On-Demand Provisioning of CernVM File System with GlideinWMS

Namratha Urs
2021 Fermilab Computational Science Internship (FCSI)
Supervisor: Marco Mambelli
18 August 2021

Problem

- Some sites (HPC resources) may not provide a local installation of CernVM File System (CVMFS)
- Make CVMFS available on HPC sites when a local installation is unavailable
- Minimize the effort required by site administrators to install CVMFS locally

Solution

- **On-demand provisioning of CVMFS using GlideinWMS**

- Extend glidein functionality to install CVMFS if not available on the node
  - Ship with the glidein a tool to provision CVMFS and use if needed
  - Select the most reliable option to enable CVMFS given the worker node setup
  - GlideinWMS sends glideins to test and setup nodes; becomes one more task

- Perform CVMFS installation in unprivileged mode
  - Required since glideins have no special privileges
  - Leverage unprivileged user namespaces and FUSE interface
  - Utilize `cvmfsexec` tool for CVMFS provisioning without a system wide installation

Why cvmfsexec?

- Package support for unprivileged CVMFS (sysadmin not required!)
  - Relies on unprivileged user namespaces and FUSE (Filesystem in Userspace) configurations
- Creates distribution with CVMFS software and configuration
  - Allows custom CVMFS configuration settings
- Self-contained distribution as a single file
  - Easy sharing with other users or to many machines
  - `singcvmfs` distributions (mounts CVMFS repositories inside a container)
- Access to one of the developers led to
  - Further understanding of the software
  - Request and implementation of new features — Thank you, Dave!
- **Four ways (modes)** to mount CVMFS as a non-root (unprivileged) user
  - More details about these modes at https://www.github.com/cvmfs/cvmfsexec

Using cvmfsexec

- Commands
  - `mountrepo/umountrepo` (mode 1)
  - `cvmfsexec` (modes 2 and 3 only)
  - `singcvmfs` (mode 4) — not considered yet
- Modes distinguished based on system configurations:
  - unprivileged user namespaces supported
  - unprivileged user namespaces enabled
  - fusermount (FUSE) available
- `cvmfsexec` works better on newer kernels (>= RHEL 7.8 or RHEL 8)
  - Older kernels (<= RHEL 7.7) do not clean up the mounts; requires explicit un-mounting with `umountrepo`
- Using only **`mountrepo/umountrepo`** commands as of now
  - Cater also to older kernels
  - Albeit simple, using `cvmfsexec` command requires slightly advanced handling — creation of a subprocess and its environment
    - Looking at using `cvmfsexec` in future release

Feasibility Study (Summer 2020)

- Parameters
  - platform
  - unprivileged user namespaces supported
  - unprivileged user namespaces enabled
  - FUSE installed, *fusermount* available
  - user is in fuse group

- Understand mount/unmount behavior using `cvmfsexec`
- Test on platforms and validate against expected behavior

CVMFS Testing Matrix

unprivileged user namespaces supported? (via sysctl)	unprivileged user namespaces enabled? (via unshare)	fuse installed?	fusermount available?	user in 'fuse' group?	cvmfsexec works?	mountrepo/umountrepo works?	test remarks for mountrepo usage
Yes	Yes	No	No	No	Yes	No	requires fuse/fusermount
Yes	Yes	No	No	Yes	Yes	No	requires fuse/fusermount
Yes	Yes	No	Yes	No	Yes	No	inconsistent FUSE config
Yes	Yes	No	Yes	Yes	Yes	No	inconsistent FUSE config
Yes	Yes	Yes	No	No	Yes	No	inconsistent FUSE config
Yes	Yes	Yes	No	Yes	Yes	No	inconsistent FUSE config
Yes	Yes	Yes	Yes	No	Yes	Yes	works even though the user is not in fuse group
Yes	Yes	Yes	Yes	Yes	Yes	Yes	works
Yes	No	No	No	No	No	No	neither cvmfsexec nor mountrepo works (error related to unshare)
Yes	No	No	No	Yes	No	No	neither cvmfsexec nor mountrepo works (error related to unshare)
Yes	No	No	Yes	No	No	No	inconsistent FUSE config
Yes	No	No	Yes	Yes	No	No	inconsistent FUSE config
Yes	No	Yes	No	No	No	No	inconsistent FUSE config
Yes	No	Yes	No	Yes	No	No	inconsistent FUSE config
Yes	No	Yes	Yes	No	No	Yes	mountrepo works (even though the user is not in fuse group)
Yes	No	Yes	Yes	Yes	No	Yes	mountrepo works
No	No	No	No	No	No	No	neither cvmfsexec nor mountrepo works (failed to exec fusermount)
No	No	No	No	Yes	No	No	neither cvmfsexec nor mountrepo works (failed to exec fusermount)
No	No	No	Yes	No	No	No	inconsistent FUSE config
No	No	No	Yes	Yes	No	No	inconsistent FUSE config
No	No	Yes	No	No	No	No	neither cvmfsexec nor mountrepo works (permission denied error)
No	No	Yes	No	Yes	No	No	inconsistent FUSE config
No	No	Yes	Yes	No	No	No	inconsistent FUSE config
No	No	Yes	Yes	Yes	Yes	Yes	mountrepo works

Prototyping (Summer 2020)

- Bash-compliant shell scripts
  - **`cvmfs_helper_funcs.sh`**
  - **`cvmfs_mount.sh`**
  - **`cvmfs_umount.sh`**
- `INFO/WARN/ERROR` messages to improve output/error/debug messages
- Inline documentation to aid code readability
- Code modularization, standard logging mechanism
- Addition of unit tests using BATS to ensure code quality

Design

System checks
- *platform* (rhel7, rhel6, rhel8, centos, other)
- *kernel* info (2.x, 3.x, 4.x, other)
- *unprivileged user namespaces* supported
- *unprivileged user namespaces* enabled
- *FUSE installed*
- *fusermount available* and *user is in fuse group*

CVMFS detection and mounting
- Check for locally installed CVMFS
- Mount CVMFS using `cvmfsexec` package when not locally available

Unmount previously mounted CVMFS when glidein terminates/expires

Working Feature (Summer 2020)

1. Configured a **custom script** using the parameters in the glidein config file
   - **`cvmfs_setup.sh`** — imports helper functions and invokes the CVMFS mount script
2. Manually created **tarball** containing auxiliary files
   - **`cvmfs_utils.tar.gz`** — contains
     - **`cvmfs_distros.tar.gz`**: Utilities to mount/unmount CVMFS: platform- and architecture-specific distributions
     - **The three scripts**: (a) helper functions, (b) mount script and (c) unmount script

- Added (1) and (2) to the Factory configuration file (via *<files>*) to ship to the glidein-customized node
- Large number of distributions created for various combinations of platform- and architecture-specifications
  - Extra level of granularity with `osg` and `egi` configuration repositories for CVMFS

Integration with the GlideinWMS codebase (Summer 2021)

- Added the custom script and the tarball to the default list of uploads
- Added three attributes to the Factory config:
  - **`CVMFS_SRC`** — enables selection of CVMFS repos based on the source, i.e. config repository (`osg`, `egi` or `default`)
  - **`GLIDEIN_CVMFS`** — for better error handling behavior in case of errors encountered during mounting of CVMFS (`required`, `preferred`, `optional` or `never`)
  - **`GLIDEIN_USE_CVMFSEXEC`** — whether the tarball should be unpacked (`1`) or not (`0`)
- Patch fix for correct execution of cleanup script at glidein termination/ expiration (#25981)
- Incorporated logic for un-mounting CVMFS with additional logic for locally installed vs. glidein-based CVMFS
- Use of error_gen.sh for reporting success and failure messages during the the execution flow

- Feature is being released in GlideinWMS v3.7.5

Supporting Big Files in GlideinWMS

- Tarball with utilities and helper scripts for CVMFS provisioning — BIG!
  - Need for alternative solution to store big files
- Considered Git-LFS and Git-Annex which are possible ways of implementing version control for large files
  - Our big file requires no versioning — the latest version is what would be used
- Developed symbolic link-based solution (by Marco Mambelli)
  - Added a new directory *bigfiles* to the codebase; no content tracking
  - Hosted the tarball on the glideinwms website
  - Added scripts to upload and download the files to *bigfiles*
  - Used symbolic links to access the downloaded “big” file

Dynamic creation/selection of platform-specific distribution

- `cvmfs_utils.tar.gz = cvmfs_distros.tar.gz + a few scripts`
  - `cvmfs_distros.tar.gz` — **static tar file** containing multiple cvmfsexec distribution files (corresponding to a CVMFS source, system platform and architecture)
  - `cvmfs_utils.tar.gz` file transported to the glidein — **inefficient** as only one distribution file is needed for customizing the worker node
- **`generate_cvmfsexec_distros.sh`** — automatically generate all possible combinations of CVMFS source and platform-specific distribution files (packaged as individual tar files)
  - Invoked at the time of reconfig/upgrade (more dynamic)
- Modified GlideinWMS code to add the generated distributions (as tar files) to the default list of uploads at the time of factory reconfiguration/upgrade
- **`cvmfsexec_platform_select.sh`** — automatically selects the appropriate distribution tar file based on the specifics of the worker node
  - Invoked during worker node customization by the glidein
  - Reduces the number of distributions that are shipped as artifacts (ONE versus many)

Code reorganization

- Tarball only contains the scripts at this point; no necessity to package the scripts as a tarball
- Modularize the entire code for CVMFS provisioning such that one script serves as both an executable and a source file
  - Instead of having separate scripts with helper functions, functions for mount and un-mounting of CVMFS

What's Next?

- Enable glideins to use mode 3 of the `cvmfsexec` tool (#26095)
  - `cvmfsexec` command can handle clean unmounting of CVMFS repositories; even when the processes are hard-killed (kill -9)
  - `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
- Version-based selective updates for cvmfsexec distribution files (#26153)
  - How to determine if re-downloading and rebuilding the cvmfsexec distribution files is needed on subsequent reconfig/upgrade of the factory
- Error reporting for cleanup scripts (#26093)
  - No reporting information found in client logs in the Factory from the cleanup procedure
- Test on RHEL8 and SuSE platforms

References

- GlideinWMS Documentation
- https://glideinwms.fnal.gov/presentations/intro/GlideinWMS.pdf
- CernVM File System Docs
- cvmfsexec - https://www.github.com/cvmfs/cvmfsexec
- Unprivileged User Namespaces - https://lwn.net/Articles/532593/
- FUSE - https://www.kernel.org/doc/html/latest/filesystems/fuse.html
- Project codebase - https://www.github.com/namrathaurs/gwms-cvmfs

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.