E4S: Extreme-scale Scientific Software Stack

M. A. Heroux, J. Willenbring S. Shende, C. Coti, W. Spear, L. Peyralans, J. Skutnik, E. Keever
Sandia National Laboratories, NM University of Oregon, Eugene, OR
{maheroux, jmwille}@sandia.gov {sameer,coti,wspear,lpeyralans,jskutnik,ekeever1}@uoregon.edu
https://e4s.io

Introduction

Building applications by composing existing libraries and using existing tools can be a tremendous productivity improvement. If existing software is high quality, accessible and reusable, one would be much better off using it than writing your own. At the same time, as HPC and AI/ML software gets more complex, it is getting harder to maintain, install, and optimize tools and libraries correctly in an integrated and interoperable software stack.

The Extreme-scale Scientific Software Stack (E4S) project [1] aims to tame both the complexity and portability problems by creating an ecosystem of numerical libraries, runtime systems, and tools that lowers the barrier for entry for the HPC and AI/ML developer communities. E4S is a community effort to provide open source software packages for developing, deploying, and running scientific applications on HPC platforms. It aims to deliver a modular, interoperable, and deployable software stack based on the Spack [2] package manager. E4S provides both source builds for native, bare-metal installations, as well as containers of a broad collection of software packages for secure, reproducible, container-based deployments. E4S exists to accelerate the development, deployment, and use of HPC software, lowering the barriers for HPC developers and users.

Motivation

With the growing complexity of hardware and software frameworks, it is getting harder to maintain and install a software stack to satisfy the growing demands of HPC and AI/ML application developers. Applications are being distributed in source form and typically need to be built using a complex set of third-party numerical libraries, runtime systems, and tools. These applications are typically tied down to a given platform and cannot migrate easily to other sites that may be running different versions of the underlying operating system, compilers, and runtime libraries.

A crisis point has been reached where supercomputing centers are employing teams of system professionals just to maintain and upgrade software packages, largely because the software dependencies for such applications typically run deep – down to the specific versions of the Kernel, glibc library and runtime communication library. Significant system resources are now being spent to maintain the dependency tree associated with each application. In many cases, applications use different versions of or options for certain components within the software stack, leaving the choice between facilities supporting many versions of some codes, or more often, computational scientists performing custom installations of some of their required software dependencies. As APIs evolve, incompatibilities arise, and it is getting increasingly difficult for computational scientists to build and maintain a consistent software stack. As the

complexity of our software and hardware grows, it becomes difficult to reproduce many published scientific results, even for the original authors!

Components of E4S

The E4S project, supported by the DOE through the Exascale Computing Project (ECP) [3], is an effort to build a complete software stack consisting of a broad set of packages that are commonly used by HPC and AI/ML developers. E4S builds upon Software Development Kits (SDKs), which are collections of related software products and packages where coordination across package teams improves usability and practices, and fosters community growth among teams that develop similar and complementary capabilities.

E4S itself is a curated release of ECP Software Technology (ST) products built using Spack. Spack is an open-source package manager geared towards simplified, reproducible builds of the otherwise complicated dependency chains common in HPC software, and provides the ability to leverage existing compilers and runtime system libraries for native software installations. E4S currently includes over 50 unique ST products spanning programming models and run times, math libraries, performance evaluation tools, compilers, data, and visualization tools. It also provides a validation test suite that helps build, execute, and validate these products.

The workflow of launching applications using containers is simplified in E4S, and leverages performant, high-speed, inter-node networking using the system MPI libraries. To simplify interoperation between containers and host MPI libraries, E4S is developing techniques to substitute MPI libraries in the containerized applications with compatible MPI libraries at runtime during the launch of containers (Figure 1). This technique helps reproducibility and makes it practical to develop applications on a local environment (desktop/laptop) and then transfer the binary executable to a supercomputer, deploying it using a secure container runtime (such as Shifter or Singularity) while being able to leverage high speed interconnects.

E4S provides both a VirtualBox image with support for Docker, Singularity, Shifter, and Charliecloud HPC container runtimes, as well as an AWS image to deploy E4S on EC2 instances. E4S also supports performance portability using a binary build-cache of Spack based packages to simplify the installation of dependencies for native, bare-metal installation of software. It supports GPUs from multiple vendors and includes their supporting runtime libraries such as CUDA and ROCm.

Deploying complex software without testing can lead to reduced confidence in the reliability of the software. The E4S validation test suite is used in a sophisticated GitLab-based continuous integration (CI) system, as shown in Figure 2, that builds E4S packages on multiple platforms using container build pipelines. E4S is the first official “distribution” of Spack in which pull requests (PR) are automatically tested against how they affect E4S builds. The Spack project builds on AWS Cloud as part of Spack PR and release testing. A CI dashboard [4] is available

to monitor the progress. Efforts are underway to stand up E4S CI at DOE facilities (such as ORNL and NERSC) using their preferred compilers and runtime libraries.

Binary caches of E4S software are available to users of Spack and this enables users to quickly use common variants of E4S packages without having to build them locally from the source code, as shown in Figure 3. These packages are built as part of the CI process and are stored in publicly accessible build caches hosted on AWS S3 and the University of Oregon's S3 servers.

A web portal, modeled after the Homebrew Formulae page, is being developed. This will allow potential users to discover what variants of E4S software exist in the build cache. An E4S DocPortal aggregates and summarizes documentation and metadata by raking ST product repositories. This DocPortal provides a single online location for accurate product descriptions for ECP ST products.

Summary

E4S is **not** a monolithic, take-it-or-leave-it software behemoth or a commercial product, or a simple packaging of existing DOE software. It is an extensible, open architecture software ecosystem, accepting contributions from US and international teams and is a framework for collaborative open-source product integration. It provides a full collection of compatible software capabilities and a manifest of a-la-carte selectable software capabilities. It acts as a vehicle for delivering high-quality reusable software products in collaboration with others and acts as a conduit for future leading-edge HPC software targeting scalable, next-generation computing systems. E4S is a hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.

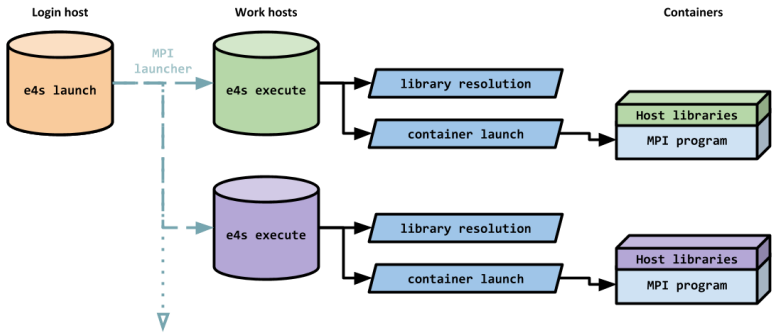

Figure 1: E4S Container Launcher (e4s-cl) tool for launching MPI applications.

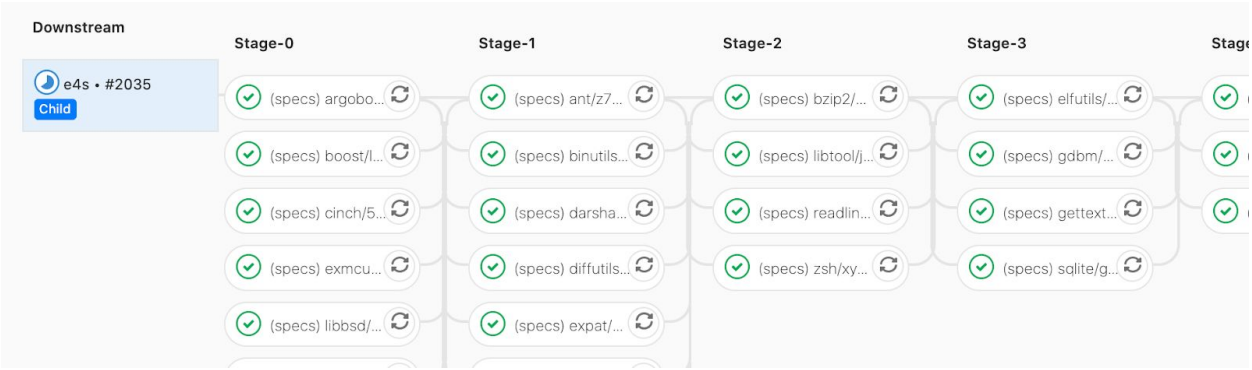

Figure 2. The downstream E4S pipeline triggered by a Spack pull-request.

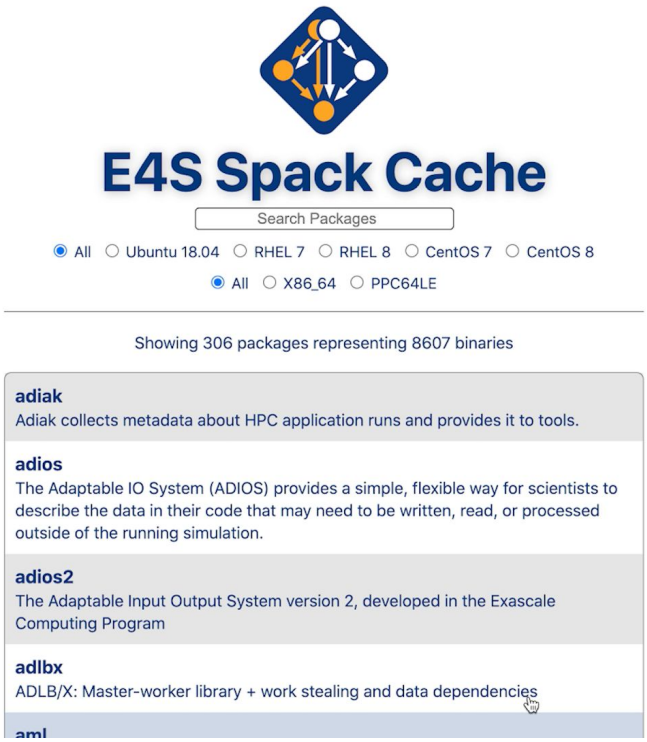

Figure 3: A searchable web-portal for users to discover pre-built E4S binaries in a Spack build cache.

References

[1] E4S: Extreme-scale Scientific Software Stack, https://e4s.io, 2020.
[2] Spack, https://spack.io, 2020.
[3] ECP: Exascale Computing Project, https://www.exascaleproject.org, 2020.
[4] Spack Dashboard: Latest PR Testing - E4S, https://cdash.spack.io, 2020.