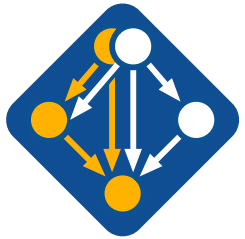


Spack

Getting Scientific Software Installed BOF

Supercomputing 2015 (SC15)

Austin, Texas. November 17, 2015

Todd Gamblin
Center for Applied Scientific Computing

LLNL-PRES-803473
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

http://bit.ly/spack-git

Lawrence Livermore National Laboratory

Spack is a flexible package manager for HPC

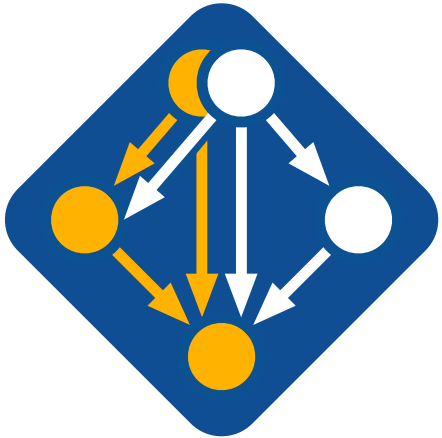
- How to install Spack:

```
$ git clone https://github.com/scalability-llnl/spack.git
```

- How to install a package:

```
$ cd spack/bin
$ ./spack install hdf5
```

- HDF5 and its dependencies are installed within the Spack directory.
- No additional setup required!



Get Spack!

https://bit.ly/spack-git

Spack finds the compilers on your machine.

```
$ spack compilers
==> Available compilers
-- gcc ----------------------------------------------------------
gcc@4.4.7

-- intel --------------------------------------------------------
intel@16.0.0  intel@14.0.3  intel@13.1.0  intel@12.1.2  intel@10.0
intel@15.0.3  intel@14.0.2  intel@13.0.1  intel@12.1.0  intel@9.1
intel@15.0.1  intel@14.0.1  intel@13.0.0  intel@12.0.4
intel@15.0.0  intel@14.0.0  intel@12.1.5  intel@11.1
intel@14.0.4  intel@13.1.1  intel@12.1.3  intel@10.1

-- clang --------------------------------------------------------
clang@3.6.2  clang@3.4  clang@3.3  clang@3.2  clang@3.1

-- pgi ----------------------------------------------------------
pgi@15.5-0   pgi@14.3-0   pgi@13.1-1   pgi@11.1-0  pgi@8.0-1
pgi@15.1-0   pgi@13.10-0  pgi@12.8-0   pgi@10.9-0  pgi@7.1-3
pgi@14.10-0  pgi@13.6-0   pgi@12.1-0   pgi@10.2-0  pgi@7.0-6
pgi@14.7-0   pgi@13.2-0   pgi@11.10-0  pgi@9.0-4
```

- Spack searches PATH
  - Can also give it a prefix
  - Searching modules coming soon
- Any package can be built with any compiler.
  - Swapping is easy and automatic.
  - Not guaranteed to work, but exploring the space is easy.
- Spack will also build compilers like gcc and clang for you, but it doesn't *have* to.

Spack can install the same package many different ways

```
$ spack install mpileaks                          default
$ spack install mpileaks@3.3                      @ custom version
$ spack install mpileaks@3.3 %gcc@4.7.3           % custom compiler
$ spack install mpileaks@3.3 %gcc@4.7.3 +threads  +/- build option
$ spack install mpileaks@3.3 =bgq                 = cross-compile
```

- These are called ***spec expressions***
  - Concise and expressive language for describing builds.

- Specs describe a configuration of a package.
  - Arbitrary configurations are possible and can coexist.
  - Install it your way!

`spack find` shows what is installed

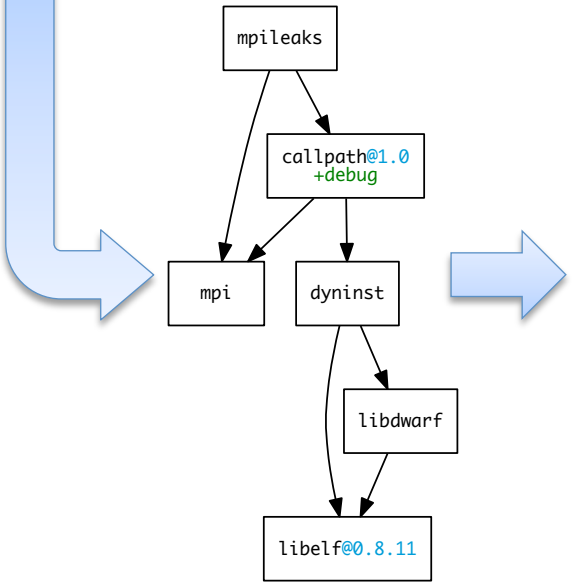
```
$ spack find
==> 103 installed packages.
-- linux-x86_64 / gcc@4.4.7 ---------------------------------
ImageMagick@6.8.9-10  glib@2.42.1        libtiff@4.0.3     pango@1.36.8          qt@4.8.6
SAMRAI@3.9.1          graphlib@2.0.0     libtool@2.4.2     parmetis@4.0.3        qt@5.4.0
adept-utils@1.0       gtkplus@2.24.25    libxcb@1.11       pixman@0.32.6         ravel@1.0.0
atk@2.14.0            harfbuzz@0.9.37    libxml2@2.9.2     py-dateutil@2.4.0     readline@6.3
boost@1.55.0          hdf5@1.8.13        llvm@3.0          py-ipython@2.3.1      scotch@6.0.3
cairo@1.14.0          icu@54.1           metis@5.1.0       py-nose@1.3.4         starpu@1.1.4
callpath@1.0.2        jpeg@9a            mpich@3.0.4       py-numpy@1.9.1        stat@2.1.0
dyninst@8.1.2         libdwarf@20130729  ncurses@5.9       py-pytz@2014.10       xz@5.2.0
dyninst@8.1.2         libelf@0.8.13      ocr@2015-02-16    py-setuptools@11.3.1  zlib@1.2.8
fontconfig@2.11.1     libffi@3.1         openssl@1.0.1h    py-six@1.9.0
freetype@2.5.3        libmng@2.0.2       otf@1.12.5salmon  python@2.7.8
gdk-pixbuf@2.31.2     libpng@1.6.16      otf2@1.4          qhull@1.0

-- linux-x86_64 / gcc@4.8.2 ---------------------------------
adept-utils@1.0.1  boost@1.55.0  cmake@5.6-special  libdwarf@20130729  mpich@3.0.4
adept-utils@1.0.1  cmake@5.6     dyninst@8.1.2      libelf@0.8.13      openmpi@1.8.2

-- linux-x86_64 / intel@14.0.2 ------------------------------
hwloc@1.9  mpich@3.0.4  starpu@1.1.4

-- linux-x86_64 / intel@15.0.0 ------------------------------
adept-utils@1.0.1  boost@1.55.0  libdwarf@20130729  libelf@0.8.13  mpich@3.0.4

-- linux-x86_64 / intel@15.0.1 ------------------------------
adept-utils@1.0.1  callpath@1.0.2  libdwarf@20130729  mpich@3.0.4
boost@1.55.0       hwloc@1.9       libelf@0.8.13      starpu@1.1.4
```

- All the versions coexist!
  - Multiple versions of same package are ok.
- Packages are installed to automatically find correct dependencies.
- Binaries work *regardless of user's environment*.
- Spack also generates module files.
  - Don't *have* to use them.

Spack tracks detailed provenance for every build.

- Every configuration is a new dependency graph.
  - Spack hashes the graph to make a version id.
  - Full graph information is stored with each build.
- Spack allows *arbitrary composition* of packages
  - Build Python with different compilers and BLAS implementations.
  - Build your code with different compiler/MPI/dependency versions.

```
spack install mpileaks ^callpath@1.0+debug ^libelf@0.8.11
```




```
spec:
- mpileaks:
    arch: linux-x86_64
    compiler:
      name: gcc
      version: 4.9.2
    dependencies:
      adept-utils: kszrtkpbzac3ss2ixcjkcorlaybnptp4
      callpath: bah5f4h4d2n47mgycej2mtrnrivvxy77
      mpich: aa4ar6ifj23yijqmdabeakpejcli72t3
    hash: 33hjjhxi7p6gyzn5ptgyes7sghyprujh
    variants: {}
    version: '1.0'
- adept-utils:
    arch: linux-x86_64
    compiler:
      name: gcc
      version: 4.9.2
    dependencies:
      boost: teesjv7ehpe5ksspjim5dk43a7qnowlq
      mpich: aa4ar6ifj23yijqmdabeakpejcli72t3
    hash: kszrtkpbzac3ss2ixcjkcorlaybnptp4
    variants: {}
    version: 1.0.1
- boost:
    arch: linux-x86_64
    compiler:
      name: gcc
      version: 4.9.2
    dependencies: {}
    hash: teesjv7ehpe5ksspjim5dk43a7qnowlq
    variants: {}
    version: 1.59.0
...
```

spec.yaml

Spack allows Python extensions to be activated and deactivated on demand

```
$ spack install python@2.7.10
==> Building python.
==> Successfully installed python.
  Fetch: 5.01s.  Build: 97.16s.  Total: 103.17s.
[+] /home/gamblin2/spack/opt/spack/linux-x86_64/gcc-4.9.2/python-2.7.10-y2zr767

$ spack extensions python@2.7.10
==> python@2.7.10%gcc@4.9.2=linux-x86_64-y2zr767
==> 49 extensions:
geos           py-h5py        py-numpy       py-pypar            py-setuptools
libxml2        py-ipython     py-pandas      py-pyparsing        py-shiboken
py-basemap     py-libxml2     py-pexpect     py-pyqt             py-sip
py-biopython   py-lockfile    py-pil         py-pyside           py-six
py-cffi        py-mako        py-pmw         py-python-daemon    py-sphinx
py-cython      py-matplotlib  py-pychecker   py-pytz             py-sympy
py-dateutil    py-mock        py-pycparser   py-rpy2             py-virtualenv
py-epydoc      py-mpi4py      py-pyelftools  py-scientificpython py-yapf
py-genders     py-mx          py-pygments    py-scikit-learn     thrift
py-gnuplot     py-nose        py-pylint      py-scipy

==> 3 installed:
-- linux-x86_64 / gcc@4.9.2 --------------------------------------
py-nose@1.3.6  py-numpy@1.9.2  py-setuptools@18.1

==> None currently activated.

$ spack activate py-numpy
==> Activated extension py-setuptools-18.1-gcc-4.9.2-ru7w3lx
==> Activated extension py-nose-1.3.6-gcc-4.9.2-vudjpwc
==> Activated extension py-numpy-1.9.2-gcc@4.9.2-45hjazt

$ spack deactivate -a py-numpy
==> Deactivated extension py-numpy-1.9.2-gcc@4.9.2-45hjazt
==> Deactivated extension py-nose-1.3.6-gcc-4.9.2-vudjpwc
==> Deactivated extension py-setuptools-18.1-gcc-4.9.2-ru7w3lx
```

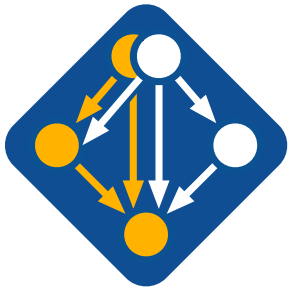
- Many interpreted languages have their own mechanisms for modules, e.g.:
  - Require installation into interpreter prefix
  - Breaks combinatorial versioning
- Spack installs each Python package in its own prefix
- "Activating" links an extension into the interpreter directory on demand
  - Supports .egg, merging .pth files
  - Mechanism is extensible to other languages
  - Similar to virtualenv, but Spack allows much more build customization.

Get Involved with Spack!

- **Come to our SC15 Talk:**
  The Spack Package Manager: Bringing Order to HPC Software Chaos
  Wednesday, 11:30am
  Hall 18AB

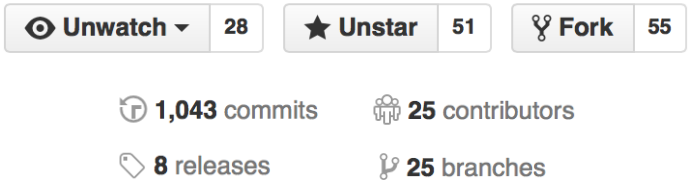
- Spack is starting to be used in production at LLNL
  - Build, test, and deployment by code teams.
  - Tools, libraries, and Python at Livermore Computing.
  - Build research projects for students, postdocs.

- Spack has a rapidly growing external community.
  - NERSC is working with LLNL on Cray support for Cori.
  - Argonne/IIT cluster challenge project.
  - Kitware contributing ParaView builds & features.
  - Users at INRIA, EPFL, U. Oregon, Sandia, LANL, others.



Get Spack!

https://bit.ly/spack-git




Lawrence Livermore
National Laboratory