



Spack: Highlights from SC-2019

Chris Green, FNAL

HSF Packaging Meeting, 2019-12-11

Overview

- Attended several of the many Spack-related events, including two round-tables with opportunity for interaction with core Spack developers.
- > 1h one-to-one with Peter Scheibel (a core Spack developer).

Concretizer

- New solver-based concretizer (dependency tree constraint resolver) due in March:
- Will be *much* faster than current *ad hoc* Python-based concretizer.
- New concretizer will enable many improvements required for Spack to do everything we need.
- Can be evaluated for (*e.g.*) LArSoft dependency tree: preview available as a *spack solve* command on a feature branch, not yet integrated into Spack proper.

Concretizer Improvements

- Use of installed/cached binary packages from older recipe versions (cope with hash change).
- Explicit (including no) rather than implicit compiler dependence.
- Language standard compliance as a virtual dependency.
- Use of pre-built older versions of packages as installed or cached binary packages over building latest without explicit version specification.
- Better reconciliation of compatible variant specifications for the same dependency in different dependents.
- Better handling of build (vs link, run, test) dependencies.

Developer Discussions

- Integration of SpackDev¹ into Spack proper as an integrated command.
- Automatic detection and use of system-installed packages.
- Integration of Spack Environments into SpackDev.
- Expansion of the concept of Spack extension commands to more general Spack additions such as extensible integrated commands (external subcommands), new build systems, new module systems.
- Other improvements desired for SpackDev and our use of Spack for package and release building and management.

¹FNAL-originated Spack external command extension for multi-package development.

Spack—Other Highlights

- Explicit callout of FNAL contributions to Spack core in BoF.
- **E4S²**: a base collection of scientific software, built and managed using Spack, part of the official ECP³ effort. Specially-developed (but now generally available) integration with Gitlab CI build pipelines enables quick builds of new software and/or new platforms / architectures. Contents will evolve: soliciting additions to the stack. Investigation required to evaluate usefulness / feasibility of adding HEP-centric software (ROOT, Geant4, *etc.*) vs losing fine control of software versions within a stack.
- **Post-SC Spack article** in **HPC Wire**.

²Extreme-Scale Scientific Software Stack.

³Exascale Computing Project.