

Software management with Spack

Trevor Khwam Tabougua

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
Burckhardtweg 4, 37077 Göttingen

November 2022

- Understand the challenges of software installations in HPC.
- Formulate valid spack SPECS
- Install packages on the cluster using Spack.
- Manage different software versions.

Very consistent naming scheme...

```
(base) niplend GMX $ 1
DFTI_LIN/
DFTI_QUAD/
gromacs16_mdstress/
gromacs16_mdstress_test/
gromacs-2019.5/
gromacs2020.3/
gromacs-2020.4/
gromacs2020.4bin/
gromacs-4.6.4_OMPstring/
(base) niplend GMX $
```

```
gromacs-4.6_DFTI/
gromacs-ls-2016.3/
install_quad_new/
STRING/
STRING_RLX/
gromacs-2019.5.tar.gz
gromacs-2020.4.tar.gz
gromacs-ls-2016.3-12282019.tar.gz
```

What exactly was new here?

Version?

What did I test there?

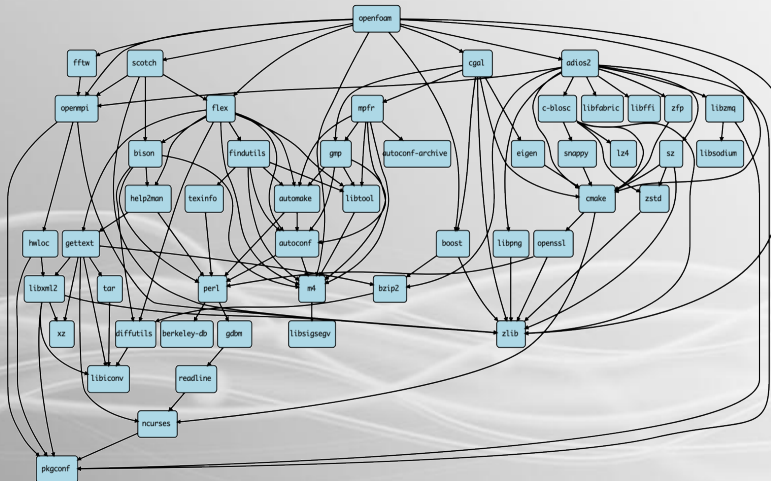
- Prepackaged software typically requires administrator (root) privileges
- admins cannot install all software required by users
- software installation very complex
- different tools may need different versions
- often trade reuse and usability for performance
- Compiling on the target system often yields better performance

HPC Software is complex



- coexistence of several builds
- specific versions of compilers, MPI, libraries, dependencies
- often many dependencies

Openfoam dependency tree



HPC Software is complex



- specific versions of compilers, MPI, libraries, dependencies
- often many dependencies
- many compiling options
- users active on several clusters

Solution: **Spack**

- manages multiple builds
- takes care of dependency relationships
- drives package-level build systems
- wrapper around built systems (cmake/autotools/make etc.)
- targeted towards users, admins and developers

Spack: Supercomputer PACKage manager



- bunch of python scripts
 - ▶ 98.2% according to github
- packages are maintained by the developers and the community
 - ▶ currently 5987 available packages
 - ▶ <https://github.com/spack/spack>
 - ▶ You can contribute!

-



```
openfoam-k5hyzc7e6menegldqfguw2nddghqfz52
```

- most of the modules on the cluster are installed using Spack
- Spack itself is installed as a module `spack-user`
 - ▶ easily find installed software
 - ▶ reuse packages that is already installed
 - ▶ manage your own builds

How to use Spack?



- Load the spack-user module with `module load spack-user`
- Follow the instructions to activate the spack shell support
- Install supported software `spack install SPEC`
- Load the installed software `spack load SPEC`

■ SPECS specify the software configuration

- ▶ i.e. constraints YOU set for your installation
- ▶ optional: only specify what you need

```
$ spack install mpileaks
```

```
$ spack install mpileaks@3.3
```

```
$ spack install mpileaks@3.3 %gcc@4.9.3
```

```
$ spack install mpileaks@3.3 %gcc@4.9.3 +threads
```

```
$ spack install mpileaks@3.3 cppflags="-O3 -g3"
```

```
$ spack install mpileaks@3.3 target=cascadelake
```

```
$ spack install mpileaks@3.3 ^mpich@3.2 %gcc@4.9.3
```

Unconstrained

@ custom version

% custom compiler

+/-/~ build option

Set compiler flags

Set CPU architecture

^ dependency information

Example with nano



```
gwdu102:61 23:24:18 ~ > spack install nano
[+] /opt/sw/rev/21.12/haswell/gcc-9.3.0/pkgconf-1.8.0-ktrb7r
[+] /opt/sw/rev/21.12/haswell/gcc-9.3.0/ncurses-6.2-2dvkjs
[+] Installing nano-4.9-hyadazgagq6itbtp4dwnrgis3xpyrxky
[+] No binary for nano-4.9-hyadazgagq6itbtp4dwnrgis3xpyrxky found: installing from source
[+] Using cached archive: /usr/users/ttaboug/.spack/0.17.1/cache/_source-cache/archive/0e/0e399729d105cb1a587b4140db5cf1b23215a0886a42b215efa98137164233a6.tar.xz
[+] No patches needed for nano
[+] nano: Executing phase: 'autoreconf'
[+] nano: Executing phase: 'configure'
[+] nano: Executing phase: 'build'
[+] nano: Executing phase: 'install'
[+] nano: Successfully installed nano-4.9-hyadazgagq6itbtp4dwnrgis3xpyrxky
Fetch: 0.02s. Build: 49.99s. Total: 50.01s.
[+] /usr/users/ttaboug/.spack/0.17.1/install/haswell/gcc-9.3.0/nano-4.9-hyadaz
```

- `spack find -vdl nano`
- `spack install nano`
- `spack load nano`
- `nano --version`

The Most Important Command



```
spack help
```

- overview over all spack commands

```
spack help <command>
```

- provides information about usage and available options for all spack commands
- usage will help you to get familiar with spack

Basic Spack Commands



spack list

- list of all available packages for spack

spack find

- list of all already installed packages

spack compilers

- list of all installed compilers

spack info

- print information about a spack package

Let's install something!



■ Follow the Tutorial

- ▶ <https://pad.gwdg.de/s/F8bHwyObF#>
- ▶ Experiment with the various spack commands!
- ▶ Discuss with your group!
- ▶ Ask us if you've got any question or need assistance!

■ Use our Spack cheat sheet!

- ▶ <https://pad.gwdg.de/s/QH2VaVqch##>