CUG 2018 BoF
Managing Effectively the User Software Ecosystem

Bilel Hadri

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology | SHAHEEN SUPERCOMPUTING LABORATORY

Guilherme Peretti-Pezzi

CSCS Centro Svizzero di Calcolo Scientifico Swiss National Supercomputing Centre

Chris Fuson

OAK RIDGE National Laboratory

Mario Melara and Helen He

NERSC

Peggy Sanchez

CRAY THE SUPERCOMPUTER COMPANY

Agenda

- Motivation and Goal of the BoF
- Survey: How do you maintain software stack ?
- Short presentation on the different strategies by NERSC, CSCS, ORNL, KAUST
- Cray Demo
- Q/A

Motivations

- Supercomputing centers supports and maintains hundreds of software packages,
  - each with multiple versions, and each version potentially built with multiple compilers and usually have complex dependencies.
  - Necessary to upgrade packages on a regular basis and to make the process of installation versatile and automated as much as possible.
  - Need more reproducible by any member of the staff and result in less issues faced by end users.

- To manage effectively the user software ecosystem, many CUG member sites have adopted different strategies and employed different tools such as EasyBuild, SWTools and Spack.
  - these tools have its advantages and inconveniences
  - in some cases, the installation does not consider the optimal configuration to achieve the best performance on Cray platform.

Goal

- Start a discussion between CUG members along with Cray Documentation and Performance teams,
  - Share the strengths and weaknesses of the strategies currently adopted,
  - Gather the best recipes of installation,
  - Merge our efforts in a common repository hosted in the Cray Portal Documentation or on the CUG website for example.

- Centralize in one place all the recipes.
  - This catalogue gathering of the different applications installations will benefit the whole HPC community using Cray systems.
  - Forum for discussion PEAD-SIG mailing list.
    - Don’t be shy, ask for advise/help. (it’s free :D)

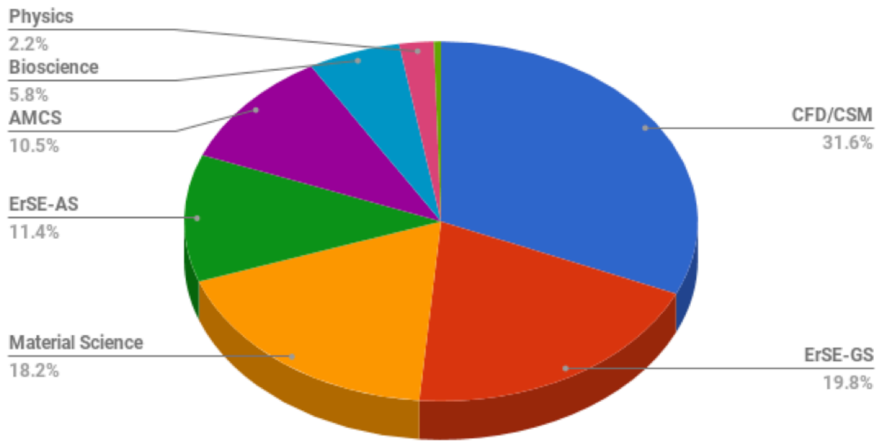
Survey

- Are you using EasyBuild, SWTools and Spack?
- Do you have your own implementation?
- Are you targeting a basic installation or optimized ?

KAUST users/ecosystem

- 7.2 PF Cray XC40 (Haswell) supporting over 700 users, 300 projects.
- Support more than 130 app/libraries

Field of Science	CPU hours	% overall
CFD/CSM	766,887,001	31.59%
ErSE-GS	479,933,780	19.77%
Material Science	442,328,890	18.22%
ErSE-AS	277,852,890	11.44%
AMCS	255,868,987	10.54%
Bioscience	141,445,255	5.83%
Physics	52,611,730	2.17%
Others	10,870,534	0.45%

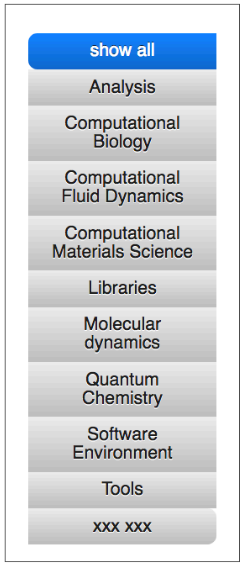


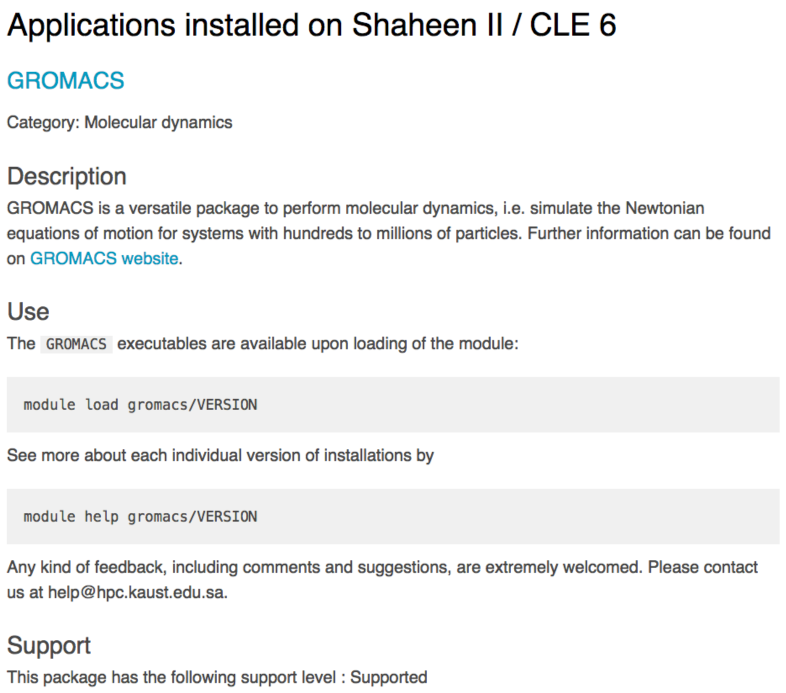
More than 2.4 Billion Core hours in the last 32 months.

SWTOOLS

- SWTools created to help manage third-party software installations at supercomputer centers. It was designed to keep the installations consistent and up-to-date while trying to avoid problems encountered with previous software repositories. Deployed at NICS, OLCF, NCSA and KAUST.
- Automated building, testing, linking, reporting
  - Scripts to rebuild, relink , and retest each application
  - Ability to do batch operations
  - Easily maintainable
- Inventory of currently installed software
  - Automate generation of many user documents
  - Enforcement of rules
- Cons:
  - Interactive installation , Module are manually created
- More information:
  - **Nick Jones and Mark R. Fahey, "Design, Implementation, and Experiences of Third-Party Software Administration at the ORNL NCCS," Proceedings of the 50th Cray User Group (CUG08), Helsinki, Finland, May 2008.**

Online documentation





Applications installed on Shaheen II / CLE 6

GROMACS

Category: Molecular dynamics

Description

GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. Further information can be found on GROMACS website.

Use

The `GROMACS` executables are available upon loading of the module:

```
module load gromacs/VERSION
```

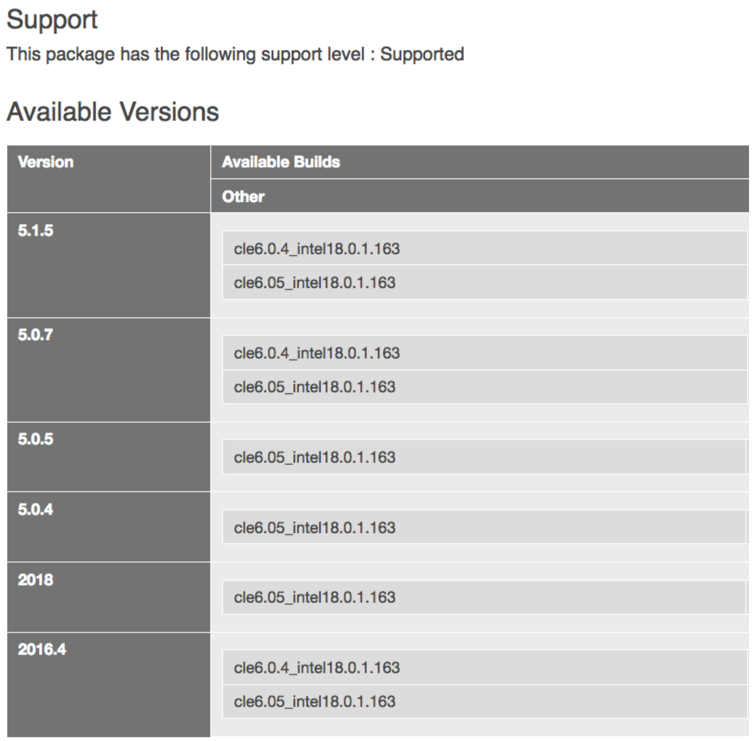
See more about each individual version of installations by

```
module help gromacs/VERSION
```

Any kind of feedback, including comments and suggestions, are extremely welcomed. Please contact us at help@hpc.kaust.edu.sa.

Support

This package has the following support level : Supported



Support

This package has the following support level : Supported

Available Versions

Version	Available Builds
	Other
5.1.5	cle6.0.4_intel18.0.1.163 cle6.05_intel18.0.1.163
5.0.7	cle6.0.4_intel18.0.1.163 cle6.05_intel18.0.1.163
5.0.5	cle6.05_intel18.0.1.163
5.0.4	cle6.05_intel18.0.1.163
2018	cle6.05_intel18.0.1.163
2016.4	cle6.0.4_intel18.0.1.163 cle6.05_intel18.0.1.163

Oak Ridge Leadership Computing Facility

- Mission: Provide the computational and data science resources required to solve the world's most impactful scientific & engineering problems.
- Users cover multiple science domains and experience levels
- New projects and users are added to system throughout year
- Average 250 projects and 1,000 active users per year
- OLCF users are spread out all over the globe and come from academia, national laboratories, other government agencies, and industry
- Software requirements often vary between projects

Titan

Cray XK7

27 petaflops

299,008 Processor Cores

18,688 GPUs

Software Environment

- Diverse user community, diverse software needs
  - Manage and provide multiple software packages, libraries, compilers, versions, build configurations
- Mix of Cray provided and center provided packages
- Modules used to help manage environment
  - Environment modules used on OLCF Cray systems
  - Lmod used on most non-Cray systems
- Center controls defaults, future and previous versions, parings
- Center software areas:

/sw/<system>

- Center installed, maintained, and supported
- Available to all system users

/proj/<project_ID>

- Project installed, maintained
- Limited center support
- Available to only members of the project

Choosing Packages to Install

- Initial packages chosen from acceptance requirements
  - Users requests packages once in production
- Approving package also assumes some center responsibility
  - Documentation, testing, updating
- Not all requests approved
  - Packages chosen based on benefit to user community
  - Assist project in building in project area
    - NetApp area accessible to all project members
    - Backed-up, quota, available from compute nodes
- Dealing with requests for multiple versions and build configurations
  - Provide limited versions
  - Provide single build configuration
  - Projects can use center’s build recipe to aid building in project area

Software Updates and Changes

- Large number of users/projects with varying software version needs
  - Some need latest version
  - Some have difficulty moving to recent releases
- Provide multiple versions
  - previous, default, more recent
    - 1.2, 1.4, **2.3**, 2.6
  - A version may be installed for specific projects, but may never become default
- Changes to default and removals can impact usability/results
- Include users in changes to defaults and removals
  - Provides notification and opportunity to test before default changes
- Notifications
  - Weekly email
  - Web
  - Module load message

Installation Tools

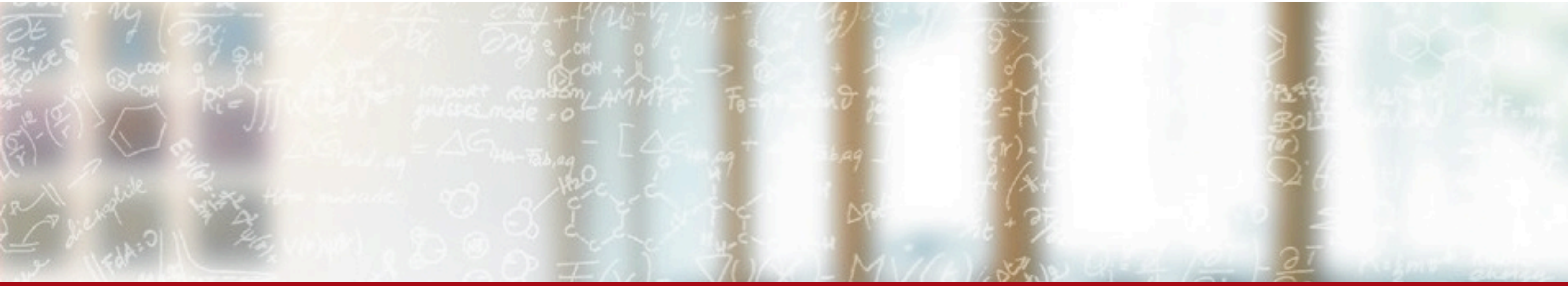
- Manually installing packages has large overhead and is problematic
  - Need repeatable standardized method for installations, modulefiles, and documentation
- Installation tools and installation processes used over the years

swtools $\Rightarrow$ **smithy** $\Rightarrow$ **spack**

- Spack on NCCS systems
  - Community support
  - Share common build recipes
- NCCS CI Management
  - GitLab runners
  - Enforce process, installation location, format
  - Install only as software user
  - Verify before installation

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Scientific Software Management @ CSCS

Cray User Group 2018 (CUG2018)
Managing Effectively the User Software Ecosystem (BoF)
May 22nd, Stockholm

Guilherme Peretti-Pezzi

Scientific Computing Support (CSCS)

Piz Daint

- #3 Top 500
  - #1 in Europe
  - 19.590 PFLOPS
- #10 Green 500
  - 10.398 MFLOPS/W
- GPU partition
  - P100 + Haswell
- MC Partition
  - 2 x Broadwell

Improving software stack quality using SCM, code review and Continuous Integration

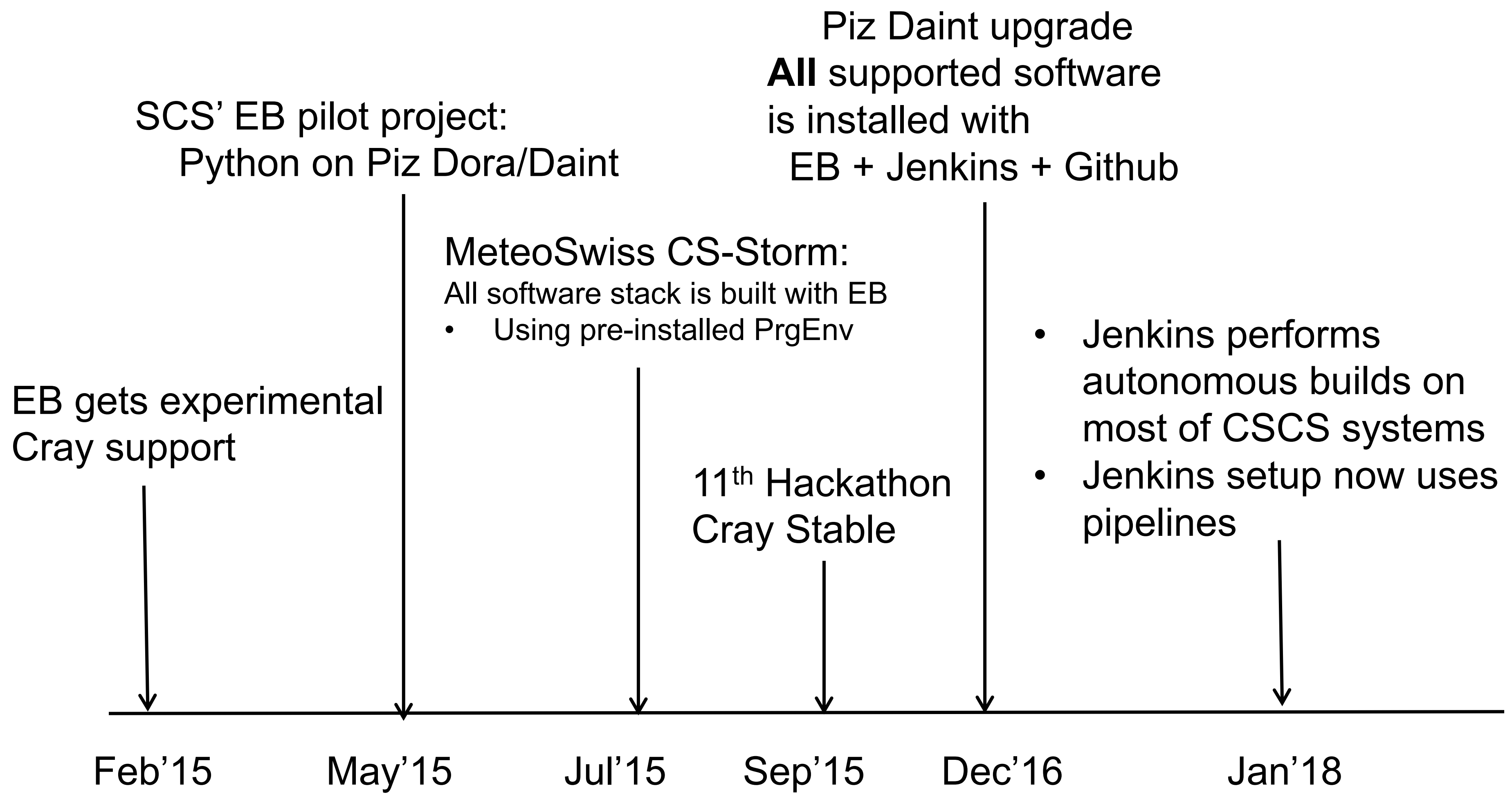
- Distributed source code management (SCM) and CI are the state of art for developing high quality software
  - Goal: improve quality while reducing the testing overhead

- Why not using these practices for a fully automated deployment of scientific software?

Tools used at CSCS for deploying scientific software

- EasyBuild
  - Strong focus on stability and has regular releases
  - Reproducible & community validated build recipes
  - Cray support since 2015 (presented at CUG'16)
- Github
  - Standard way of performing distributed source code management
  - Version control, code review (pull requests) and CI hooks
- Jenkins
  - Enable automated testing across different systems
    - Triggered by Pull Requests
- ReFrame
  - Portable framework for writing regression tests for HPC systems

EasyBuild timeline @ CSCS



Feb'15 — EB gets experimental Cray support

May'15 — SCS' EB pilot project: Python on Piz Dora/Daint

Jul'15 — MeteoSwiss CS-Storm:
All software stack is built with EB
- Using pre-installed PrgEnv

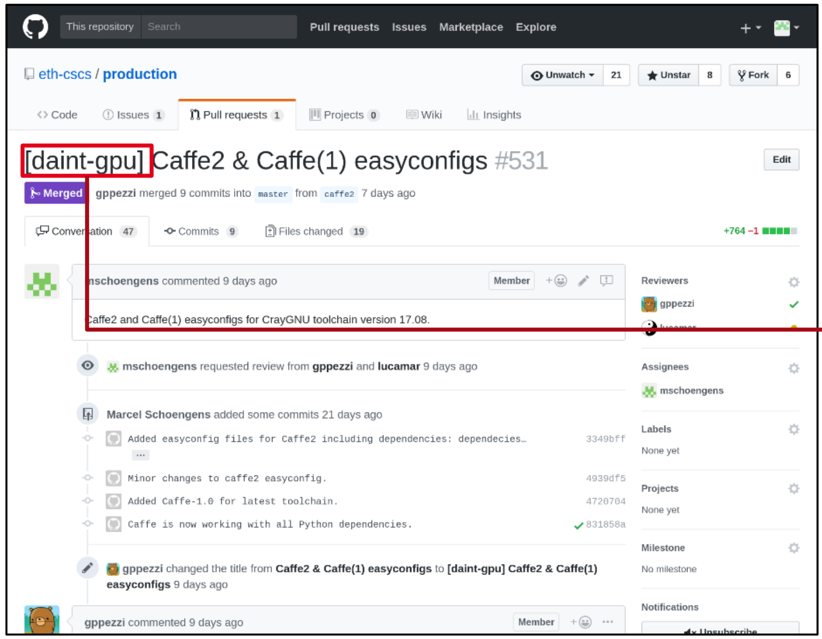
Sep'15 — 11th Hackathon Cray Stable

Dec'16 — Piz Daint upgrade
All supported software is installed with EB + Jenkins + Github

Jan'18
- Jenkins performs autonomous builds on most of CSCS systems
- Jenkins setup now uses pipelines

CI – Testing a new Easyconfig submitted to Github (1)

Github PR ([daint-gpu])



eth-cscs / **production**

[daint-gpu] Caffe2 & Caffe(1) easyconfigs #531

Merged **gppezzi** merged 9 commits into `master` from `caffe2` 7 days ago

mschoengens commented 9 days ago

Caffe2 and Caffe(1) easyconfigs for CrayGNU toolchain version 17.08.

mschoengens requested review from **gppezzi** and **lucamar** 9 days ago

Marcel Schoengens added some commits 21 days ago

```
Added easyconfig files for Caffe2 including dependencies: dependecies…   3349bff
Minor changes to caffe2 easyconfig.                                      4939df5
Added Caffe-1.0 for latest toolchain.                                    4720704
Caffe is now working with all Python dependencies.                     ✓ 831858a
```

gppezzi changed the title from **Caffe2 & Caffe(1) easyconfigs** to **[daint-gpu] Caffe2 & Caffe(1) easyconfigs** 9 days ago

gppezzi commented 9 days ago

Reviewers: gppezzi ✓, lucamar

Assignees: mschoengens

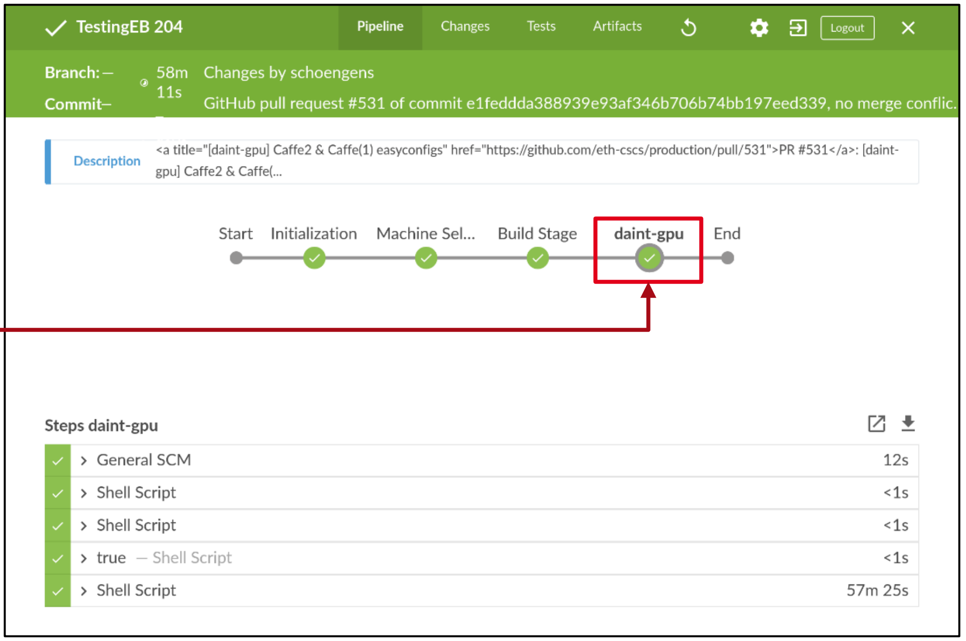
Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

Triggered Pipeline



✓ TestingEB 204

Branch: —
Commit—

58m 11s

Changes by schoengens

GitHub pull request #531 of commit e1feddda388939e93af346b706b74bb197eed339, no merge conflic.

Description <a title="[daint-gpu] Caffe2 & Caffe(1) easyconfigs" href="https://github.com/eth-cscs/production/pull/531">PR #531</a>: [daint-gpu] Caffe2 & Caffe(...

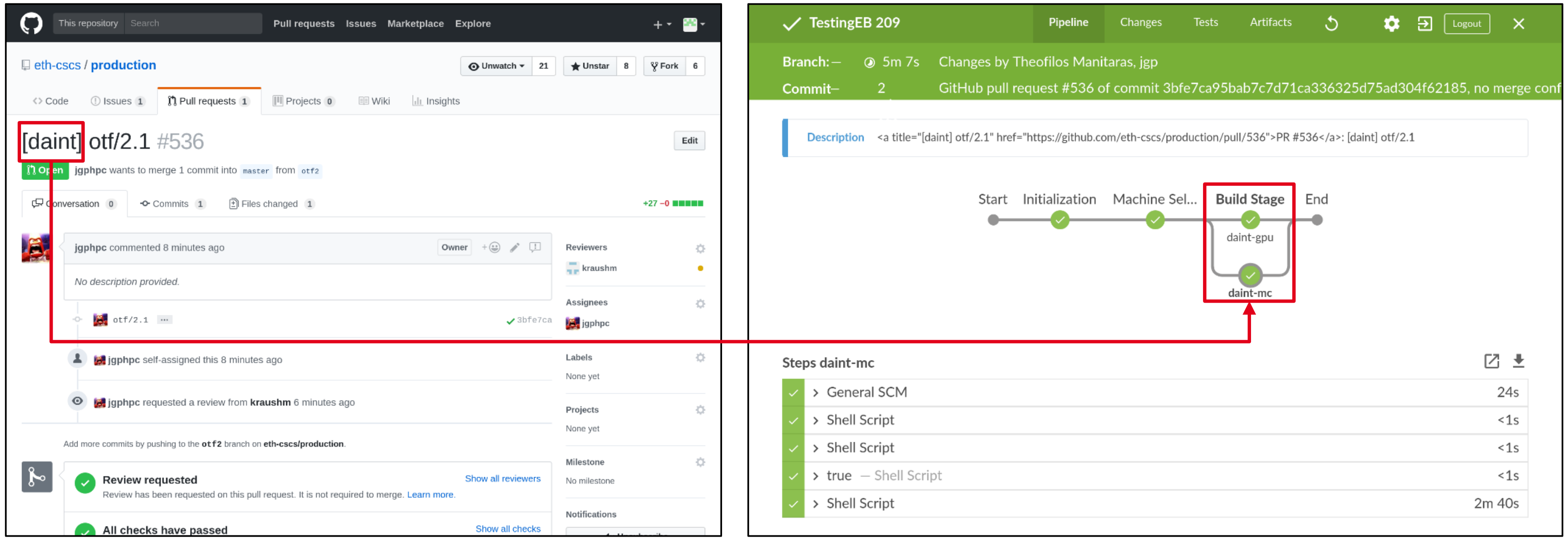
Steps daint-gpu

✓	General SCM	12s
✓	Shell Script	<1s
✓	Shell Script	<1s
✓	true — Shell Script	<1s
✓	Shell Script	57m 25s

CI – Testing a new Easyconfig submitted to Github (2)

Github PR ([daint])

Triggered Pipeline


This repository Search Pull requests Issues Marketplace Explore

eth-cscs / **production**

Unwatch 21 Unstar 8 Fork 6

Code Issues 1 Pull requests 1 Projects 0 Wiki Insights

[daint] otf/2.1 #536

Open **jgphpc** wants to merge 1 commit into `master` from `otf2`

Conversation 0 Commits 1 Files changed 1 +27 −0

jgphpc commented 8 minutes ago Owner

No description provided.

`otf/2.1` ✓ `3bfe7ca`

jgphpc self-assigned this 8 minutes ago

jgphpc requested a review from **kraushm** 6 minutes ago

Add more commits by pushing to the **otf2** branch on **eth-cscs/production**.

Review requested Show all reviewers
Review has been requested on this pull request. It is not required to merge. Learn more.

All checks have passed Show all checks

Reviewers: kraushm
Assignees: jgphpc
Labels: None yet
Projects: None yet
Milestone: No milestone
Notifications

✓ TestingEB 209 Pipeline Changes Tests Artifacts Logout

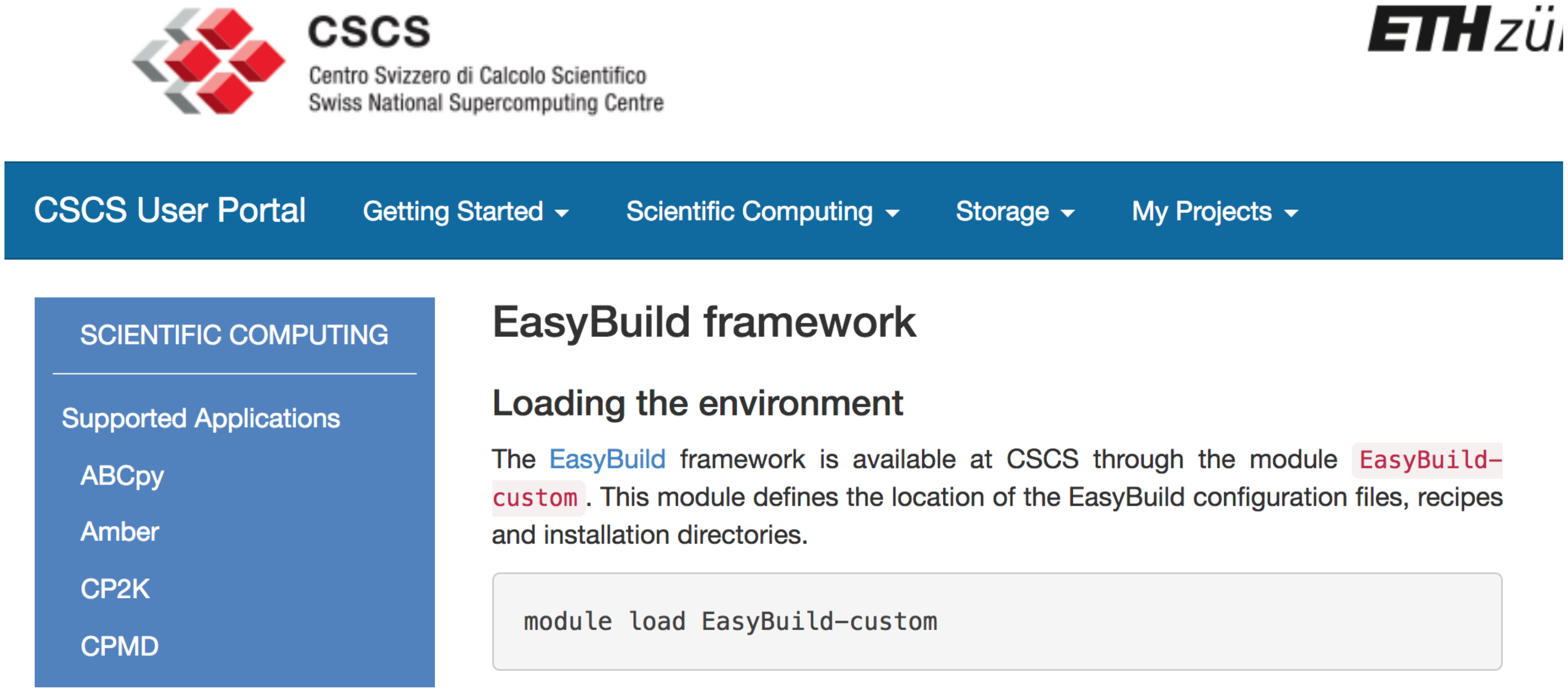
Branch: — 5m 7s Changes by Theofilos Manitaras, jgp
Commit: — 2 GitHub pull request #536 of commit 3bfe7ca95bab7c7d71ca336325d75ad304f62185, no merge conf

Description <a title="[daint] otf/2.1" href="https://github.com/eth-cscs/production/pull/536">PR #536</a>: [daint] otf/2.1

Steps daint-mc

✓	General SCM	24s
✓	Shell Script	<1s
✓	Shell Script	<1s
✓	true — Shell Script	<1s
✓	Shell Script	2m 40s

User instructions for EasyBuild



EasyBuild framework

Loading the environment

The EasyBuild framework is available at CSCS through the module `EasyBuild-custom`. This module defines the location of the EasyBuild configuration files, recipes and installation directories.

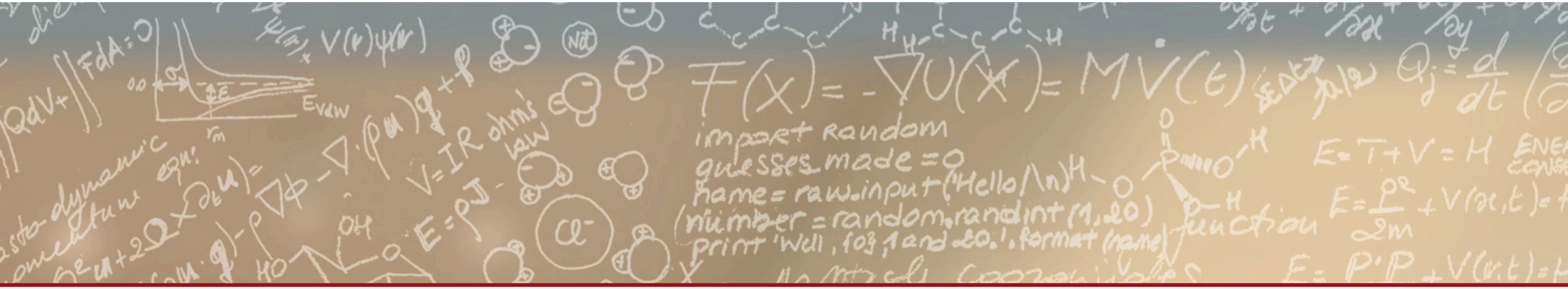
```
module load EasyBuild-custom
```

Useful links for EasyBuild @ CSCS

- EasyBuild User Documentation at CSCS
  - https://user.cscs.ch/scientific_computing/code_compilation/easybuild_framework/
- Easyconfig files repositories
  - List of production builds performed by Jenkins
    - https://github.com/eth-cscs/production/tree/master/jenkins-builds
  - Custom CSCS easyconfigs and easyblocks:
    - https://github.com/easybuilders/CSCS
  - ReFrame
    - https://eth-cscs.github.io/reframe/
- Acknowledgments
  - Scientific Computing Support team

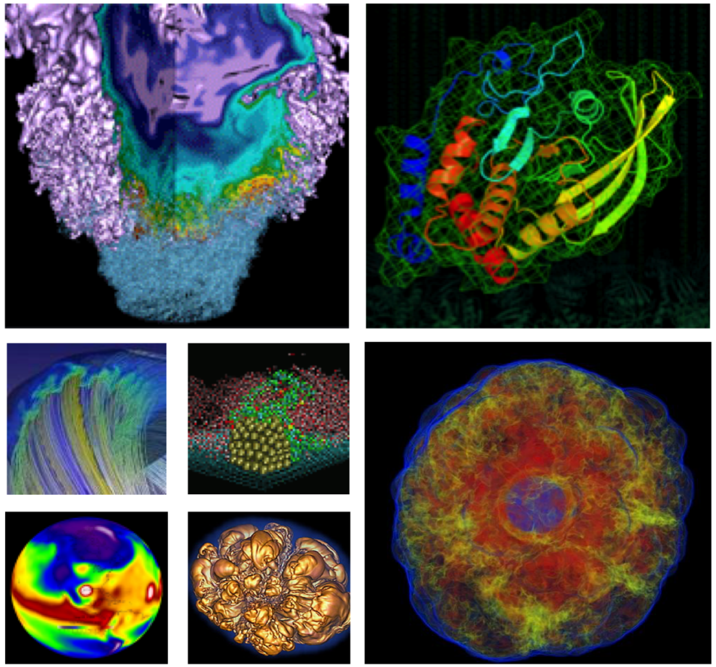
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETHzürich



Thank you for your kind attention

Maintain and Support User Software Environment at NERSC

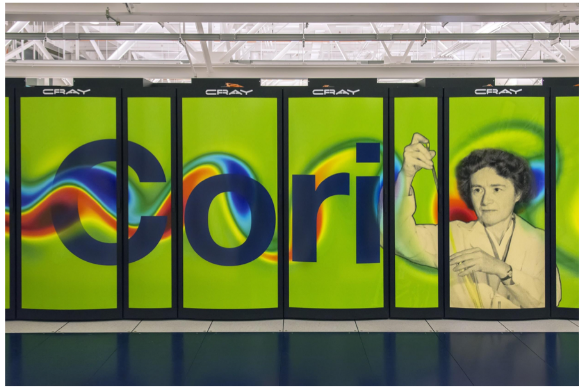


Mario Melara and Helen He
User Engagement Group

May 22nd, 2018
CUG 2018

Who We Are: NERSC

- Cori (Cray-XC40)
  - 34 double-width cabinets
  - 9,688 KNL + 2,388 Haswell nodes
  - Aries-High-Speed Network
  - Intel/Cray/GNU
- Edison (Cray-XC30)
  - 30 cabinets
  - 5,586 Ivybridge nodes
  - 12 Sandybridge login nodes
  - Intel/Cray/GNU
- Genepool (soon to be deprecated) and Denovo



NERSC Software Environment

- Among three systems NERSC maintains: 700+ software packages
  - Includes different:
    - architectures - haswell, mic-knl, sandybridge
    - compilers - gcc, intel, cce
- Different categories:
  - High Energy Physics
  - Climate
  - Bioinformatics
  - Libraries
  - Machine Learning/Deep Learning libs and frameworks
  - Debugging tools

Software Environment Policy

NERSC has established a software programming environment policy:

- Maintain 4 CDT (Cray Development Tools) versions on the system at a time
    - Users have access to any given software version for at least 18 months
- Install new CDT every 3 months (Mar, Jun, Sept, Dec releases)
- Change software default versions every 6 months
    - Mostly for changing Cray provided software versions
    - Promotes to rebuild NERSC supported packages with these new defaults
- Fix critical bugs as needed with user notices
- Provide pe_archive module for archived older versions

Package Installation without Spack

- Software Owners List for each System
  - Consultant assigned a package to install and maintain
  - Consultant usually has experience installing (and maybe using) package
- Bash scripts
    - Each maintained their own
    - No central repository for scripts
- No documentation
- Python – Anaconda
  - Conda install
  - Virtual environments

Choosing Spack

- Easy to install and use
  - Git clone → $ spack install → installed!
- Lawrence Livermore National Laboratory close to us and easy to collaborate with
  - At the time Spack did not have Cray support but now does thanks to close work with Spack developers
- Powerful package database querying system
  - $ spack find
- Allows for combinatorial installs
  - Can install the same packages with different features turned on and off
  - No interference with other installs
- Rapid adoption from other facilities
  - OLCF
  - ALCF

Software Management Today

- Started using Spack for production software June-2017 on Edison
- Spack-built packages on Cori and Edison (700+ packages each for different architecture)
- Some software still done by with bash scripts
  - Usually the difficult to manage scientific applications (CP2K)
- Cross compilation still difficult
- Python
  - Managed separately through anaconda and environments
- Users sometimes told to install their own package using Shifter
  - Some build systems are difficult to use on Cray's compilation environment
- Entire software stack is static
- Spack modulefiles not used
  - Some NERSC custom logic in modulefiles (moving to the auto-generated ones)
  - In-house script to generate modulefiles.

Spack Manages Software Through SWOwner

- SWOwner
  - Pseudo-user with an elevated access (still no root privileges)
  - Consultants login as “swowner” (using yubikey)
  - Single spack instance
- Default configuration files
  - packages.yaml - linking with external Cray tools and libraries
  - config.yaml - for customizing install path format and location
- Run a spack install or transfer their custom bash script to pseudo-user
- Create modulefile using in-house script
  - make_modulefiles.sh
- Collaboration between consultants
  - Done via PRs
  - Each consultant maintains their own fork and works on their own package

Impressions of Spack

- Pros
  - Spack makes installing libraries and large software stacks easier
  - Community is very active
  - Close collaboration with developer
  - Spack is easy to use
  - Combinatorial installs
- Cons
  - Spack re-builds a lot of packages
  - Cross compilation not possible at the moment
    - NERSC is working on this!
  - Static linking support is lacking
    - Most package files, like Linux systems, assume shared linking.
    - This does not work well on Cray.

NERSC/Spack Roadmap

- Improve cross-compilation
- Improve linking of dependencies on Cray systems
- Automation of the creation of modulefiles
  - Use spack built modulefiles instead of creating our own
- Provide spack as a module for users.
  - Users can use Spack to install packages
  - Developers part of the Exascale Computing Project (ECP) can use Spack to install

Thank You