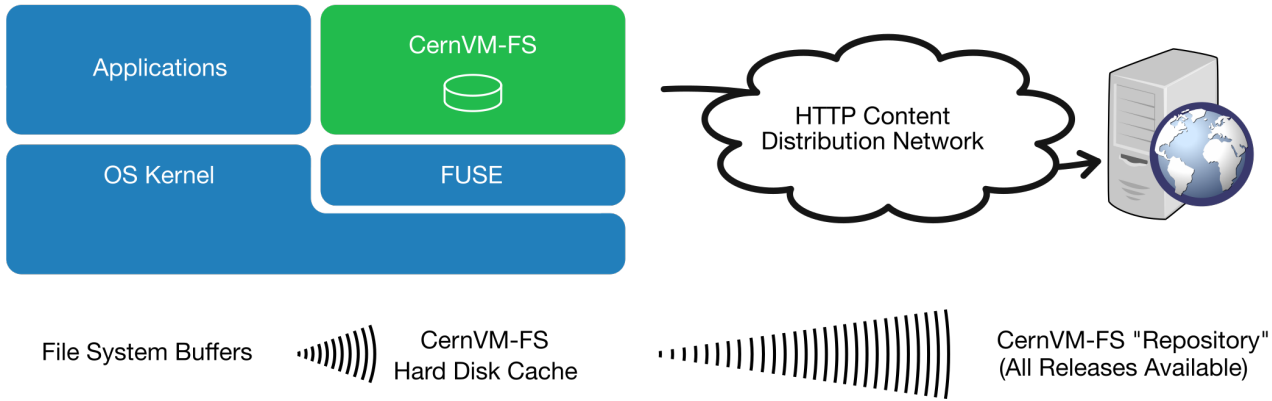
GlideinWMS: Components

- **Frontend:** look for user jobs and request the Factory to provide glideins
- **Factory:** self-advertise, listen for requests from Frontend and submit glideins
- **Glideins (pilots):** provide a customized execution environment for user jobs



Cern-VM File System (CVMFS)

- **Read-only**, globally distributed file system based on HTTP
- Optimized to distribute and deploy scientific software/data to nodes
- Scalable, reliable and low-maintenance software distribution service



Graphic Credits: https://cvmfs.readthedocs.io/en/stable/

Difficult Access to Data

- Abundance of computing resources (“sites”) for HEP
- CVMFS used in collaborations within the particle physics community
- Some HPC sites may not provide a local installation of CVMFS
  - Requires effort by system admins to install and maintain CVMFS locally
  - Limits the use of HPC resources in the absence of CVMFS

Desired Outcome

- Ability to use CVMFS even when locally unavailable
- Provision CVMFS with minimal human intervention

The Solution

- Provision CVMFS on demand using GlideinWMS
  - Extend functionality of Glidein to install CVMFS when unavailable on the site
  - Testing and configuration of nodes by Glidein - just one more task!
- Provision CVMFS without a system wide installation using *cvmfsexec*[†]
  - Leverage unprivileged user namespaces and FUSE interface
  - Perform installation in unprivileged mode since Glidein has no special privileges

[†] Blomer, Jakob, Dave Dykstra, Gerardo Ganis, Simone Mosciatti, and Jan Priessnitz. "A fully unprivileged CernVM-FS." In *EPJ Web of Conferences*, vol. 245, p. 07012. EDP Sciences, 2020.

cvmfsexec

- Package support for unprivileged CVMFS
  - Relies on unprivileged user namespaces and FUSE configurations
- Allows creating distributions with CVMFS software and configuration
  - Using latest cvmfs and configuration rpms
- Supports easy sharing with other users or many machines
  - Self-contained distribution as a single file
- Four ways (modes) of mounting CVMFS as a non-root (unprivileged) user

Using cvmfsexec

- Modes distinguished based on system configurations
  - Unprivileged user namespaces supported
  - Unprivileged user namespaces enabled
  - fusermount (FUSE) available
- *cvmfsexec* better for newer kernels (>=RHEL 7.8 or RHEL 8)
  - Older kernels (<= RHEL 7.7) do not clean up the mounts — requires explicit unmounting with *umountrepo*
- Current solution uses *mountrepo*/*umountrepo*
  - Also caters to older kernels
  - *cvmfsexec* requires slightly advanced handling

Mode 1
mountrepo/ umountrepo

Mode 2 and 3
cvmfsexec

Mode 4
singcvmfs

* https://github.com/cvmfs/cvmfsexec

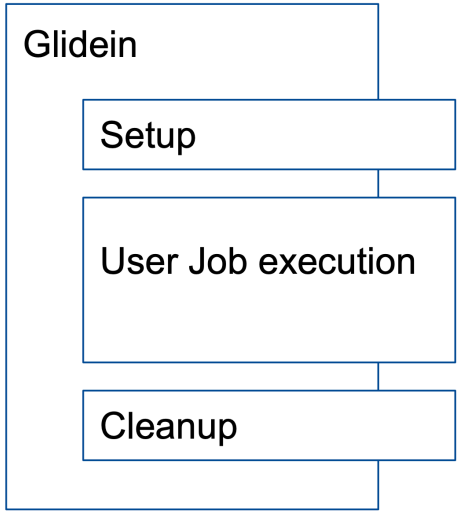
Fermilab

From a User/Operator Standpoint

- GlideinWMS installed and configured
  - >= 3.7.6 (Python 2)
  - >= 3.9.4 (Python 3)
- Three attributes in Factory configuration file
  - **GLIDEIN_USE_CVMFSEXEC** — whether cvmfsexec is used for provisioning CVMFS if not locally available (`1`/`0`)
  - **CVMFS_SRC** — enables selection of CVMFS repos based on the source, i.e., config repository (`osg`, `egi` or `default`)
  - **GLIDEIN_CVMFS** — specify whether CVMFS should be installed on a resource [or `Entry`] (`required`, `preferred`, `optional` or `never`)

Under The Hood

- During initial setup
  - Check OS (platform and kernel information)
  - Check support for unprivileged user namespaces
  - Check FUSE (fuse/fusermount installed, user in fuse group)
  - Download cvmfsexec distribution based on worker node specifics
  - Decide best option between *cvmfsexec* and *mountrepo* (given the OS and platform configuration)
    - Mounted on `/cvmfs` if possible
- During singularity startup (job wrapper)
  - Bind mount the mount directory to `/cvmfs`
- During cleanup
  - Unmount CVMFS if mounted



Creating Possible CVMFS Distributions

- Two parameters
  - Platform (OS + hardware architecture) — `rhel6`, `rhel7`, `rhel8`, `suse15`
  - source of latest rpms — `osg`, `egi`, `default`
- Large number of distributions for possible combinations
- Dynamic creation of distributions during factory reconfiguration/upgrade
  - Version-based selective rebuilding of distributions
- Added to the default list of uploads for use by the Glidein

Selecting the CVMFS Distribution for the Worker Node

- Occurs at the time of worker node customization by the Glidein
- Appropriate distribution file selected based on the specs of the worker node
- Reduces the number of distributions to be shipped (ONE versus many)

Next Steps

- Extend to support mode 3 of cvmfsexec
  - *cvmfsexec* command handles cleanup of mounted CVMFS repositories when processes are hard-killed
  - *cvmfsexec* spawns a sub-process in a namespace unshared from the parent process
    - Find a way to make the Glidein configuration variables visible inside the new namespace
- Test on RHEL8 and SuSE platforms
- Improve existing error handling behavior for extreme edge cases
  - errors encountered during mounting of CVMFS (`required`, `preferred`, `optional` or `never`)

Benefits

- Lower overhead for site administrators
  - Less software to install!
- Easy code and data access for scientists
  - End-user jobs run in desired container image and access software/data distributed by CVMFS
- Improved flexibility to use new resources (e.g. HPC resources)
  - Allows GlideinWMS pilots to support HPC sites

Acknowledgements

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Thanks to Dave Dykstra for addressing our questions and providing clarifications regarding cvmfsexec and for helping with our requests for new features.

Thanks to Marco Mambelli for his direction and guidance on this work.

Thank You!

Open for Q&A and/or discussions

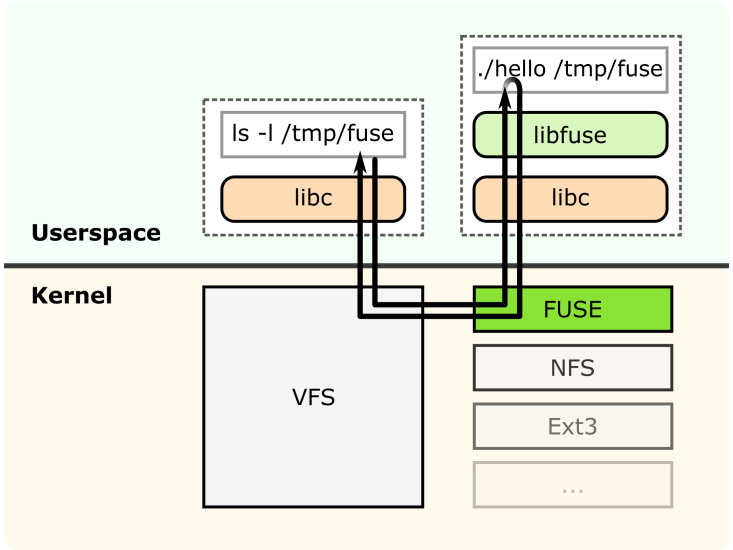
Appendix

Unprivileged User Namespaces

- Namespaces isolate global system resources between independent sets of processes
  - User namespaces: created by regular users; used by unprivileged processes to access privileged capabilities
  - Limits the scope of that privilege to the user's namespace
  - Can allow a process to create namespaces and to mount a filesystem for all the processes in the same namespace
  - More modern

Filesystem in Userspace (FUSE)

- Framework that allows secure, non-privileged mounts without modifying kernel code
  - Bridge to the actual kernel interfaces
  - Kernel module + userspace library + mount utility (`fusermount`)
  - An opportunity to use filesystems!
  - More available



Credit: https://en.wikipedia.org/wiki/Filesystem_in_Userspace