EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

EESSI meeting

September 3rd 2020

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

Agenda

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

1. Short introduction by new people
2. Progress update
   a. Brainstorm meeting Aug 19th (Peter)
   b. Filesystem layer (Bob)
   c. Compatibility layer (Peter)
   d. Software layer (Kenneth)
3. FENIX project proposals (Alan)
4. Next steps
5. Q & A

Short introduction by new partners

- 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 layer(s) of the project?

Brainstorm meeting Aug 19th

- Topics:
  - Packages: balance between compat layer and the software layer
  - Keeping track of changes in CVMFS repositories: changelog?
  - Only do security updates in compat layer between releases
  - Testing could grow over time: start with EB install bzip2 from source
  - Directory structure: version 2020.08
- Meeting notes available at https://github.com/EESSI/compatibility-layer/wiki/Brainstorm-meeting-(Aug-19th-2020)

Directory structure (2020.08 pilot): compat layer

```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org/
        └── 2020.06/
            ├── compat/
            │   ├── aarch64
            │   ├── ppc64le
            │   └── x86_64/
            │       ├── bin
            │       ├── etc
            │       ├── lib64
            │       ├── sbin
            │       └── usr
            └── software/
                ├── aarch64
                ├── ppc64le
                └── x86_64
```

Compatibility layer provides generic binaries, so one Gentoo Prefix installation per CPU “family”.

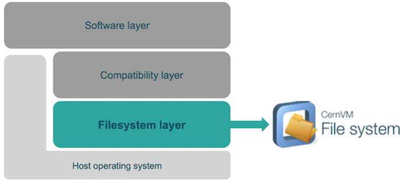
Directory structure (2020.08 pilot): compat layer

```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org/
        └── 2020.06/
            ├── compat
            └── software/
                ├── aarch64
                ├── ppc64le
                └── x86_64/
                    ├── amd/
                    │   └── zen2
                    └── intel/
                        ├── haswell
                        └── skylake/
                            ├── modules
                            └── software/
                                ├── GROMACS/
                                │   └── 2020.1-foss-2020a-Python-3.8.2
                                └── OpenFOAM/
                                    └── 8-foss-2020a
                                    └── v2006-foss-2020a
```

Installations in software layer are optimized for specific CPU microarchitectures.

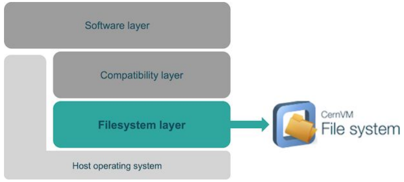
Subdirectory to use is determined via `archspec` Python library.

Progress update: filesystem layer (1)



- Improvements to Ansible playbooks contributed by Pablo
  - (Almost) merged, see https://github.com/EESSI/filesystem-layer/pull/28
  - Easy way of overriding local (site- and machine-specific) variables in a single file
- First (virtual) meeting with CernVM-FS team
  - See https://github.com/EESSI/meetings/wiki/meeting-CernVM-FS-Aug-17th-2020
  - EESSI introduction, upcoming features in CernVM-FS
  - Discussion on best way for allowing multiple build hosts to deploy to CVMFS
  - To be continued in a follow-up meeting on Friday, September 18 (5pm CEST)
  - Good basis for mutually beneficial collaboration
- Some initial tuning/optimizations made in CVMFS configuration

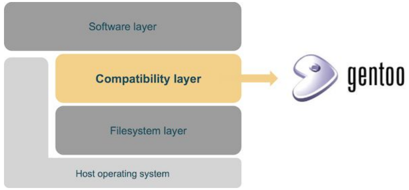
Progress update: filesystem layer (2)



Singularity container to facilitate adding software to `/cvmfs`

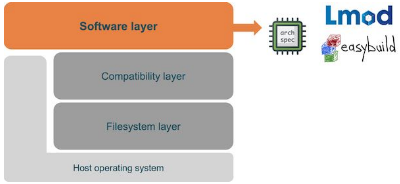
- Everyone can test installations on any machine with (only) Singularity!
- `fuse-overlayfs` is used for making a writable overlay
- Currently in contact with Singularity and `fuse-overlayfs` developers
  - Weird issues with new `fuse-overlayfs` versions, ancient version seems to work fine :-)
- Installations go to `/cvmfs` in container, but actually end up in an overlay dir on host
  - These can then be tarred and ingested into the repository
- The gateway/publisher mechanism can be used for ingestion
  - Probably won’t work without root or using Singularity...
  - Publishers can be given write permission on subtrees of the repository

Progress update: compatibility layer



- Fully re-deployed last week: `/cvmfs/pilot.eessi-hpc.org/2020.08`
- Now includes Lmod and `archspec`
- Automation with Ansible has been merged (see https://github.com/EESSI/compatibility-layer)
- Support for sets to keep track of which packages are installed
- First steps towards a more reproducible and robust way of installing Gentoo Prefix
  - Snapshots etc (see issues #3, #8)
  - Location for gentoo snapshot (see issue #26)
- WIP: GitHub Action workflow for testing Ansible automation in clean Prefix (using a Docker container)

Progress update: software layer



Software layer

Compatibility layer

Filesystem layer

Host operating system

- *Lots* of progress on first version of pilot software stack
- Over 15 pull requests merged in EasyBuild to facilitate installing software in a Gentoo Prefix environment (alternate “sysroot”)
    - All changes will be included in upcoming EasyBuild v4.3.0 release
    - No need for “hacks” or hooks to install the software included in current pilot repository!
- Deploying of software installations to Stratum-0 still “manual”
    - Good setup with Singularity container (isolation from host)
    - Leveraging fuse-overlayfs from container to get “write” access (into host dir of choice)
    - No admin rights or CVMFS (or other software) required on build hosts
    - Still need to figure out the CernVM-MS publisher/gateway approach

EESSI pilot repository

NOT FOR PRODUCTION USE!

- **First version of pilot repository (2020.08) is now available!**
- Over 100 software pkgs included: `foss/2020a`, Python, GROMACS, OpenFOAM
- Easy to test via Singularity container (nothing else required, no admin rights needed)
- Quickly tested by several people, cross OS seems to work (CentOS 7 vs Debian 10)
- Several caveats:
  - Only for systems compatible with Intel Haswell
  - No Lmod cache yet
  - Very basic Open MPI configuration (no support for IB, etc.)
- **See documentation: https://eessi.github.io/docs/pilot**

Demo: EESSI pilot repository

NOT FOR PRODUCTION USE!

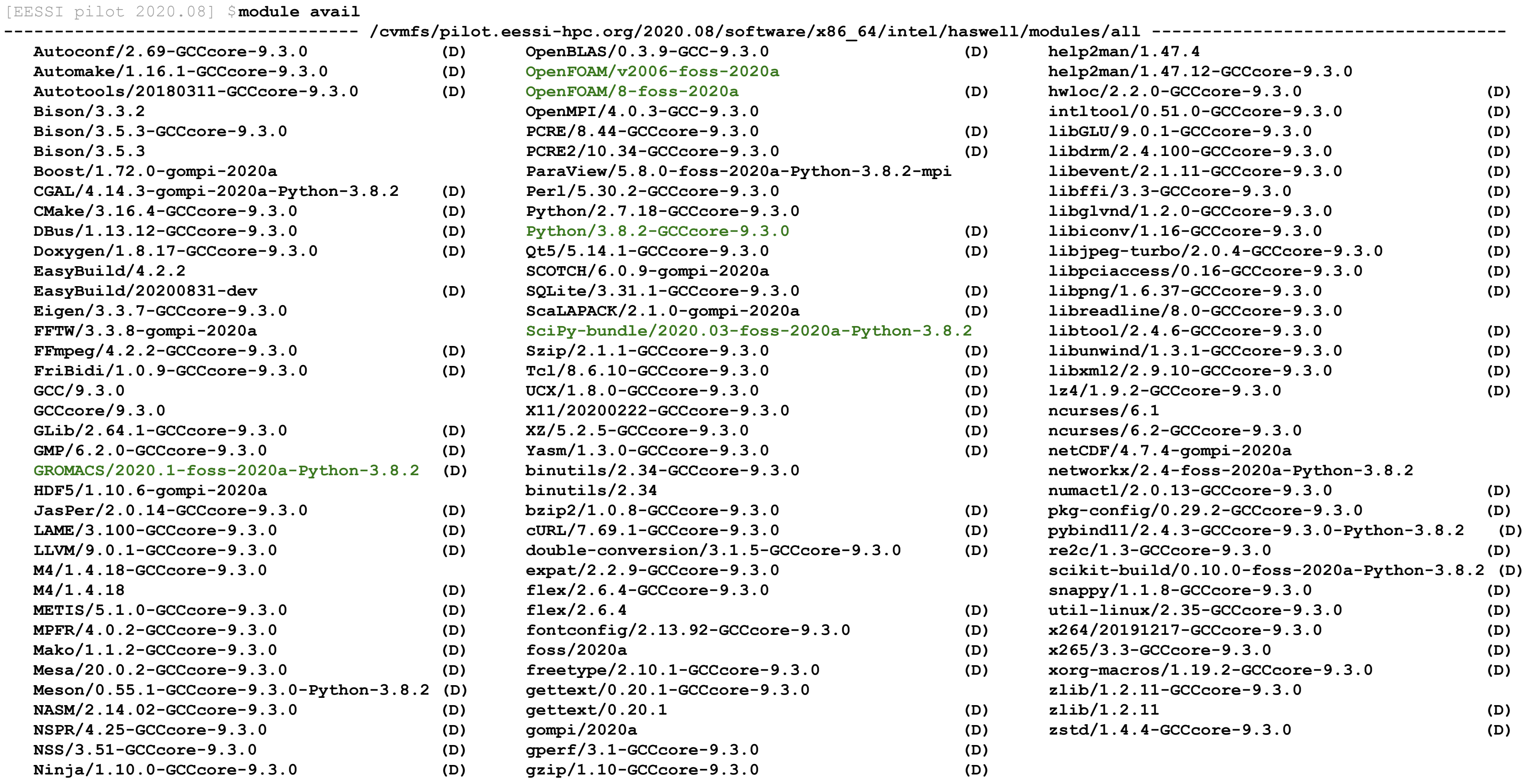
```
$  source /cvmfs/pilot.eessi-hpc.org/2020.08/init/bash
Found EESSI pilot repo @ /cvmfs/pilot.eessi-hpc.org/2020.08!
Derived subdirectory for software layer: x86_64/intel/cascadelake
Using x86_64/intel/haswell subdirectory for software layer (HARDCODED)
Initializing Lmod...
Prepending /cvmfs/pilot.eessi-hpc.org/2020.08/software/x86_64/intel/
haswell/modules/all to $MODULEPATH...
Environment set up to use EESSI pilot software stack, have fun!

[EESSI pilot 2020.08] $  echo $EESSI_PREFIX
/cvmfs/pilot.eessi-hpc.org/2020.08

[EESSI pilot 2020.08] $  module avail gromacs openfoam
---- /cvmfs/.../2020.08/software/x86_64/intel/haswell/modules/all ----
   GROMACS/2020.1-foss-2020a-Python-3.8.2 (D)
   OpenFOAM/v2006-foss-2020a
   OpenFOAM/8-foss-2020a (D)
```

Demo: EESSI pilot repository

NOT FOR PRODUCTION USE!



```
[EESSI pilot 2020.08] $module avail
----------------------------------- /cvmfs/pilot.eessi-hpc.org/2020.08/software/x86_64/intel/haswell/modules/all -----------------------------------
   Autoconf/2.69-GCCcore-9.3.0             (D)     OpenBLAS/0.3.9-GCC-9.3.0                   (D)      help2man/1.47.4
   Automake/1.16.1-GCCcore-9.3.0           (D)     OpenFOAM/v2006-foss-2020a                           help2man/1.47.12-GCCcore-9.3.0
   Autotools/20180311-GCCcore-9.3.0        (D)     OpenFOAM/8-foss-2020a                      (D)      hwloc/2.2.0-GCCcore-9.3.0                  (D)
   Bison/3.3.2                                     OpenMPI/4.0.3-GCC-9.3.0                             intltool/0.51.0-GCCcore-9.3.0              (D)
   Bison/3.5.3-GCCcore-9.3.0                       PCRE/8.44-GCCcore-9.3.0                    (D)      libGLU/9.0.1-GCCcore-9.3.0                 (D)
   Bison/3.5.3                                     PCRE2/10.34-GCCcore-9.3.0                  (D)      libdrm/2.4.100-GCCcore-9.3.0               (D)
   Boost/1.72.0-gompi-2020a                        ParaView/5.8.0-foss-2020a-Python-3.8.2-mpi          libevent/2.1.11-GCCcore-9.3.0              (D)
   CGAL/4.14.3-gompi-2020a-Python-3.8.2    (D)     Perl/5.30.2-GCCcore-9.3.0                           libffi/3.3-GCCcore-9.3.0                   (D)
   CMake/3.16.4-GCCcore-9.3.0              (D)     Python/2.7.18-GCCcore-9.3.0                         libglvnd/1.2.0-GCCcore-9.3.0               (D)
   DBus/1.13.12-GCCcore-9.3.0              (D)     Python/3.8.2-GCCcore-9.3.0                 (D)      libiconv/1.16-GCCcore-9.3.0                (D)
   Doxygen/1.8.17-GCCcore-9.3.0            (D)     Qt5/5.14.1-GCCcore-9.3.0                   (D)      libjpeg-turbo/2.0.4-GCCcore-9.3.0          (D)
   EasyBuild/4.2.2                                 SCOTCH/6.0.9-gompi-2020a                            libpciaccess/0.16-GCCcore-9.3.0            (D)
   EasyBuild/20200831-dev                  (D)     SQLite/3.31.1-GCCcore-9.3.0                (D)      libpng/1.6.37-GCCcore-9.3.0                (D)
   Eigen/3.3.7-GCCcore-9.3.0                       ScaLAPACK/2.1.0-gompi-2020a                (D)      libreadline/8.0-GCCcore-9.3.0
   FFTW/3.3.8-gompi-2020a                          SciPy-bundle/2020.03-foss-2020a-Python-3.8.2        libtool/2.4.6-GCCcore-9.3.0                (D)
   FFmpeg/4.2.2-GCCcore-9.3.0              (D)     Szip/2.1.1-GCCcore-9.3.0                   (D)      libunwind/1.3.1-GCCcore-9.3.0              (D)
   FriBidi/1.0.9-GCCcore-9.3.0             (D)     Tcl/8.6.10-GCCcore-9.3.0                   (D)      libxml2/2.9.10-GCCcore-9.3.0               (D)
   GCC/9.3.0                                       UCX/1.8.0-GCCcore-9.3.0                    (D)      lz4/1.9.2-GCCcore-9.3.0                    (D)
   GCCcore/9.3.0                                   X11/20200222-GCCcore-9.3.0                 (D)      ncurses/6.1
   GLib/2.64.1-GCCcore-9.3.0               (D)     XZ/5.2.5-GCCcore-9.3.0                     (D)      ncurses/6.2-GCCcore-9.3.0
   GMP/6.2.0-GCCcore-9.3.0                 (D)     Yasm/1.3.0-GCCcore-9.3.0                   (D)      netCDF/4.7.4-gompi-2020a
   GROMACS/2020.1-foss-2020a-Python-3.8.2  (D)     binutils/2.34-GCCcore-9.3.0                         networkx/2.4-foss-2020a-Python-3.8.2
   HDF5/1.10.6-gompi-2020a                         binutils/2.34                                       numactl/2.0.13-GCCcore-9.3.0               (D)
   JasPer/2.0.14-GCCcore-9.3.0             (D)     bzip2/1.0.8-GCCcore-9.3.0                  (D)      pkg-config/0.29.2-GCCcore-9.3.0            (D)
   LAME/3.100-GCCcore-9.3.0                (D)     cURL/7.69.1-GCCcore-9.3.0                  (D)      pybind11/2.4.3-GCCcore-9.3.0-Python-3.8.2   (D)
   LLVM/9.0.1-GCCcore-9.3.0                (D)     double-conversion/3.1.5-GCCcore-9.3.0      (D)      re2c/1.3-GCCcore-9.3.0                     (D)
   M4/1.4.18-GCCcore-9.3.0                         expat/2.2.9-GCCcore-9.3.0                           scikit-build/0.10.0-foss-2020a-Python-3.8.2 (D)
   M4/1.4.18                               (D)     flex/2.6.4-GCCcore-9.3.0                            snappy/1.1.8-GCCcore-9.3.0                 (D)
   METIS/5.1.0-GCCcore-9.3.0               (D)     flex/2.6.4                                 (D)      util-linux/2.35-GCCcore-9.3.0              (D)
   MPFR/4.0.2-GCCcore-9.3.0                (D)     fontconfig/2.13.92-GCCcore-9.3.0           (D)      x264/20191217-GCCcore-9.3.0                (D)
   Mako/1.1.2-GCCcore-9.3.0                (D)     foss/2020a                                 (D)      x265/3.3-GCCcore-9.3.0                     (D)
   Mesa/20.0.2-GCCcore-9.3.0               (D)     freetype/2.10.1-GCCcore-9.3.0              (D)      xorg-macros/1.19.2-GCCcore-9.3.0           (D)
   Meson/0.55.1-GCCcore-9.3.0-Python-3.8.2 (D)     gettext/0.20.1-GCCcore-9.3.0                        zlib/1.2.11-GCCcore-9.3.0
   NASM/2.14.02-GCCcore-9.3.0              (D)     gettext/0.20.1                             (D)      zlib/1.2.11                                (D)
   NSPR/4.25-GCCcore-9.3.0                 (D)     gompi/2020a                                (D)      zstd/1.4.4-GCCcore-9.3.0                   (D)
   NSS/3.51-GCCcore-9.3.0                  (D)     gperf/3.1-GCCcore-9.3.0                    (D)
   Ninja/1.10.0-GCCcore-9.3.0              (D)     gzip/1.10-GCCcore-9.3.0                    (D)
```

FENIX project proposals

- See https://fenix-ri.eu/news/prace-icei-calls-proposals-0
- Goals:
  - Access to additional resources for building and scalability tests
  - Additional Stratum 1
  - Magic Castle with EESSI stack (2nd proposal)
- 8 submitted proposals, 2 related to EESSI, both rejected 😭
  - Rejected mostly due to technical restraints
  - FENIX resources are the same machine but the OpenStack part is completely partitioned (no shared mounts, additional ports,...)
  - No overcommitment on OpenStack (disaster for Magic Castle!)

Next steps

- Filesystem layer
  - Work out better way to let build hosts deploy software into pilot repository
  - Additional Stratum-1 servers
  - Optimizations
- Compatibility layer:
  - Automated testing (CI in GitHub Actions)
  - More robust installation of Gentoo Prefix
- Software layer:
  - Software installations for more CPU microarchitectures + auto-select via archspec
  - Revise EasyBuild configuration (don’t define `$LD_LIBRARY_PATH`)
  - Adding more software (TensorFlow, …) + better Open MPI configuration
  - Collecting test scripts in https://github.com/EESSI/software-layer/tree/master/tests
- Pilot software stack: testing, testing, testing!
  - Kickstart new version of pilot repository (2020.09)?