

Spack Community BOF

Supercomputing 2018
November 12, 2018
Dallas, Texas



LLNL-PRES-806064

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

github.com/spack/spack

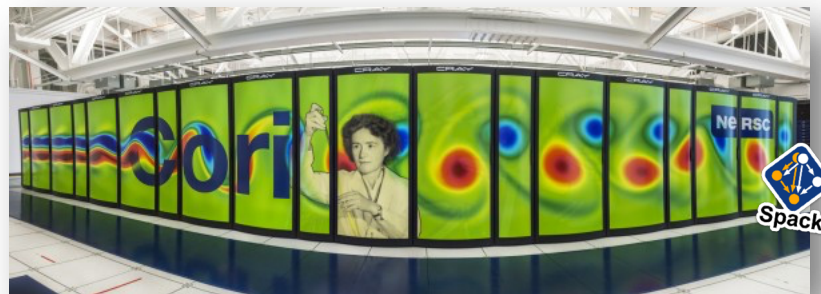


Spack is used at some of the largest supercomputer centers in the world



Spack deploys user software stacks #1 Summit (ORNL) and #2 Sierra (LLNL). They wouldn't let me put a sticker on it, though...

Veronica Vergara Larrea from OLCF will present more on Spack on Summit!



There IS a sticker somewhere on Cori 😊



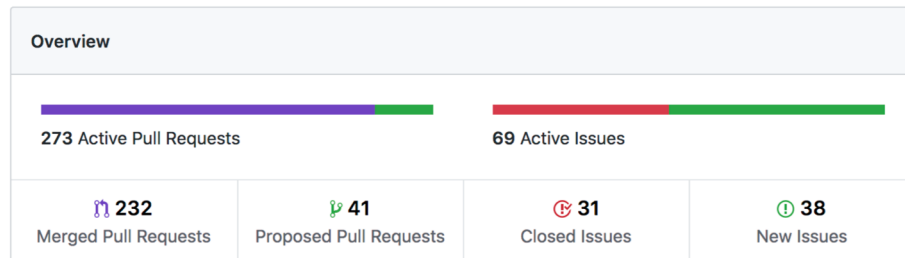
SuperMUC-NG at LRZ (not sure if there's a sticker)
Johannes van der Gonna to present

Spack has a very active open source community

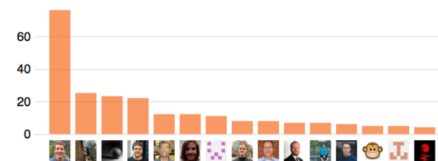
- Started Spack development in 2013
 - Paper at SC15
 - Tutorials at SC16, SC17, SC18
 - GitHub community has grown steadily!
- 232 pull requests merged in the past month!
 - By 74 contributors
 - We've been gradually increasing core contributors
- Adam Stewart will talk more about managing the community

October 13, 2018 – November 13, 2018

Period: 1 month ▾



Excluding merges, **78 authors** have pushed **289 commits** to develop and **344 commits** to all branches. On develop, **3,487 files** have changed and there have been **25,989 additions** and **81,462 deletions**.



1 Release published by 1 person

Published v0.12.0 25 minutes ago

232 Pull requests merged by 74 people

Merged #9783 perl-net-ssleay: fix build 2 minutes ago

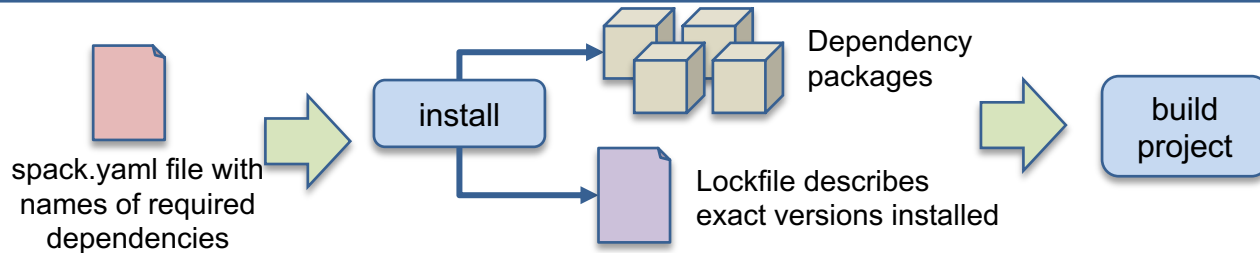
Spack v0.12.0 was just released

- **New stuff (selected):**
 1. Spack environments (covered today)
 2. `spack.yaml` and `spack.lock` files for tracking dependencies (covered today)
 3. Custom configurations via command line (covered today)
 4. Better support for linking Python packages into view directories (pip in views)
 5. Support for uploading build logs to CDash
 6. Packages have more control over compiler flags via flag handlers
 7. Better support for module file generation
 8. Better support for Intel compilers, Intel MPI, etc.
 9. Many performance improvements, improved startup time
- Spack is now permissively licensed under **Apache-2.0 or MIT**
 - previously LGPL
- Over 2,900 packages (800 added since last year)

Why Apache-2.0/MIT instead of LGPL?

- Many potential contributors approached us about obstacles using Spack in their organization, due to the LGPL license
 - Approvals required
 - How to deal with NDA issues?
 - What counts as a derivative work
- We want Spack to be used as broadly as possible, so we changed to a permissive license to make this clear and remove confusion.
 - We want to encourage companies and hardware vendors to make sure Spack packages work well on their architecture!
 - We hope that this will **increase** the contributor base and keep the Spack community growing.
- We've obtained consent from 300+ contributors to make this happen in 0.12.0

Spack has added environments and spack.yaml / spack.lock



- Allows developers to bundle Spack configuration with their repository
- Can also be used to maintain configuration together with Spack packages.
 - E.g., versioning your own local software stack with consistent compilers/MPI implementations
- Manifest / Lockfile model pioneered by Bundler is becoming standard
 - spack.yaml describes project requirements
 - spack.lock describes exactly what versions/configurations were installed, allows them to be reproduced.
- Can help with making purpose-built containers!

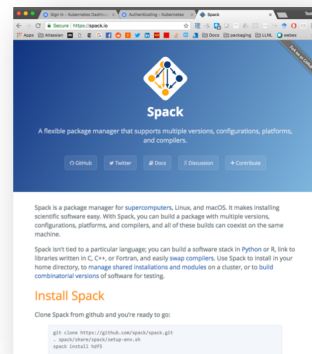
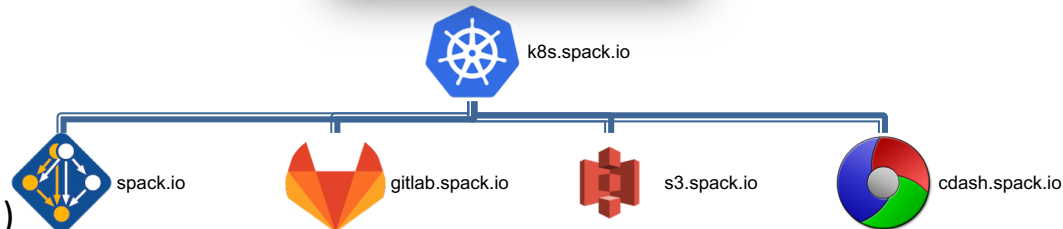
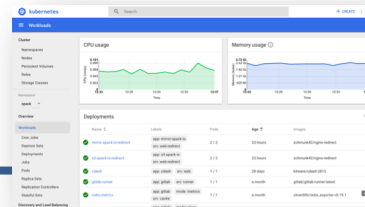
```
spack:
  # include external configuration
  include:
  - ../special-config-directory/
  - ./config-file.yaml

  # add package specs to the `specs` list
  specs:
  - hdf5
  - libelf
  - openmpi
```

Simple spack.yaml file

What we've built

- All Spack infrastructure is consolidated under Kubernetes in Amazon EKS
 - Allows us to document and distribute management of infrastructure
 - All services are containerized
- A mirror is set up in S3 (to be populated soon)
- Kitware has built a release pipeline in Gitlab CI
 - Currently working on stable CentOS7 and Ubuntu pipelines with gcc and clang
 - We would like to expand this!
- GitLab CI builds Spack packages as they are added to release branches
- CDash is deployed in Kubernetes to display build status for releases



The screenshot shows the CDash dashboard for the 'Spack' project. It displays a table of build results with columns for 'Build Name', 'Build Status', 'Start Time', and 'End Time'. The table shows several successful builds for different Spack versions and architectures.

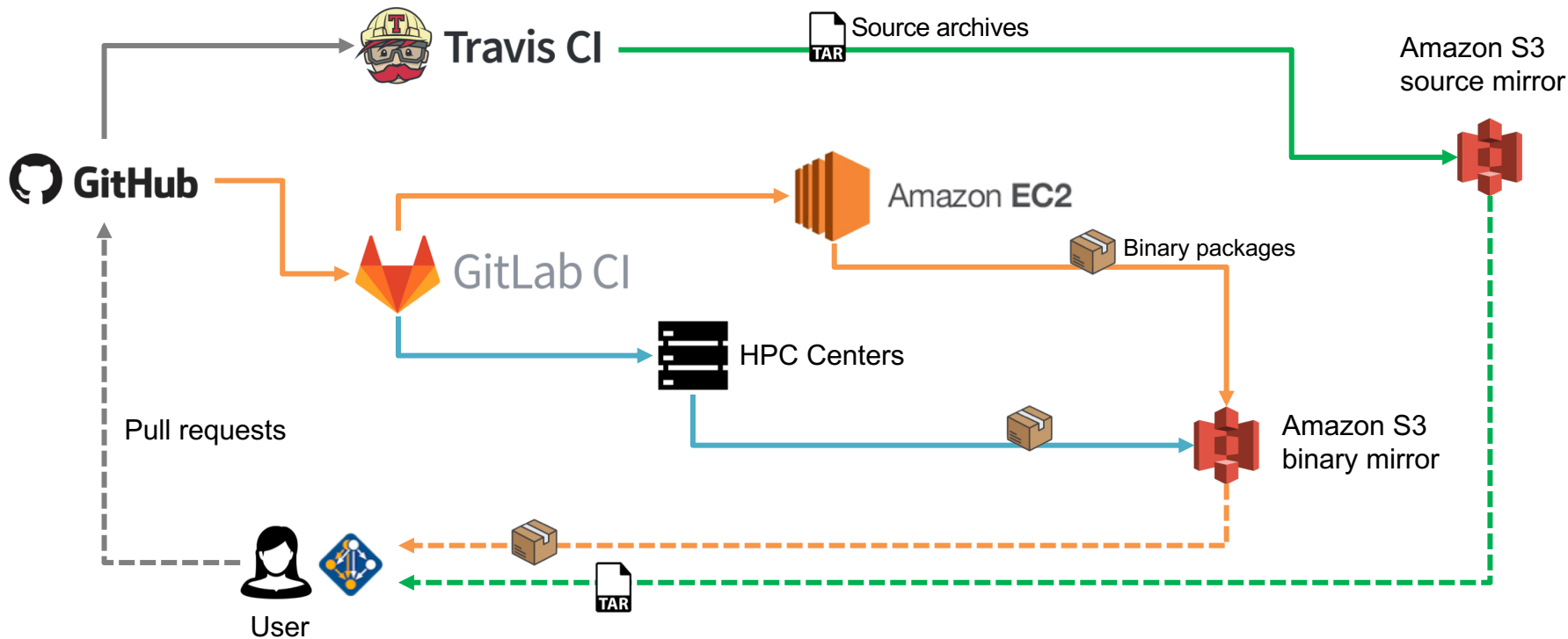
Build Name	Build Status	Start Time	End Time
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00
spack-0.12.0-gcc-4.8.4-ubuntu14.04	Success	2017-02-01 10:00:00	2017-02-01 10:00:00

Future Spack directions

- **Spack stacks:** Build on environments to enable more automated deployment at HPC centers.
 - Single YAML-file configuration for entire site stack
 - Build combinatorial package installations, modules, etc. with one command.
- **Better dependency resolution**
 - Handle newer C++ libraries better
 - More aggressive concretizer support
 - Support for depending on language levels/compiler features (e.g., C++14, lambdas, OpenMP@version)
- **Optimized binary packages**
 - Add more support for binary provenance in builds
 - In conjunction with ECP CI, start to build binaries for HPC facilities



We are building CI infrastructure for source and binary distribution



Spack is the delivery platform for the ECP software stack

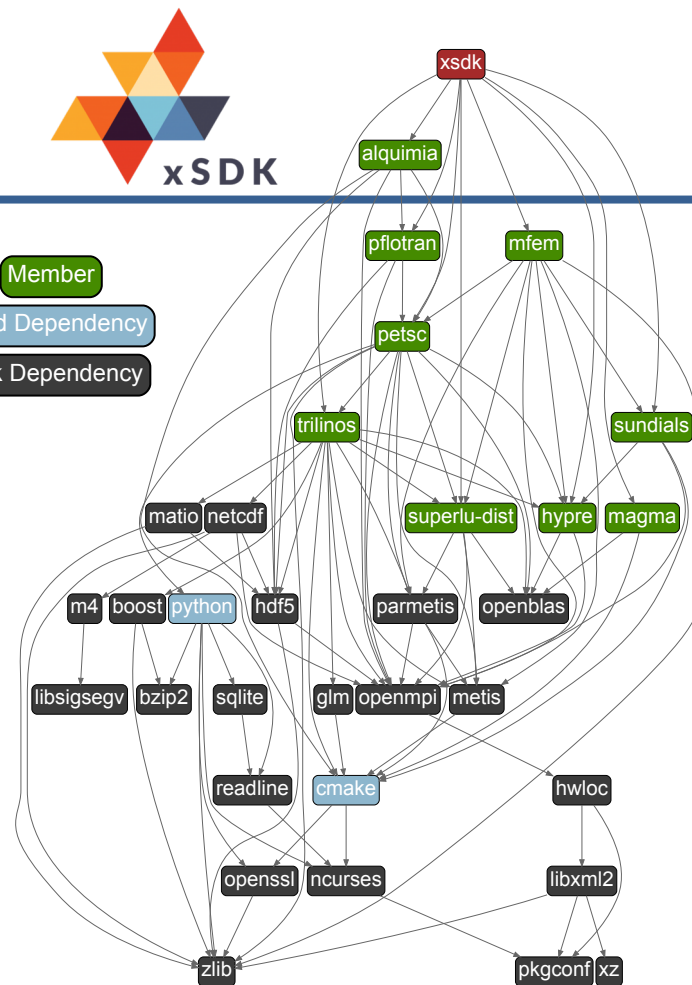
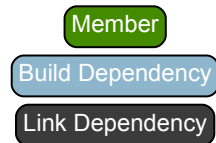
- U.S. Exascale Computing Project (ECP) will release software through Spack
- Software in ECP stack needs to run on ECP platforms, testbeds, clusters, laptops
 - Each new environment requires effort.
- ECP asks us to build a robust, reliable, and easy-to-use software stack
- We are working to provide the infrastructure necessary to make this tractable:
 1. A dependency model that can handle HPC software
 2. A hub for coordinated software releases (like xSDK)
 3. Build and test automation for large packages across facility
 4. Hosted binary and source software distributions for *all* ECP HPC platforms



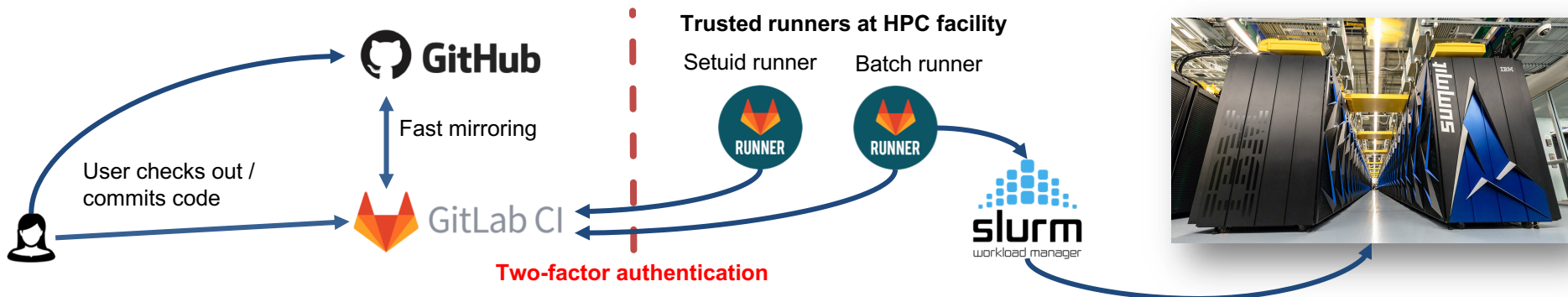
Under ECP, “SDK” teams manage releases using Spack



- xSDK pioneered the SDK concept
 - 8 member projects, 22 required dependencies
 - Includes many major solver library teams
 - Next release will have 10+ additional packages
- Teams work together on regular releases
 - Helps to work out compatibility issues
 - Gets developers talking to each other
 - Encourages teams to factor into smaller libraries
- ECP is establishing more SDKs for different areas
 - SDKs will be released using Spack
 - Each will be its own meta-package like xSDK

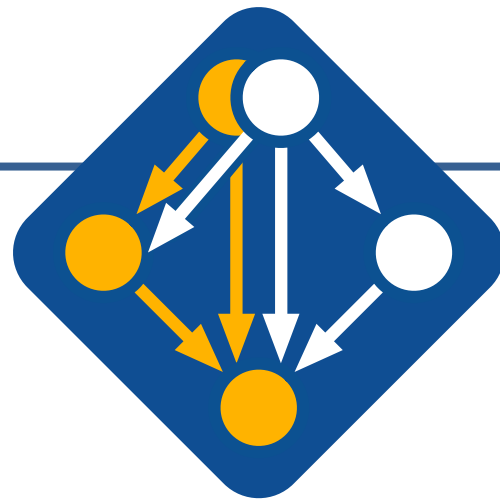


Through ECP, we are working with Onyx Point to deliver continuous integration for HPC centers



- CI at HPC centers is notoriously difficult
 - Security concerns prevent most CI tools from being run by staff or by users
 - HPC centers really need to deploy trusted CI services for this to work
- We are developing a secure CI system for HPC centers:
 - Setuid runners (run CI jobs as users); Batch integration (similar, but parallel jobs); multi-center runner support
- Onyx Point will upstream this support into GitLab CI
 - Initial rollout in FY19 at ECP labs: ANL, ORNL, NERSC, LLNL, LANL, SNL
 - Upstream GitLab features can be used by anyone!

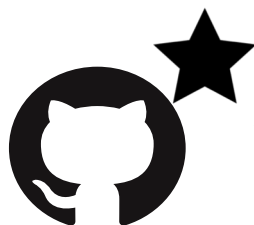
More ways to engage with Spack!



- Take our survey!
<https://bit.ly/sc18-spack-bof>



- Spack Roundtable at the DOE Booth (2433)
3-4pm Wednesday
Meet the developers, talk to us about anything!



Star us on GitHub!
github.com/spack/spack



Follow us on Twitter!
[@spackpm](https://twitter.com/spackpm)

We hope to make distributing & using HPC software easy!

