**Rough Draft of Wikipedia Article**

**Overview**

The Extreme-scale Scientific Software Stack (E4S) is a community initiative that provides open-source software packages for scientific applications on high-performance computing (HPC) platforms. It includes from-source builds and containers of various HPC software packages (“The Extreme-Scale Scientific Software Stack”). E4S allows contributions from teams around the world and enables collaborative integration of open-source products in addition to offering a variety of software capabilities to choose from. It aims to deliver high-quality reusable software products through collaboration and provides a framework to enhance software interoperability and quality expectations. E4S also targets scalable next-generation computing platforms (“Extreme-Scale Scientific Software Stack (e4s) Release 0.2”). The software is based on Spack and includes ECP software technologies (ST) and related software tested for compatibility and portability across multiple architectures. It aims to distinguish between usability, quality, and community goals for SDK, deployment, and testing purposes. Additionally, E4S will enhance the interoperability of the software development kit (Gibson). The ECP software stack, E4S Release 0.2, is a subset of the full ECP ST software stack and demonstrates the target approach for future delivery. It includes some ECP ST software products that are supported by Spack packages but are not yet interoperable. This release is mainly focused on demonstrating the ST software stack approach, and not all ECP ST software products were included. With each release, more software products will be added until the goal of including all ECP ST software products in a single release is achieved (“Extreme-Scale Scientific Software Stack (e4s) Release 0.2”).

**Interoperability and Software Delivery**

The porting of individual scientific software products is difficult already, but achieving interoperability between packages is even more intractable. To achieve software interoperability, E4S uses a dual-pronged approach consisting of Spack and SDKs (“The Extreme-Scale Scientific Software Stack”). As software becomes increasingly complex, individual users are having a more difficult time deploying HPC tools and libraries in a consistent interoperable software stack. E4S, however, helps solve that issue by allowing the use of containers and being able to deploy these packages on bare metal systems. Spack serves as a package manager for deploying systems using E4S recipes. Alternatively, it is possible to download a container and image from e4s.io and deploy a container based on the chosen image." (Gibson). It is used to manage different versions of products, interoperability, and access to runtimes and compilers. It is tested for both interoperability and portability (Gibson). Spack uses Python to streamline the process of installing libraries and simulations and optimize their performance. Spack also allows users to build code configurations, ensures that all installed software runs correctly, regardless of environment, and streamlines file management. It can additionally install numerous variants of the same build using different message-passing interface (MPI) implementations, different options, and different compilers (“Spack: A Flexible Package Manager for HPC Software”). E4S uses the Spack packages manager for software delivery since it provides the ability to distinguish between versions of software packages that are interoperable and those that are not. Spack is also a common build layer to many software packages outside of Exascale Computing Project Software Technology (ECP ST), which support the achievement and maintenance of interoperability between software packages (“The Extreme-Scale Scientific Software Stack”). Writing package recipes for Spack is quite easy, considering a single file contains a templated recipe for other builds of the same package, meaning recipe authors can differentiate between different versions by using a specification language developed by Gamblin. Spack simplifies the installation of software packages by handling the complexity of connecting them with a consistent set of dependencies. It does this by creating unique configurations for each software package, installing them in unique directories, and using RPATH linking so that each package knows where to find its dependencies. Spack helps ensure compatibility between program modules by guaranteeing one configuration of each library per dependency graph. It enables users to specify dependencies without needing to understand the dependency graph structure, as well as offer an optional concretization feature that fills in missing configuration details based on user/site preference (“Spack: A Flexible Package Manager for HPC Software”). E4S facilitates the testing and use of an abundance of reusable HPC software packages through the use of Spack as a meta-build tool, along with supplying containers of pre-built binaries for Singularity, CharlieCloud, Docker, and Shifter (“The Extreme-Scale Scientific Software Stack”). An ECP ST Software Development Kit is a group of related software tools (called packages) that can be used together to enhance user experience, provide consistent practices, and promote collaboration among teams that create similar or related capabilities. Activities that take place inside an SDK enhance interoperability between products. The Extreme-Scale Scientific Software Development Kit (xSDK) is featured in the initial version (0.2) of E4S, although additional SDKs will be included in future versions (“The Extreme-Scale Scientific Software Stack”). SDKs will help reduce the complexity of delivery by utilizing hierarchical build targets and distributing software integration responsibilities. The goal is to extend the SDK approach to all ECP ST domains since SDKs create a horizontal coupling of software products and teams in addition to creating opportunities that are better, faster, and cheaper. Long-term goals include establishing community policies that enhance best practices sharing, providing a mechanism for shared materials, and enabling community expansion beyond ECP (Gibson).

**Objectives and Challenges**

The objectives of composing a modular, interoperable, and deployable software stack include lowering the barrier to using ST products and enabling facilities to install all or parts of it, allowing for interoperability between the ST products, along with uniform APIs, and establishing an open, hierarchical software architecture that allows for collaboration between agencies and software institutions. These objectives also come with challenges, such as the large, diverse group of ST products, the difference in project management styles, the lack of initial drivers, and the complicated software ecosystem (Heroux). The E4S Software Stack provides a complete HPC software stack that does not significantly increase the number of products in the HPC ecosystem. It provides an efficient way of building, installing, and testing the software, and also improves the stability of the ST stack (Gibson). Users can get E4S on the web page, e4s.io, by downloading the containers and accessing the docker hub and the other build cache. There are also many other available resources and tutorials about E4S on the website (Gibson).

**Features**

The E4S software stack offers a range of open-source libraries and tools for high-performance computing applications, including:

* High-performance numerical libraries
* Productivity tools and frameworks
* Performance analysis and profiling tools
* Machine learning libraries and frameworks
* Data management and visualization tools
* Support for multiple programming languages
* Distributed-memory parallel computation support through MPI, and shared-memory parallel computation support

**Programming Languages**

The E4S software stack includes support for a variety of programming languages.

Most of the packages in the E4S software stack are written in C++, including packages such as Spack and Kokkos. These packages require C++11 or later support.

Other programming languages are also supported in the E4S software stack. Python is a popular language for scientific computing and data analysis, and many E4S packages have Python bindings, including packages like PyTorch and TensorFlow.

In addition to C++ and Python, Fortran is also supported in some packages in the E4S software stack. For example, PETSc is a popular package for solving partial differential equations, and it has optional implementations of some computational kernels in Fortran.

Other programming languages with bindings to E4S packages include C and Matlab. The E4S software stack aims to provide a variety of options for users to work with, regardless of their preferred programming language.

**Software Licenses**

The E4S software stack consists of various packages, each with their own specific software license. Most of the packages in the E4S stack are open-source and licensed under the Apache License, Version 2.0. However, there are some packages that use other open-source licenses, such as the BSD-3-Clause license and the GNU Lesser General Public License (LGPL). Additionally, some packages use proprietary licenses, such as the Intel Math Kernel Library (MKL).

The E4S software stack also includes various dependencies, which have their own licensing terms. For example, the BLAS and LAPACK libraries, which are required dependencies for some packages in the E4S stack, are typically licensed under the BSD-3-Clause license.

**See Also**

* [BLAS: Basic Linear Algebra Subprograms](http://www.netlib.org/blas/)
* [LAPACK: Linear Algebra Package](http://www.netlib.org/lapack/)
* [Message Passing Interface (MPI)](https://www.mpi-forum.org/)
* [High-Performance Computing (HPC)](https://www.cray.com/)
* [Exascale Computing](https://en.wikipedia.org/wiki/Exascale_computing)
* [Sandia National Laboratories](https://www.sandia.gov/)
* [Open-Source Licenses](https://gist.github.com/nicolasdao/a7adda51f2f185e8d2700e1573d8a633)

**References**

1. E4S. “The Extreme-Scale Scientific Software Stack.” <https://e4s-project.github.io/>. Retrieved April 4, 2023.
2. LLNL Computing. “Spack: A Flexible Package Manager for HPC Software.” <https://computing.llnl.gov/projects/spack-hpc-package-manager>. Retrieved March 27, 2023.
3. E4S. “Extreme-Scale Scientific Software Stack (e4s) Release 0.2.” <https://oaciss.uoregon.edu/ecp/>. Retrieved March 28, 2023.
4. Gibson, Scott. “Helping Users Deploy High-Performance Computing Tools and Libraries in a Consistent Interoperable Software Stack.” Exascale Computing Project, 14 Dec. 2021, <https://www.exascaleproject.org/helping-users-deploy-high-performance-computing-tools-and-libraries-in-a-consistent-interoperable-software-stack/>. Retrieved April 2, 2023.
5. Heroux, Mike. “The Extreme-Scale Scientific Software Stack (e4s) and Its Promise for...” 18 Nov. 2019, <https://www.csm.ornl.gov/srt/conferences/Scala/2019/keynote_3.pdf>. Retrieved March 30, 2023.
6. Heroux, Michael A. “The Extreme-Scale Scientific Software Stack (e4s).” The Extreme-Scale Scientific Software Stack (E4S). (Conference) | OSTI.GOV, 1 June 2019, <https://www.osti.gov/servlets/purl/1640619>. Retrieved March 27, 2023.
7. Willenbring, James, and Sameer Shende. “E4s: Extreme-Scale Scientific Software Stack.” *E4S: Extreme-Scale Scientific Software Stack. (Conference) | OSTI.GOV*, 1 Apr. 2021, <https://www.osti.gov/servlets/purl/1882038>. Retrieved April 4, 2023.