

The Spack package manager



Erik Schnetter, Perimeter Institute

Einstein Toolkit Call, 2022-05-19

Building software from source is difficult

- Consistent compilers / versions / build options
 - Weird errors (if you are lucky), or no errors (just segfaults at run time)
 - Consistent set of dependencies (HDF5, MPI)
 - Load the correct modules (every time)
 - Might need to manage different versions of stuff
 - So many systems (laptop, workstations, various HPC systems), each unique snowflakes in their own way
-
- Reproducibility? New students?
 - This made sense in 1973.

Industry Standard

- If you manage software, you need to be in complete control of all dependencies (reliability, reproducibility, ease of use)
 - 2000: disk images
 - 2010: virtual machines
 - 2020: containers (Docker, Shifter, Singularity)
- HPC supports none of these, requires building from source
- HPC is stuck in the past

Spack is ...



- “A flexible package manager that supports multiple versions, configurations, platforms, and compilers.”
- <https://spack.readthedocs.io>
- https://ipo.llnl.gov/sites/default/files/2020-01/spack-rd100-2019_final190924.pdf
- Spack:
 - Configures, builds and installs software
 - Installs necessary dependencies automatically
 - Ensures versions and options are chosen consistently
 - Has recipes for thousands of packages, many for HPC
- `spack install amrex`

How I use Spack for the Einstein Toolkit

- Einstein Toolkit: <http://einsteintoolkit.org>
- I install only the ET dependencies via Spack
 - Spack always installs each package into its own directory
 - One directory per package, version, set of options, etc.
- I create a “view” for these dependencies (and their dependencies)
- I build and use the ET the usual way

[show spack.yaml for Einstein Toolkit]

[show package.py for Reprimand]

Spack



- Installing software from source on foreign systems is difficult
- Standard approaches (containers) don't work well on HPC systems
- Spack offers reliability, reproducibility, makes things much simpler
- There are many ways of using Spack (e.g. “spack develop”, generate modules)
- Spack has lots of features for Python (can replace Anaconda)
- There are other package managers (e.g. easybuild <https://easybuild.io>)