EESSI meeting

October 1st 2020

https://github.com/EESSI/meetings/wiki

Agenda

1. Short introduction by new people
2. Meeting with CernVM-FS developers (Sept 18th) [Bob]
3. Testing of 2020.08 version of pilot repository [Alan, Thomas, Jaco?]
4. Progress update
    a. Filesystem layer [Bob]
    b. Compatibility layer [Peter]
    c. Software layer [Kenneth]
5. NESSI [Thomas]
6. Potential sponsorship by Microsoft via Azure [Davide]
7. Next steps
8. Q & A

Short introduction by new people

New people on the call: feel free to quickly introduce yourself!

- Who are you, where do you work, on what?
- Why are you interested in the EESSI project?
- Are you planning to actively contribute,
  and if so, to which aspect(s) of the project?

2nd meeting with CernVM-FS devs

- Sept 18th, attended by people from EESSI and CERN (20)
- Topics:
  - Software installation practices @ CERN
  - Different approaches for adding software to CernVM-FS repos
  - Use of Spack for building software in Key4HEP project
  - How EESSI is building software in pilot setup (with our Singularity container)
  - Future/development releases of CVMFS
  - `cvmfs push` and `cvmfs enter` (upcoming features)
- Meeting notes available at

  https://github.com/EESSI/meetings/wiki/meeting-CernVM-FS-Sept-18th-2020

[Bob]

Testing of 2020.08 version of pilot repository

- Tested at JSC: no CernVM-FS, no external access on compute nodes
  - Everything done through Singularity
- Complication is running MPI codes (requires multiple mounts per node)
  - CernVM-FS `lib` and `run` directories need to be unique: use `--scratch` option
  - Alien cache can solve these problems (pre-populated if necessary)
  - Script in open issue: github.com/EESSI/filesystem-layer/issues/37
- GROMACS performance tested
  - Haswell + GROMACS 2020.1 (EESSI) vs. Skylake + GROMACS 2020.3 (native)
  - Single node: 43 ns/day vs. 51 ns/day (higher is better)
  - Two nodes: 54 ns/day vs. 95 ns/day (with TCP-only OpenMPI, so no surprise)

[Alan, Kenneth]

Testing of 2020.08 version of pilot repository

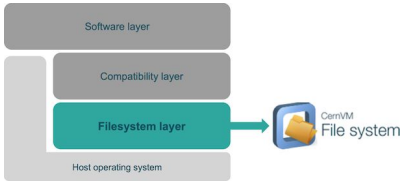
- Tested on HPC and IaaS platform in Norway
  - Only frontend nodes used so far (but proxy in place on one HPC)
  - Worked fine when followed the pilot instructions (bind mounts)
  - Only saw known issues (`$LD_LIBRARY_PATH`)
  - OpenFOAM test script runs smoothly on EESSI stack
  - Didn't get OpenFOAM test script to run on native EB stack
- Wish to have more test/benchmarking scripts which are compatible with system EB stack, to see differences in performance
- Tried to use pilot on Kubernetes based platform, not yet done...

[Thomas]

More testing of pilot repo (2020.08)

- Jaco did some testing in Windows Subsystem for Linux (WSL)
- Kenneth did some quick testing in Azure cloud (using 2020.09)
  - Running OpenFOAM test on a single-core Broadwell CentOS 8 VM
  - Works like a charm!
- **Any more experiences with testing of 2020.08 version of pilot repo?**

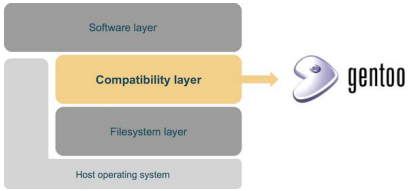
Progress update: filesystem layer



- Improved functionality for building `cvmfs-config-eessi` packages
  - Built for each PR/push, stored as build artifacts
  - Also stored as release assets for each tag: can be easily fetched from GitHub
  - Old packages on the Stratum 0 server will be removed this week!
- Singularity recipe for a Squid proxy
  - Can be run on, for instance, a login node; no privileges required
- Improved `.cvmfsdirtab` for automatically generating CVMFS catalogs
- No solution for the `fuse-overlayfs` issue yet… still using the old version
- More automation needed for adding software/files into repo

[Bob]

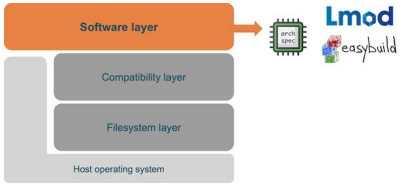
Progress update: compatibility layer



- Compatibility layer for 2020.09 pilot rebuilt from scratch
  - Portage snapshots for making bootstrapping of Gentoo Prefix (more) reproducible
  - Python 2 has been removed from compatibility layer
  - Compatibility layer is now built for both `x86_64` and `aarch64` (Arm 64-bit)
  - Symlinks to host files: use host's user/group database (`/etc/group`, `/etc/passwd`)
  - Locale now configured with Ansible to `en_US.UTF-8` (to fix problem with R)
- Improved documentation in README of compatibility-layer repo
- Github Action for testing the installation playbook in CI
  (everything except Prefix installation, which takes too long)

[Peter]

Progress update: software layer (1)



Script to install software in EESSI pilot repository

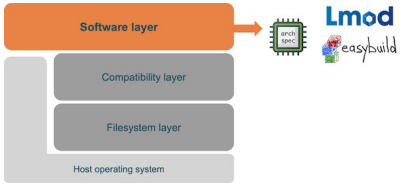
- See https://github.com/EESSI/software-layer/blob/master/EESSI-pilot-2020.09.sh
- Ugly bash script, not what we have in mind for the long term…
- Uses EasyBuild v4.3.0, but also pulls down stuff from pull requests (EasyBuild v4.3.1)
- Was very useful to facilitate others to install software in 2020.9 version of pilot repo
- Just a starting point, we can do a lot better!

Enhanced init script

- See https://github.com/EESSI/software-layer/blob/master/init/bash-pilot-2020.09
- Leverages `archspec` to automatically pick best target for current host
  - Even when there's no exact match!
- Also does fallback to `<arch>/generic` if it's there and no compatible target is available

[Kenneth]

Progress update: software layer (2)



Fixed problems in 2020.08 version of pilot software stack in 2020.09 update

- `$LD_LIBRARY_PATH` is no longer set by environment module files (wasn't needed)
- OpenMPI is now properly configured (depends on libevent, libfabric, PMIx, UCX)
- Some (build) dependencies are now leveraged from compatibility layer
  - Autoconf, Automake, binutils, bzip2, gettext, libtool, M4, ncurses, libreadline, XZ, zlib
  - Ran into some software installations problems because of this...
    - zlib in Gentoo is patched: `OF` macro was renamed to `_Z_OF` (broke Qt5)
    - ncurses library is split in libncurses.so + libtinfo.so (but this is optional?)
    - `gcc` provided by `GCCcore` module picks up `ld.gold` from host OS when compiling outside of Prefix environment
  - To discuss further, see https://github.com/EESSI/software-layer/issues/23

[Kenneth]

EESSI pilot repository

NOT FOR PRODUCTION USE!

2020.09 version of pilot software stack

- Software:
  - `foss/2020a` toolchain (GCC 9.3, OpenMPI 4.0.3, OpenBLAS 0.3.9, FFTW 3.3.8)
  - GROMACS 2020.1
  - OpenFOAM (two variants: version 8 and v2006)
  - R 4.0.0 (incl. ~800 R packages as extensions)
  - Bioconductor 3.11 (bundle of 262 R packages)
- Targets:
  - `aarch64/generic`
  - `x86_64/generic`
  - `x86_64/intel/{ivybridge,haswell,broadwell,skylake_avx512,cascadelake}`
  - `x86_64/amd/zen2`

Updated documentation at https://eessi.github.io/docs/pilot

[Kenneth]

Testing of 2020.09 version of pilot repository

- Tested `software/arch64/generic` on a Raspberry Pi 4
- Software tree compiled by Terje
  - Using `eb --optarch=GENERIC`
  - Build host with unidentified Arm64 CPU
  - (does not include Bioconductor yet)
- Using 64-bit kernel for Raspbian 10
- Singularity compiled from source using 32-bit Go binaries
- Works like a charm!
  - OpenFOAM test completed in ~1.5 hours on 4 cores (vs ~10 min on 2 x86_64 cores)
  - GROMACS on 4 cores: ~0.24 ns/day (vs ~5.24 on 4 broadwell cores)

[Bob]

Norwegian Environment for Scientific Software Installations (NESSI)

- Pilot phase started September 2020, full project could start 2021
- ~ 6-7 HPC and sysadmins involved (Tromsø, Trondheim, Oslo, Bergen)
- Very similar to goals of EESSI …
  - Original idea was to distribute existing EB stacks via CernVM-FS (didn't think about archspec and compatibility layer)
- So far mostly experimenting with EESSI pilot
- Some are actively contributing to EESSI (aarch64, CernVM-FS)
- Ultimate goal would be to use a common stack on every national system, and locally where it fits

[Thomas]

Potential sponsorship by Microsoft via Azure (1)

- Interest from Microsoft to “sponsor” the EESSI project
- Mainly through budget to spend in Azure cloud

- Potential use cases:
  - Hosting Stratum-0 server?
    - Not tied to one project partner, managed by multiple people, ...
  - Hosting multiple Stratum-1 servers
    - geographically distributed
    - behind load balancer (auto-started squid proxies based on load)
  - Build nodes for multiple (x86_64) microarchitectures + GPUs

[Davide]

Potential sponsorship by Microsoft via Azure (2)

- More potential use cases:
  - Testing of software layer
    - Across different types of clients (OS, diff. CPU/GPU generations)
    - After (security) updates in compatibility layer
  - Mirror for source tarballs for open source software
- Easy (?) integration with GitHub Actions (also running in Azure)
- Initial exploratory meeting on Fri Sept 11th (Kenneth, Alan, Davide, Laura)
- **Next meeting on Wed Oct 7th** (Kenneth, Alan + more MS people)
  - Better view on EESSI project, initial estimate of resources, ...

[Davide]

Next steps

- Better build script for software layer
  - Proper way of specifying target software stack
  - Leveraging EasyBuild as a library
- Automating of deploying software to CernVM-FS repo (via GitHub App?)
- Continuous Integration in GitHub repos
  - Verifying what's installed in compatibility layer + software layer
  - Running smoke tests in GitHub Actions using EESSI repository
- Testing
  - We need more "real" test cases (now only OpenFOAM)
  - GROMACS, MPI, Python, R, Bioconductor, …
- Documentation
  - Different "use cases": clients, build nodes, Stratum-1, proxies, …
  - Benchmark results?