Building HEP Software with Spack: Experiences from Pilot Builds for Key4hep and Outlook for LCG Releases

vCHEP 2021 <u>Valentin Volkl (CERN)</u>, Thomas Madlener, Tao Lin, Joseph Wang, Dmitri Konstantinov, Ivan Razumov, Andre Sailer, and Gerardo Ganis

Valentin Volkl: Kev4HEP & Spack

Intro: Goals for a HEP build tool / package manager



- Goal: efficiently expoit computing resources
 - Make the most out of what is available: take advantage modern architectures if possible
- Goal: good tooling for **Users**, **Developers** and **Librarians**
- Goal: Deal with complex software stacks
 - Variety of build systems, patches, platforms ...



Spack for HEP



- Originally written for/by HPC community
 - Emphasis on dealing with multiple configurations of the same packages
 - Different versions, compilers, external library versions ...
 - ... may **coexist** on the same system
- Not the only solution in this problem space:
 - EasyBuild
 - Nix/Guix
 - Conda
 - Gentoo Prefix

See this paper for a nice, if possibly biased, comparison

spack install podio

? version ? microarchitectures ? build arguments ? compiler ? patches





^ root@6.24.00%gcc@8.3.0 ...











- Common Software for STC/SCT, FCC, ILC/CLIC, CEPC
- built with spack

Stack with focus on detector simulations:

- 60 experiment packages
- 50 HEP libraries
- ~14 GB install size
- ~6h to build (4 cores)







key4hep-stack

 Common Software for STC/SCT, FCC, ILC/CLIC, CEPC

• Previously at CHEP 2019!

- Modern Software Stack Building for HEP
 - ... still mostly built against externals, first steps towards a full spack build

Spack: for Users

• Simple:

source /cvmfs/sw.hsf.org/key4hep/setup.sh

- Transparent:
 - Very similar structure as previous installations
 - Setup script generation recently developed
 - View creation possible
- Installations available for many different architectures, OS's*

Spack: for Librarians

- Simple:
 - spack install key4hep-stack
- Shared maintenance
 - Among large and growing spack community

Spack: for Developers



• Simple:

- # build contents of current directory
- spack dev-build podio@master
- Flexible, fine-grained loading of dependency
 - spack dev-build podio@master ^root@6.22.04
- Test against many **different dependencies** with spec matrices
- Develop **multiple packages** at once (and use as dependencies)
 - spack develop podio@master -p /path/to/local/podio
 - spack develop root@master -p /path/to/local/root

- Previously at CHEP 2019!
- <u>SpackDev: Multi-Package Development with Spack</u>
 - ... many ideas now part of upstream spack!

Recent Improvements (* see proceedings)

- Spack distributable on read-only filesystems and installed on cvmfs
- Automated release workflow (gitlab-ci.yaml) with testing before deployment

 Image: Second secon

- Build against **Gentoo Prefix** (usable on any* linux distribution)
- Declaration of Geant4 data packages as "external" to allow sharing between installation
- Efficient **nightly builds** (use git commit as version)
- New clingo-based concretizer already shows more robust concretization (variant forwarding)
- Automatic Generation of a **setup.sh / setup.csh**

Issues - Towards the LCG Releases

- Prototype of LCG_99 available
 - <u>https://gitlab.cern.ch/sft/sft-spack-repo</u>
 - /cvmfs/sw.hsf.org/sft-spack
- Validation required
 - Differences in the Build Configuration can lead to non-obvious errors, even at runtime!
 - Key4hep already validated a good fraction of packages
- Some packages still missing
 - ... but continuously added
- Distribution of binaries (without cvmfs) still to be figured out
- No blocking issues!

Conclusion

- Spack is a viable solution for building HEP Software
 - Strong and growing community

(check spack list --tags hep)

- Complete software stack for Key4hep:
 - STC/SCT, FCC, ILC/CLIC, CEPC

is already being build with spack and in use.

source /cvmfs/sw.hsf.org/key4hep/setup.sh

- Crucial Improvements (nightly builds, read-only spack) added and are being pushed upstream
- Last missing packages are being added for build of LCG releases



Backup: HEP Packages in the Spack repository

spack list --tags hep

	g4ensdfstate	k4marlinwrappe	r pandoraanalysis
acts	g4incl	kaldet	pandorapfa
aida	g4ndl	kaltest	pandorasdk
aidatt	g4neutronxs	kassiopeia	photos
ccs-qcd	g4particlexs	key4hep-stack	physsim
ced	g4photonevaporatio	n kitrack	podio
cedviewer	g4pii	kitrackmarlin	py-gosam
cepcsw	g4radioactivedecay	larcontent py-he	pdata-validator
clhep	g4realsurface	lccd	py-hepunits
clicperformance	eg4saiddata	lccontent	py-particle
clupatra	g4tendl	lcfiplus	py-uproot4

collier	garlic	lcfivertex	pythia6
conddbmysql	gaudi	lcgeo	pythia8
conformaltrack ⁻	ing geant4	lcio	qd
dd4hep	geant4-data	lctuple	qgraf
ddkaltest	geant4-vmc	lhapdf	raida
ddmarlinpandora	agear	lich	relax
delphes	generalbrokenlines	madgraph5amc	rivet
dire	gosam-contrib	marlin	root
dual-readout	guinea-pig	marlindd4hep	syscalc
edm4hep	hepmc	marlinfastjet	tauola
evtgen	hepmc3	marlinkinfit	thepeg
fastjet	hepmcanalysis	marlinpandora	tricktrack
fcalclusterer	heppdt	marlinreco	vbfnlo

fcc-edm	heputils	marlintrkprocessor	s vecgeom
fccsw	herwig3	marlinutil	∨gm
fjcontrib	herwigpp	mcutils	whizard
forwardtrackingilcutil		njet	yoda
g4abla	ildperformance	openloops	
g4emlow	k4fwcore	overlay	

Build Systems, CMake HSF Template etc

- Standardized and well configured package-level build tools like CMake help a lot to reduce maintenance burden
 - Good: explicitly declared build options instead of if(PACKAGE_FOUND)
 - Further HSF best practices:
 - HSF-TN-2020-01
- CMake is the clear favorite for modern C++ packages
- Only a few packages that have diy or exotic build systems
 Openloops (scons), ...
- The HSF project template (<u>https://github.com/HSF/tools/</u>) should continue to be used!





Welcome to Spack!

Spack is a package manager for <u>supercomputers</u>, Linux, and macOS. It makes installing scientific software easy. Spack isn't tied to a particular language; you can build a software stack in <u>Python</u> or R, link to libraries written in C, C++, or Fortran, and easily <u>swap compilers</u> or target <u>specific microarchitectures</u>. Learn more here.

spack.io

- Many more topics than could be covered here
 - CDash, containers, gitlab ci integration, environments, ...
- Refer to the <u>documentation</u> or the recent <u>CPPCon lightning talk</u>
 - Also, <u>presentation</u> by Ben Morgan at pre-GDB software deployment meeting
- Slack channel and mailing list for informal discussions
- Github repository with a lot of activity

CVMFS Deployment to /cvmfs/sw.hsf.org

Several possible workflows (see presentation by Jakob Blomer)

- 1. The Postscript relocation approach
- 2. The rsync approach
- 3. The Gateway approach
- 4. The Container approach

The rsync approach:

- Builder mounts a read/write copy of the /cvmfs tree
- Builder changes/installs software in place
- Publisher uses rsync to pull changes from the builder

Non-trivial to maintain multiple, synchronized publishers



CVMFS Deployment to /cvmfs/sw.hsf.org

Use <u>Gitlab-Cl</u> with a few dedicated openstack build machines to automate deployment.

Ongoing developments in CVMFS like cvmfs_rsync are very helpful

The rsync approach:

- Builder mounts a read/write copy of the /cvmfs tree
- Builder changes/installs software in place
- Publisher uses rsync to pull changes from the builder

Non-trivial to maintain multiple, synchronized publishers



Microarchitectures

- Package specs have "architecture" attribute: platform-os-target
 - E.g. linux-centos7-skylake
- Aware of both generic families, e.g. x86_64, and specific implementations, e.g. skylake (Intel), bulldozer (AMD)
 - https://spack.readthedocs.io/en/latest/basic_usage.html#support-for-specific-microarchitectures
- Architecture is used in install_path_scheme, the directory layout template for installed packages, so they can be distinguised



- In practice, less broadly compatible than lcg release, leading to possible "Illegal Instruction" errors on older CPUs
- See also: <u>https://github.com/archspec/archspec</u>

Spack Specs

- Core concept in Spack is that every package has a "spec" that describes how it should be, or was, built, including
 - Version, compiler-version built with, options used ("variants")
 - Those parameters for each dependency used down the dependency graph
- Taking an example from the Spack docs
 - spack install root@6.20.04:6.22.0%gcc@9.3.0
 arch=linux-ubuntu20.04-broadwell +tmva
 ^python@3.8.2
 - Means: Install root at some version between 6.20.04 and 6.22.0 (inclusive) with the tmva build option enabled, built using gcc at version 9.3.0 for the broadwell architecture, Additionally, it says to use python version 3.8.2 to build root
 - Variants and build options are defined in the package recipe (python file hosted in upstream spack or a dedicated repo)

Spack Specs

This is an abstract spec; spack concretizes it to obtain a hashable, concrete one:

root@6.22.0%gcc@9.3.0~aqua+davix~emacs+examples~fftw~fits~fo
rtran+gdml+gminimal~graphviz+gsl~http~ipo~jemalloc+math~mems
tat+minuit+mlp~mysql+opengl~postgres~pythia6+pythia8+python~
qt4+r+root7+rootfit+rpath~shadow+sqlite+ssl~table+tbb+thread
s+tmva+unuran+vc+vdt+vmc+x+xml+xrootd
build_type=RelWithDebInfo cxxstd=17
patches=22af3471f3fd87c0fe8917bf9c811c6d806de6c8b9867d30a1e3
d383a1b929d7 arch=linux-ubuntu20.04-broadwell

^cmake@3.18.4%gcc@9.3.0~doc+ncurses+openssl+ownlibs~qt
patches=bf695e3febb222da2ed94b3beea600650e4318975da90e4a71d6
f31a6d5d8c3d arch=linux-ubuntu20.04-broadwell

^ncurses@6.2%gcc@9.3.0~symlinks+termlib arch=linux-ubuntu20.04-broadwell

System packages

- Where to draw the line between system and experiment software?
- For the LCG releases, efforts have been made to formalise this (<u>HEPOSLibs</u>)
- For spack, we try to build as much as possible



System packages 2

- Where to draw the line between system and experiment software?
- In the future, it may be possible to even build glibc with spack.
 - "Containerization",
 would allow to use one
 "distroless" build on all
 linux platforms —



System packages 3

- Where to draw the line between system and experiment software?
- Tried an install that uses more external packages for arch linux - works, but will not support

