A build system for multiple package development utilizing Spack
James Amundson and Patrick Gartung, Fermilab

Spack

https://github.com/LLNL/spack

- Spack is a flexible package manager for high performance computing
- Linux, OSX, and various supercomputing platforms
- Has a rigorous model that includes multiple platforms, compilers, versions and variants
- Heavily uses RPATH
  - Relocatability through patchelf (Linux) or install_name_tool (OSX)
- Environment management is factored out
- Multiple options: modules, dotkit, Imod and ups in development
- Has a community including many developers from multiple fields

For more information, see poster “Towards more common build tools - experience with using spack in HEP”

SpackDev

- SpackDev is a development manager for packages installable with Spack
- SpackDev uses Spack
- SpackDev assists in the compilation of one or more dependent packages
- The packages themselves depend on neither Spack nor SpackDev

SpackDev does what Spack does not

- Spack is designed to build and install
  - Incremental build are not part of the model
  - While Spack does allow for hand builds of packages…
    - …hand builds are not necessarily identical to those performed by Spack
  - …no easy way to work on more than one package at once

SpackDev creates a build environment

- Installs external dependencies
- Enables incremental builds of dependent packages
- Dependencies handled correctly
- Creates a build area that depends only on standard build tools (CMake, Ninja and/or Make)
- Builds exactly as Spack would
  - including Spack’s compiler wrappers
- IDE-friendly
  - IDE’s such as QtCreator and CLion can be used for development

SpackDev examples

- `spackdev init corge`
  - installs quux and garply, then creates build area for corge
- `spackdev init corge quux`
  - installs garply, then creates build area for corge and quux
- `spackdev init corge garply`
  - creates build area for corge, quux, and garply
  - SpackDev discovers that including quux is necessary for consistent builds, so it automatically adds it to the build area

Spack dependency tree for Root

For more information, see poster “Towards more common build tools - experience with using spack in HEP”