Key4HEP, a Case Study for Software Deployment

Valentin Volkl for the Key4HEP software group

23/09/2020

CERN

Key4hep

- Future detector studies critically rely on **well-maintained** software stacks to model detector concepts and to understand a detector's limitations and physics reach
- We have a scattered landscape of **specific software tools** on the one hand and **integrated frameworks** tailored for a specific experiment on the other hand
- Aim at a low-maintenance common stack for FCC, ILC/CLIC, CEPC with ready to use "plug-ins" to develop detector concepts
- Reached consensus among all communities for future colliders to develop, KEY4HEP, a common turnkey software stack
- Also used as a testbed for some software R&D

The Spack Package Manager

- Spack is a package management tool
  - replaces system package managers like yum and apt with a differentphilosophy
- Originally developed for HPC
  - considered for wide adoption at CERN
    - Future Collider Studies used as a prototype
- Emphasis on dealing with multiple configurations of the same packages
  - Different versions, compilers, external library versions
  - may coexist on the same system
  - Spec: Syntax to describe package version configuration and dependencies

All the complexity of the build system and dependencies goes into **Recipes** * try to push to central spack repository * https://github.com/key4hep/k4-spack for experiment-specifics

The build afterwards becomes very simple:

```
git clone https://github.com/spack/spack.git
git clone https://github.com/key4hep/k4-spack.git
alias spack='python $PWD/spack/bin/spack'
spack repo add k4-spack
# install the bundle-package for the key4hep-stack
spack install key4hep-stack
```

The installation in numbers

- The full installation consists of about 300 packages
- ~50 experiment specific packages
- ~50 HEP specific packages
- ~200 general purpose/system libraries
- 14 GB on disk
- about 6 hours to build (single 4-core machine)

`source setup.sh`

- The runtime environment of a package is also described in the recipe
  - Spack can symlink many packages into a single "view" directory, which is easy to set up
  - Still need to take care of custom runtime environment variables
- Alternatives:
  - Spack provides `modulefiles` for all packages which can be used with the `module` command.
  - More developer friendly setup than views (packages can be loaded individually)

CVMFS Deployment

- CVMFS deployment
  - Spack is right now used with the “rsync workflow”
  - The spack install directory and the spack database files are simply mirrored from the build machine
  - Requires very little effort, but has limitations:
    - parallel builds
    - tuning of filesystem catalogs
  - Existing spack installations on cvmfs could be used as an “upstream installation”

More info on the documentation page:

https://cern.ch/key4hep