Software Deployment at Facilities

E4S at DOE Facilities with Deep Dive at NERSC
Oct 4, 2021

Ryan Adamson
2.4.4 L3 Lead
Overview:

ECP Software Deployment at Facilities (SD) Activity
2.4.4 Software Deployment at Facilities - Mission

Each software team and facility has individual (and often unique!) preferences, constraints, and strategies regarding scientific software development and deployment.

Integration of ECP ST products and facility capabilities:

- Maintain facility-based continuous integration platforms for build and test automation
- Deploy E4S-curated ST products using the Spack package manager
- Evaluate vendor system software

Extreme Scale Scientific Software Stack

Spack

GitLab / CI

Vendor

HW / SW

ALCF

OLCF

NERSC
### WBS 2.4.4: Software Deployment at Facilities

<table>
<thead>
<tr>
<th>Project Short Name</th>
<th>PI Name, Inst</th>
<th>Short Description/Objective</th>
<th>Program Office(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.4.01 Software Integration</td>
<td>Shahzeb Siddiqui (LBL)</td>
<td>Build/Test/Deploy ST products at facilities</td>
<td>ASCR</td>
</tr>
<tr>
<td>2.4.4.04 Continuous Integration</td>
<td>Paul Bryant (ORNL)</td>
<td>Develop and Deploy ECP CI infrastructure</td>
<td>ASCR</td>
</tr>
<tr>
<td>2.4.4.03 Shasta Testing</td>
<td>Jay Srinivasan (LBL)</td>
<td>Evaluate and expedite Shasta releases</td>
<td>ASCR</td>
</tr>
<tr>
<td>2.4.4.05 HPCM / Slingshot Testing</td>
<td>Scott Atchley (ORNL)</td>
<td>Explore HPCM and Slingshot as alternatives to Shasta</td>
<td>ASCR</td>
</tr>
</tbody>
</table>
Integrated testing strengthens the ECP software ecosystem

Software Deployment Efforts

- Software Integration interfaces with the Testing Task Force, E4S, and ST to adapt builds, tests, and deployments to facility systems
- Continuous Integration Infrastructure supports the automation of activities on facility systems
- Slingshot and System Software testing provides insight into facility hardware from a system functionality perspective
- The Testing Task Force is the vertical integration of all layers of the ECP software ecosystem
Software Integration:

Integrating the ECP Software Ecosystem with Facility Systems
Software Integration Team

Frank Willmore  Aditya Kavalur  Shahzeb Siddiqui  Matt Belhorn  Jamie Finney

Ryan Adamson

2.4.4.01 Software Integration
2.4.4.04 Continuous Integration
2.4.4.03 Shasta Testing
2.4.4.05 HPCM/Slingshot Testing
ECP Software Ecosystem Overview

AD and ST teams implement ECP CI regression testing as appropriate into existing CI frameworks and merge new build/test recipes into spack/develop.

E4S Team Prepares Quarterly Releases for Facility Installation, freezing on a point-in-time commit of spack/develop.

AD teams access stable ST software through Facility-maintained modules. Build cache and config available to AD/ST teams for integrating hotfixes.

SI Team Deploys tested, versioned E4S on Facility Resources.

**Continuous Integration (2.4.4.04)**

**Software Integration (2.4.4.01)**
## Roles and Responsibilities within the ECP Ecosystem

### ST Developers
- AD and ST teams implement ECP CI regression testing as appropriate into existing CI frameworks and merge new build/test recipes into spack/develop
  - [https://e4s-project.github.io/policies.html](https://e4s-project.github.io/policies.html)
- These software development teams typically have other automated CI pipelines, developer-driven unit testing, and other software assurance best practices as determined by their internal policies
  - [https://ideas-productivity.org/ideas-ecp/](https://ideas-productivity.org/ideas-ecp/)

### E4S Software Curators
- Spack Team implements CI build and smoke testing for spack/develop
  - [https://github.com/spack/spack](https://github.com/spack/spack)
  - [https://cdash.spack.io](https://cdash.spack.io)
- E4S Team Prepares Quarterly Releases for Facility Installation, freezing on a point-in-time commit of spack/develop
  - [https://github.com/E4S-Project/e4s/tree/master/environments/21.05](https://github.com/E4S-Project/e4s/tree/master/environments/21.05)

### SI / Facilities
- SI Team Deploys packages selected from a tested and versioned E4S release onto Facility Resources.
  - [https://docs.nersc.gov/applications/e4s/cori/21.05/](https://docs.nersc.gov/applications/e4s/cori/21.05/)
- Facility software and operations teams have additional verification and validation tests that are performed according to facility software and operations policies
  - [https://www.exascaleproject.org/event/buildtest-21-09/](https://www.exascaleproject.org/event/buildtest-21-09/)
Five types and six layers of testing for the ECP Software Ecosystem

<table>
<thead>
<tr>
<th>Test type</th>
<th>Complexity</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Low</td>
<td>Build tests check that all software components have been built successfully by using the appropriate compilers and tunables.</td>
</tr>
<tr>
<td>Validation</td>
<td>Low</td>
<td>Validation tests are run to demonstrate that a piece of software is installed correctly and to clearly demonstrate usage of the appropriate unique hardware features of the system, such as accelerators or high-speed interconnects.</td>
</tr>
<tr>
<td>Integration</td>
<td>Medium</td>
<td>Integration tests are implemented to ensure that versions of libraries along with compile time options do not conflict adversely with other related libraries as part of a larger software ecosystem.</td>
</tr>
<tr>
<td>Regression</td>
<td>High</td>
<td>Regression tests are maintained and run as frequently as developers create new software changes to determine whether the changes impact any already working features of the software.</td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
<td>Performance tests are implemented to detect regressions in runtime or degradation in time to solution or network bandwidth.</td>
</tr>
</tbody>
</table>

Teams and Testing Assurance

- ST teams implement all test types on developer, cloud, or product specific build and test CI pipelines
- ST teams run selected, high-value tests of all test types on facility CI platforms
- Spack CI pipelines test builds of packages in the cloud and at some facilities
- E4S team performs build, validation, and integration tests on facility systems and elsewhere
- SI teams run build, validation, and integration tests on facility systems
- Facility users run validation tests such as ‘spack test’ and the E4S test suite
E4S Documentation is Live!

- E4S User documentation is now available at [https://e4s.readthedocs.io/en/latest/](https://e4s.readthedocs.io/en/latest/)

- The E4S Product Dictionary is now available with list of E4S products with CI badges, reference to Spack package and assigned maintainers

- E4S Facility Dashboard outlines E4S deployments at the facility and anyone can contribute to this page provided they have a deployment of E4S

E4S Nightly Build Pipeline

• SI maintains a CI dashboard for E4S nightly builds which reports the project, maintainers, and CI badges. This can be found at [https://software.nersc.gov/ecp/e4s-ci-pipelines/dashboard](https://software.nersc.gov/ecp/e4s-ci-pipelines/dashboard).

• All projects are located at [https://software.nersc.gov/ecp/e4s-ci-pipelines/](https://software.nersc.gov/ecp/e4s-ci-pipelines/) with a project name per spack package.

### E4S CI Pipeline Dashboard

This project provides a snapshot of all E4S products that have a nightly build pipeline on NERSC system. If you would like to have a nightly build of your product, please contact Shahzeb Siddiqui ([shahzebsiddiqui@lbl.gov](mailto:shahzebsiddiqui@lbl.gov)) to create a project space with the name of your spack package that you plan to install.

Please refer to [https://docs.nersc.gov/applications/e4s/spack_github_pipeline/](https://docs.nersc.gov/applications/e4s/spack_github_pipeline/) for documentation on Spack Gitlab Pipeline to get started.

<table>
<thead>
<tr>
<th>E4S Product</th>
<th>Maintainer</th>
<th>Pipeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>amrex</td>
<td></td>
<td>pipeline</td>
</tr>
<tr>
<td>conduit</td>
<td>cynush</td>
<td>pipeline</td>
</tr>
<tr>
<td>darshan</td>
<td>snyder</td>
<td>pipeline</td>
</tr>
<tr>
<td>hdf5</td>
<td>lknox</td>
<td>pipeline</td>
</tr>
<tr>
<td>kokkos</td>
<td></td>
<td>pipeline</td>
</tr>
<tr>
<td>tau</td>
<td>lpayraja</td>
<td>pipeline</td>
</tr>
</tbody>
</table>
There is a **e4s/21.05** modulefile on Cori with up to 157 installed specs built with **intel@19.1.3.304** compiler.


Deployment issues are tracked in an ECP software deployment specific repository, for example: [https://gitlab.com/ecp-swd/issuetracking/-/issues/35](https://gitlab.com/ecp-swd/issuetracking/-/issues/35)

### 1.2. E4S Facility Deployment Dashboard

<table>
<thead>
<tr>
<th>E4S Version</th>
<th>Gitlab Project</th>
<th>Installed Specs</th>
<th>Compiler</th>
<th>Spack commit</th>
<th>Spack.yaml</th>
<th>Installed Specs</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.02</td>
<td><a href="https://software.nersc.gov/NERSC/e4s-2102">https://software.nersc.gov/NERSC/e4s-2102</a></td>
<td>149</td>
<td>intel@19.1.2.254 and gcc@10.1.0</td>
<td><a href="https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-2102/spack.yaml">https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-2102/spack.yaml</a></td>
<td><a href="https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-2102/spack.yaml">https://github.com/spack/spack-configs/blob/master/NERSC/cori/e4s-2102/spack.yaml</a></td>
<td></td>
</tr>
</tbody>
</table>
Installed packages of E4S 21.05 are discoverable with Spack on Cori
Continuous Integration:

Automating builds and tests of the ECP Software Ecosystem
There are three main use cases for CI on facility systems

1) Software Development Testing
   - Regression Testing
     - Software failures are detected when new code is introduced. This prevents latent bugs from existing well after a feature has been implemented.
     - Correctness of results is assessed by examining changes in the output of a well-understood and tested problem.
   - Performance Testing
     - Performance regressions are caught in the exact same environment where performance is important.

2) Ecosystem Integration
   - Extreme Scale Scientific Software Stack (E4S) integration tests
     - Regular builds of spack versions and release candidates of E4S ensure that facility software integrators will not experience issues during installation.
     - Recurring tests of E4S team installed software stacks detect underlying issues that change over time.
   - Individual software ‘build’ recipes
     - Individual products can be built periodically on facility systems and failures addressed by developers or integrators as appropriate.

3) Facility Operations Assurance
   - Regular tests of installed facility software stack
     - Environments drift over time. CI can catch issues related to hardware, vendor PE, or other Facility service updates.
   - Regular tests of user managed container images
     - Security assessments can be automated with CI pipelines to inspect container images, run static analysis tools, and pass security unit tests.

There is a Value Per Cycle tradeoff
### Runner/Server Capability – Deployment Overview

<table>
<thead>
<tr>
<th>Resource</th>
<th>CI Status</th>
<th>GitLab Ver.</th>
<th>Jacamar CI Ver.</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLCF - Ascent</td>
<td>Functional and available to ECP</td>
<td>14.1+</td>
<td>0.8.0</td>
<td>Starter</td>
</tr>
<tr>
<td>OLCF - Spock</td>
<td>CI deployed and planning ECP access</td>
<td>14.1+</td>
<td>0.8.0</td>
<td>Premium</td>
</tr>
<tr>
<td>ALCF - JLSE</td>
<td>Functional and available to ECP</td>
<td>14.1+</td>
<td>0.8.2</td>
<td>Premium</td>
</tr>
<tr>
<td>ALCF – Theta</td>
<td>Functional and available to ECP</td>
<td>14.1+</td>
<td>0.8.2</td>
<td>Starter</td>
</tr>
<tr>
<td>NERSC - Cori</td>
<td>Functional and available to ECP</td>
<td>14.1+</td>
<td>N/A</td>
<td>Core</td>
</tr>
</tbody>
</table>
A typical month of CI at facilities (Apr 2021 – May 2021)

<table>
<thead>
<tr>
<th>Apr 2021 – May 2021</th>
<th>ALCF</th>
<th>NERSC</th>
<th>OLCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Projects</td>
<td>23</td>
<td>86</td>
<td>36</td>
</tr>
<tr>
<td>Average Job Times</td>
<td>1-120 minutes</td>
<td>1-120 minutes</td>
<td>1-120 minutes</td>
</tr>
<tr>
<td>Total Jobs Executed</td>
<td>2773</td>
<td>3584</td>
<td>3837</td>
</tr>
<tr>
<td>Total Hours Spent</td>
<td>1553</td>
<td>46000</td>
<td>~1500 (estimated)</td>
</tr>
<tr>
<td>Tickets Per Week</td>
<td>~3</td>
<td>~2</td>
<td>~2</td>
</tr>
<tr>
<td>Special Queue Policies</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Resource Adequacy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Jobs are generally short, and a couple of projects have longer running CI pipelines
- Queue policies are sufficient for starting out, but tuning can make CI ‘easier’ for certain workloads
- Ticket load is reasonable, and typically about 10 or so a month
- Resources are sufficient for the workloads being performed at the moment
- Software Development CI users typically self regulate and find cycles elsewhere where they can
E4S: Better Quality, Documentation, Test, Integration, Delivery, Build & Use

Delivering HPC software to facilities, vendors, agencies, industry, international partners in a brand-new way

- **Community Policies**
  Commitment to software quality

- **DocPortal**
  Single portal to all E4S product info

- **Portfolio testing**
  Especially leadership platforms

- **Curated collection**
  The end of dependency nightmares

- **Quarterly releases**
  Release 21.08 – August

- **Build caches**
  10X build time improvement

- **Turnkey stack**
  A new user experience

- **https://e4s.io**

- **E4S Strategy Group**
  US agencies, industry, international
Thank you

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy’s Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation’s exascale computing imperative.

Thank you to all collaborators in the ECP and broader computational science communities. The work discussed in this presentation represents creative contributions of many people who are passionately working toward next-generation computational science.

https://www.exascaleproject.org
Questions?