FNAL Spack / SpackDev status update

Chris Green, FNAL

LArSoft Coordination Meeting, 2019-06-04

Recap

Previous status reports in this forum:

- 2018-05-22.
- 2018-08-14.

Quick recap:

- Developing a long term replacement for our current UPS-based ecosystem with wide applicability across HEP.
- Spack / cetmodules / SpackDev / BuildCache vs UPS & ssibuildshims / cetbuildtools / MRB / SciSoft.
- MVP: technology preview including art suite stack was almost complete as of 2018-08-14: finalized and released 2018-08-31.
- SpackDev a departure from familiar multi-package development:
    - Top-level build is package-based: each package is installed before dependents are built.
    - “Zoom in” for single-package development and return to the top-level build later.

Lessons Learned from MVP

- Needed wider applicability to spark engagement: LArSoft stack.
- Multiple calls to the Spack concretization system during initialization of a SpackDev development area were very time-consuming and would be expected to get combinatorically worse with a larger software stack.
- Needed a way to handle parallel test execution within the top-level build context.

Progress

- Many enhancements to Spack (from us and other contributors), including the ability to have external Spack commands with their own subcommands (*cf* `git flow release start`).
- SpackDev completely overhauled to become an external Spack command: `spack dev init` vs `spackdev init`.
- Full LArSoft stack, > 160 packages!
- Ability to obtain specified branch of any given package for development from source control, as specified by the recipe.
- Use of BuildCache (Spack functionality) for tools (compiler, git, *etc.*).
- Top-level build supports parallel execution of tests.
- Improvements to algorithm to detect extra checkouts required for a consistent development environment.

User Story

```
# Obtain bootstrap-mvp, one-time setup.
chmod +x bootstrap-mvp && ./bootstrap-mvp -v <scratch-dir>/MVP
# Per session setup.
cd <scratch-dir>/MVP && . setup.sh
# Dev area setup (once, or after dependency changes).
spack dev init -b spackdev-larsoft --dag-file \
../spack_glue/MVP/templates/larsoft-dag.txt -v --default-branch=MVP1a \
larsoftobj larsim larsoft
# Per dev session setup.
cd spackdev-larsoft && . spackdev-aux/env.sh
# Top-level build.
CTEST_PARALLEL_LEVEL=<#> cmake --build ./build -j <#>
# Single package development cycle: edit, make, test ...
spack dev build-env --cd --prompt larsim
...
exit
# Repeat top level integration / single package dev ...
```

Status and Plans

- MVP1a (LArSoft edition) is ready but for tweaks for the stack build against art 3.02.05 and updates to the documentation.
- Hopeful for release by the end of the week: **need interested volunteers** to poke around and see what still needs to be done to meet needs.
- Upcoming:
  - Integrate feedback, improve usability.
  - More `cetbuildtools -> cetmodules` bridging.
  - Support more package build types for development (`MakefilePackage`, `AutotoolsPackage`, *etc.*).
  - Recipe abstraction and simplifications for `cetmodules`-using packages (`CetmodulesPackage`).
  - Platforms, C++ standards, compilers, Python 3.
  - Concurrent releases, release sharing, release management, all-binary installations.