E4S: The Extreme-scale Scientific Software Stack for Collaborative Open Source Software

EXASCALE COMPUTING PROJECT

Michael A. Heroux, Sandia National Laboratories
Director of Software Technology, ECP

Sameer Shende, University of Oregon
E4S Technical Lead

Getting Started with E4S for Industry and Agencies Workshop, June 14, 2021

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Brief Intro to the Exascale Computing Project (ECP)

ECP's holistic approach uses co-design and integration to achieve exascale computing

Performant mission and science applications at scale

Aggressive RD&D project	Mission apps; integrated S/W stack	Deployment to DOE HPC Facilities	Hardware technology advances

Application Development (AD)

Develop and enhance the predictive capability of applications critical to DOE

24 applications
National security, energy, Earth systems, economic security, materials, data

6 Co-Design Centers
Machine learning, graph analytics, mesh refinement, PDE discretization, particles, online data analytics

Software Technology (ST)

Deliver expanded and vertically integrated software stack to achieve full potential of exascale computing

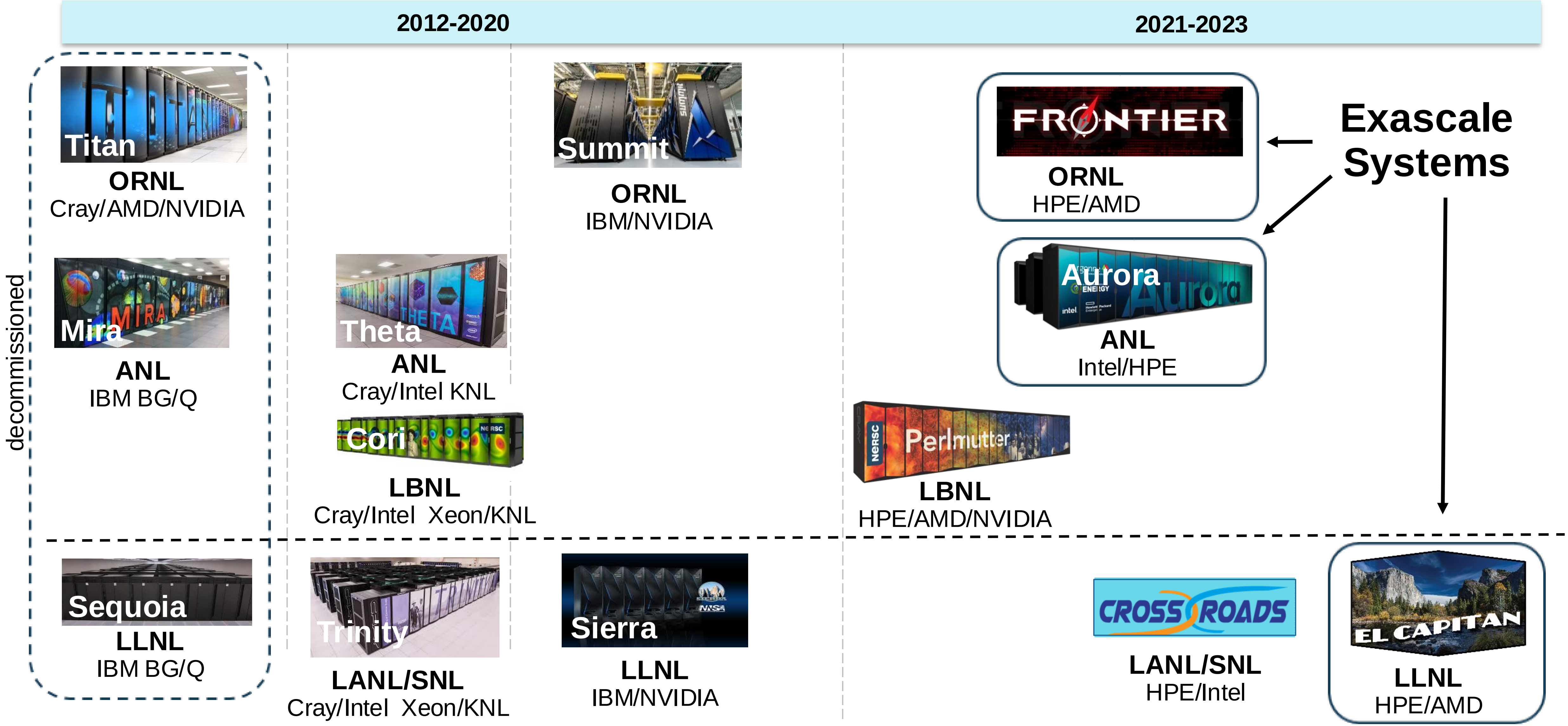
71 unique software products
spanning programming models and run times,
math libraries,
data and visualization

Hardware and Integration (HI)

Integrated delivery of ECP products on targeted systems at leading DOE HPC facilities

6 US HPC vendors
focused on exascale node and system design; application integration and software deployment to Facilities

DOE HPC Roadmap to Exascale Systems



EXASCALE COMPUTING PROJECT

Brief Intro to ECP Software Technology (ST) Focus Area

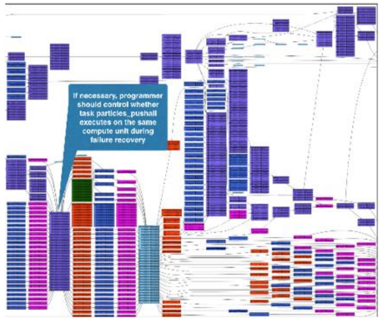
EXASCALE COMPUTING PROJECT

ECP **Software Technology (ST)**

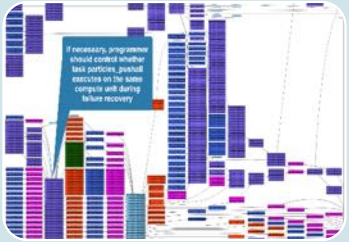
Goal

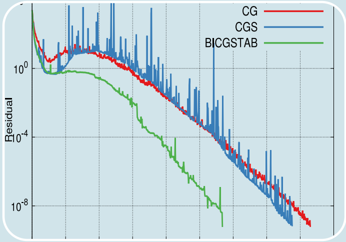
Build a comprehensive, coherent software stack that enables application developers to productively develop highly parallel applications that effectively target diverse exascale architectures

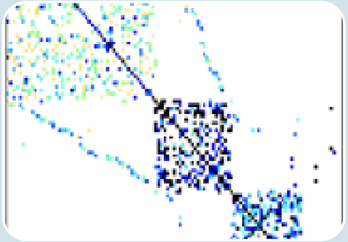
- Prepare SW stack for scalability with massive on-node parallelism
- Extend existing capabilities when possible, develop new when not
- Guide, and complement, and integrate with vendor efforts
- Develop and deliver high-quality and robust software products

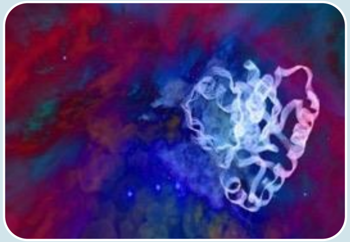


ECP ST has six technical areas









National Nuclear Security Administration

E4S

Programming Models & Runtimes

- Enhance and get ready for exascale the widely used MPI and OpenMP programming models (hybrid programming models, deep memory copies)
- Development of performance portability tools (e.g. Kokkos and Raja)
- Support alternate models for potential benefits and risk mitigation: PGAS (UPC++/GASNet) ,task-based models (Legion, PaRSEC)
- Libraries for deep memory hierarchy and power management

Development Tools

- Continued, multifaceted capabilities in portable, open-source LLVM compiler ecosystem to support expected ECP architectures, including support for F18
- Performance analysis tools that accommodate new architectures, programming models, e.g., PAPI, Tau

Math Libraries

- Linear algebra, iterative linear solvers, direct linear solvers, integrators and nonlinear solvers, optimization, FFTs, etc
- Performance on new node architectures; extreme strong scalability
- Advanced algorithms for multi-physics, multiscale simulation and outer-loop analysis
- Increasing quality, interoperability, complementarity of math libraries

Data and Visualization

- I/O via the HDF5 API
- Insightful, memory-efficient in-situ visualization and analysis – Data reduction via scientific data compression
- Checkpoint restart

NNSA ST

- Open source NNSA Software projects
- Projects that have both mission role and open science role
- Major technical areas: New programming abstractions, math libraries, data and viz libraries
- Cover most ST technology areas
- Subject to the same planning, reporting and review processes

Software Ecosystem

- Develop features in Spack necessary to support all ST products in E4S, and the AD projects that adopt it
- Development of Spack stacks for reproducible turnkey deployment of large collections of software
- Optimization and interoperability of containers on HPC systems
- Regular E4S releases of the ST software stack and SDKs with regular integration of new ST products

We work on products applications need now and into the future

Key themes:
- Focus: GPU node architectures and advanced memory & storage technologies
- Create: New high-concurrency, latency tolerant algorithms
- Develop: New portable (Nvidia, Intel, AMD GPUs) software product
- Enable: Access and use via standard APIs

Software categories:
- **Next generation established products:** Widely used HPC products (e.g., MPICH, OpenMPI, PETSc)
- **Robust emerging products:** Address key new requirements (e.g., Kokkos, RAJA, Spack)
- **New products:** Enable exploration of emerging HPC requirements (e.g., SICM, zfp, UnifyCR)

Example Products	Engagement
MPI – Backbone of HPC apps	Explore/develop MPICH and OpenMPI new features & standards
OpenMP/OpenACC –On-node parallelism	Explore/develop new features and standards
Performance Portability Libraries	Lightweight APIs for compile-time polymorphisms
LLVM/Vendor compilers	Injecting HPC features, testing/feedback to vendors
Perf Tools - PAPI, TAU, HPCToolkit	Explore/develop new features
Math Libraries: BLAS, sparse solvers, etc.	Scalable algorithms and software, critical enabling technologies
IO: HDF5, MPI-IO, ADIOS	Standard and next-gen IO, leveraging non-volatile storage
Viz/Data Analysis	ParaView-related product development, node concurrency

Progress toward Exascale readiness

SLATE port to AMD and Intel platforms

ECP WBS	2.3.3.13 CLOVER (SLATE)
PI	Jack Dongarra, UTK
Members	UTK

Scope and objectives

- SLATE is a distributed, GPU-accelerated, dense linear algebra library, intended to replace ScaLAPACK
- SLATE covers parallel BLAS, linear system solvers, least squares, eigensolvers, and the SVD

Port to AMD and Intel

- SLATE and BLAS++ now support all three major GPU platforms

NVIDIA CUDA   ROCm   oneAPI

Impact

- Initially supported NVIDIA's cuBLAS for use on current machines like Summit
- Can now use AMD's rocBLAS in preparation for Frontier, and Intel's oneMKL in preparation for Aurora
- Other projects can also leverage BLAS++ for portability

Accomplishment

- Refactored SLATE to use BLAS++ as portability layer
- Ported BLAS++ to AMD rocBLAS and Intel oneMKL

Deliverables Report: https://www.icl.utk.edu/publications/swan-016
Code in git repos: bitbucket.org/icl/slate/ and bitbucket.org/icl/blaspp/

Kokkos: Support and AMD Functionality.

ECP WBS	2.3.6.03 – SNL ATDM ST
PI	Christian Trott, SNL
Members	SNL

Scope and objectives

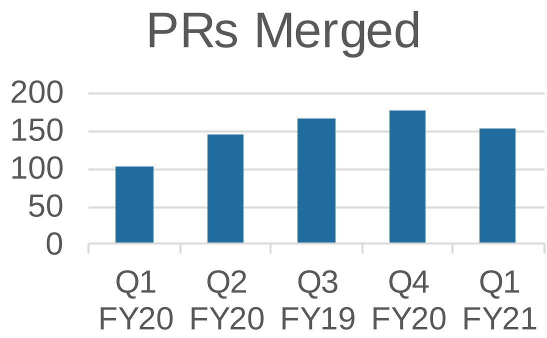
- Kokkos provides the C++ based based Programming Model for Performance Portability for Sandia and many applications at partner institutions
- The goal is to enable single source applications and libraries to simply recompile for new architectures including Exascale Platforms.

AMD Support

- Support for everything KokkosKernels and Trilinos need
- Provided changes for Trilinos to enable Krylov solver
  - Tpetra, Belos fully compile
  - Tpetra >95% of tests pass
- Still running into known AMD bug, reported mid 2020

Support and Development

- Slack continues to be primary support vehicle
- Regular meetings with NNSA customers for progress updates held
- Continue >50 PRs merged per month



Kokkos Update and Maintenance

- Release 3.3 rolled
  - Near Feature Complete Support for HIP
- Added support for Intel OneAPI compiler
- Added support for Fujitsu ARM A64FX and Fujitsu compiler
- Improved Spack support

Deliverables Kokkos: https://github.com/kokkos/kokkos
Slack: https://kokkosteam.slack.com

EXASCALE COMPUTING PROJECT

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

FY21: FFT: Application-specific FFT optimizations and integration within Copa and ExaAM

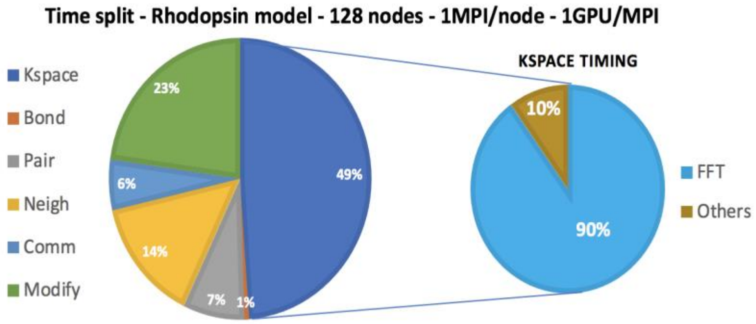
ECP WBS	2.3.3.13 FFT-ECP
PI	Jack Dongarra, (UTK – ICL)
Members	Stan Tomov (UTK – ICL), Alan Ayala (UTK – ICL), Miroslav Stoyanov (ORNL)

Scope and objectives

- Design and implement a sustainable FFT library for Exascale platforms
- Define consistent FFT-ECP APIs for FFTs on Exascale systems to help key ECP applications that need FFT functionalities to run at exascale
- FY21 plan: develop application-specific FFTs, optimizations, and integration in ECP applications; add HIP and DPC++ backends to support AMD and Intel GPUs;
- Milestone driver: Implement multidimensional FFTs and optimizations in heFFTe for applications where the input data is purely real.

*he**FFT**e 2.1 Release*

- Profile on running LAMMPS Rhodopsin benchmark with FFTMPI using 128 Summit nodes on a 1024^3 FFT;
- heFFTe with cuFFT backend accelerates FFT 2X compared to FFTMPI and 25% the entire application



Impact

- Developed application-specific FFT optimizations and integration within Copa and ExaAM applications;
- Provide ECP applications acceleration for their FFT computations on various GPU-accelerated heterogeneous architectures with GPUs from Nvidia, AMD, and Intel;
- FFT-ECP stakeholders are application developers, e.g., LAMMPS and HACC, Copa and ExaAM, as well as ECP vendors where heFFTe enables FFT applications to run more efficiently on current and upcoming platforms.

Project accomplishment

- Implemented application-specific FFT optimizations and tuning for systems with Nvidia and AMD GPUs;
- Added Intel GPU support;
- Released heFFTe 2.1 featuring with new application-specific optimizations, tuning, and added Intel GPU support;
- heFFTe Integration and acceleration within CoPA projects and ExaAM/Meumapps.

Deliverables heFFTe 2.1 http://icl.utk.edu/fft/ multidimensional FFTs and application-specific optimizations with added Intel GPUs support. Relation projects: ECP LAMMPS, HACC, CoPA, Cabana, Alpine, FFTX, SLATE, xSDK, and MAGMA (http://icl.cs.utk.edu/magma/) ExaWind, EMPIRE/PIC/PICSAR, WarpX, ExaSky, LaticeQCD/MILC, EXAALT, ExaAM, QMCPACK, NWChemEx

Document on Performance Evaluation of Solvers in *hypre* 2.20.0

ECP WBS	WBS 2.3.3.12
PI	Carol Woodward, LLNL
Members	LLNL
Milestone Lead	Ulrike Meier Yang, LLNL

Scope and objectives

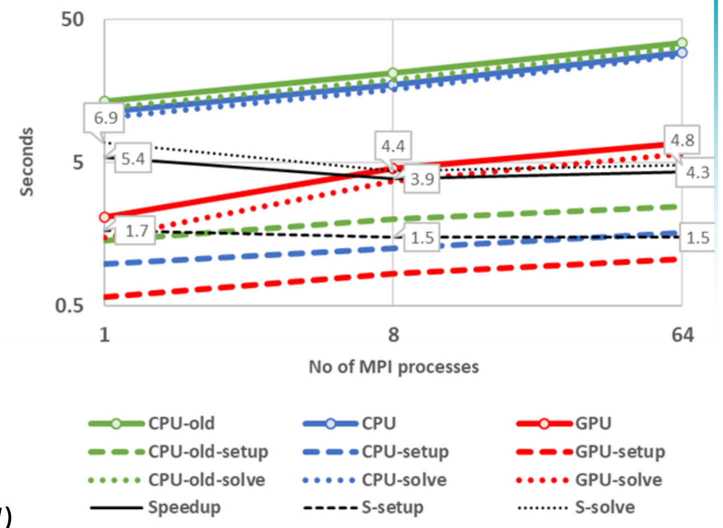
- This project focuses on enhancements for *hypre* and SUNDIALS in preparation for exascale systems
- Goals for *hypre* include increasing GPU-enabled portions as well as portability.
- This milestone evaluates and analyzes the GPU and CPU performance of structured and unstructured solvers in hypre for various problems on Lassen and Summit.

Impact

- Linear systems are an important part of many application codes, and often make up a large portion of their execution times.
- Efficient linear solvers are crucial for ECP applications, and any improvements in performance and memory usage positively impact the applications.

Comparing hypre's GPU and CPU performance

Weak scaling study of AMG-PCG applied to an unstructured problem on a crooked pipe on Lassen using 1, 2, or 16 nodes. 'CPU' uses same parameters as 'GPU' including a newly designed interpolation. 'CPU-old' uses the old Interpolation. Presented are setup, solve and total times, including Speedups (CPU/GPU)



Project accomplishment

- Measured GPU and CPU performance of hypre's structured and unstructured solvers on a variety of problems on Lassen and Summit.
- Analyzed and summarized the results in a document that is available on confluence.

Deliverables The document is available at https://confluence.exascaleproject.org/display/STLM12/Software+Documents in file 'Performance Evaluation of hypre Solvers.pdf'

Key Product Development Takeaways

ECP ST teams are creating new algorithms that effectively expose and exploit massive on-node parallelism, in addition to MPI

ECP ST products are expanding support to all GPU architectures: Nvidia, AMD, Intel

Application teams are increasingly relying on ECP ST products to get performance & portability

Getting portable performance via E4S products

Challenge: How can I port my code effectively and efficiently to diverse and emerging architectures?

- Goals:
  - Get performance
    - Get all or most of the potential performance on a platform (varies with specific situation)
    - Get on the commodity performance curve: Porting to next similar system, say 2X faster, your code is about 2X faster
  - Get portability
    - Minimize how much special code needs to be written for each target platform
    - Can be done by using portability layers, language features, libraries that provide functionality across many systems
- Examples:
  - Use Kokkos to write your parallel loops:
    - Enables performance across multiple platforms by compiling with a backend that transforms your loops for the target
    - Targets can be Intel CPU, Nvidia GPU, AMD GPU, Intel GPU, Arm SVE, future parallel devices
  - Use PETSc to solve large sparse linear systems:
    - PETSc runs well on CPUs and GPUs, adapting algorithms and implementations behind the scenes
    - Note: Assembling the sparse linear system for GPU systems needs to be done on the GPU, using, e.g., Kokkos

Writing *your* code for portable performance

OpenMP

- An open standard
- Target offload supports GPUs
- Commonly used by Fortran codes, uncommon for C++

Cuda/HIP/SYCL

- Vendor specific, esp CUDA
- HIP portable in principle, but really driven by AMD
- SYCL portable in principle, but really driven by Intel

Kokkos/RAJA

- Kokkos uses C++ template meta-programming, widely used, lots of training and documentation
- RAJA more modular design (e.g., loop vs memory management), fundamental to LLNL ecosystem

Using libraries for portable performance

Dense Lin Alg

- Vendors typically provide, e.g., MKL
- ECP efforts provide alternative for reference and design ideas

FFTs

- Vendors provide building blocks, e.g., 1D
- Many apps have their own 3D framework
- heFFTe provides new portable 3D library emphasizing internode scalability

Sparse Lin Alg

- Strong tradition for DOE
- Sparse direct: SuperLU/STRUMPACK
- Sparse iterative: PETSc, Trilinos/KokkosKernels
- Apps will need to move problem construction to GPU

Addressing IO Bottlenecks

HDF5

- Continued evolution for modern platforms

ADIOS

- Alternative, customizable library
- Also becoming available via HDF5 API

Data compression

- VeloC/SZ
- ZFP
- Libraries that support *in situ* compression

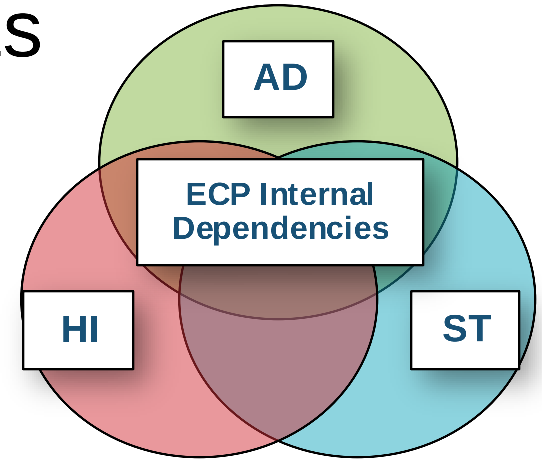
The Growing Complexity of Scientific Application Software Stacks

EXASCALE COMPUTING PROJECT

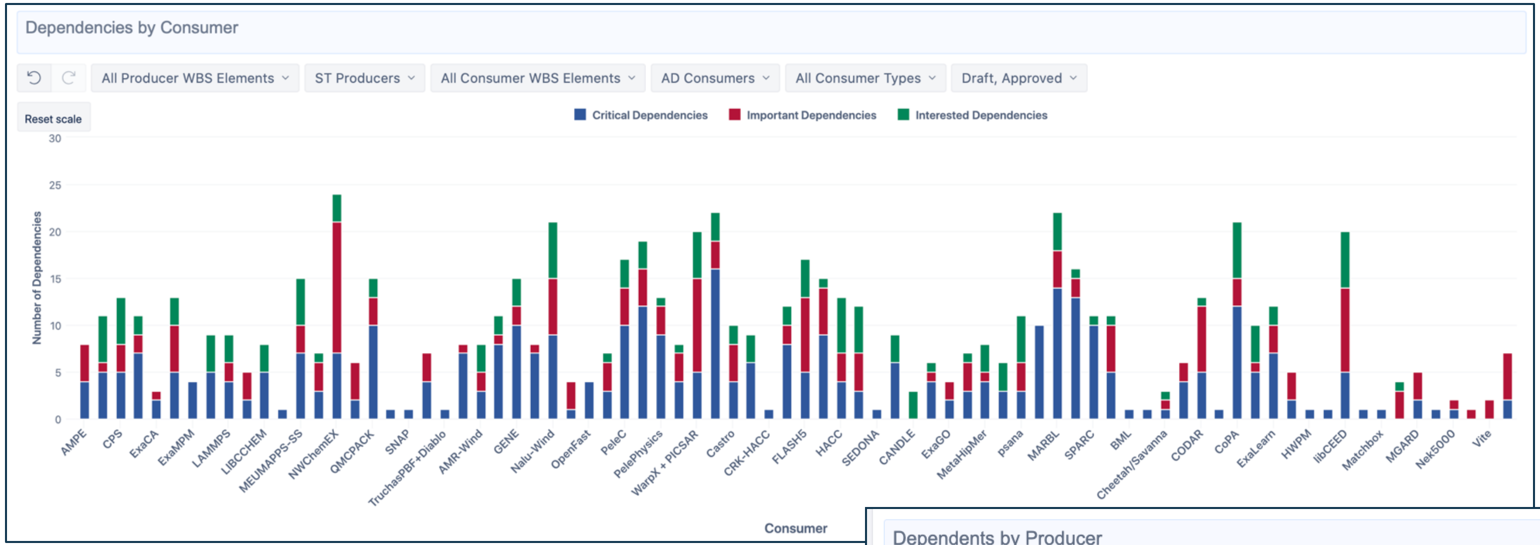
Challenges

- As our software gets more complex, it is getting harder to install tools and libraries correctly in an integrated and interoperable software stack.

ECP apps (AD) are primary consumers of ST products



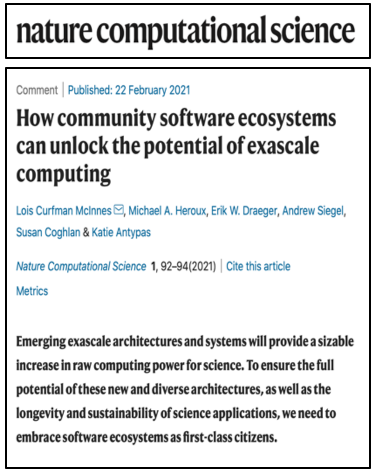
Dependency Database



View by ST producers

View by AD consumers

https://dx.doi.org/10.1038/s43588-021-00033-y



nature computational science

Comment | Published: 22 February 2021

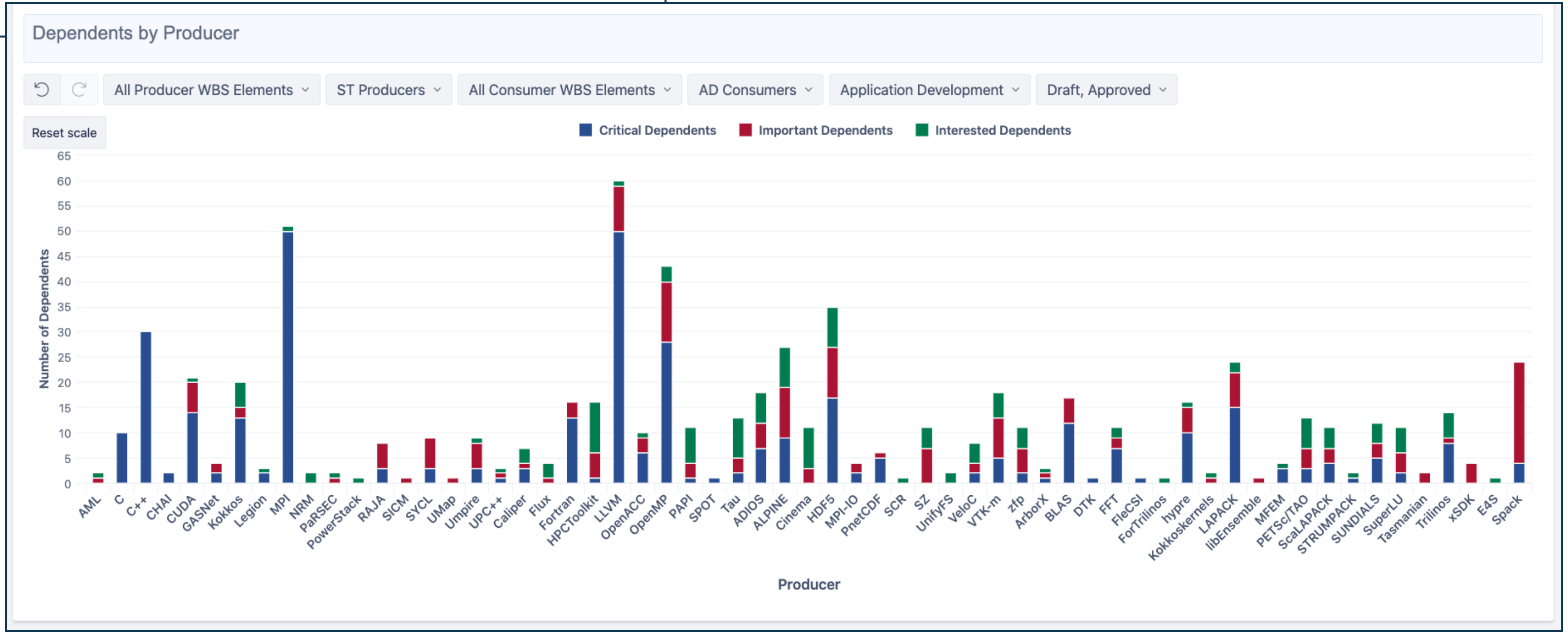
How community software ecosystems can unlock the potential of exascale computing

Lois Curfman McInnes, Michael A. Heroux, Erik W. Draeger, Andrew Siegel, Susan Coghlan & Katie Antypas

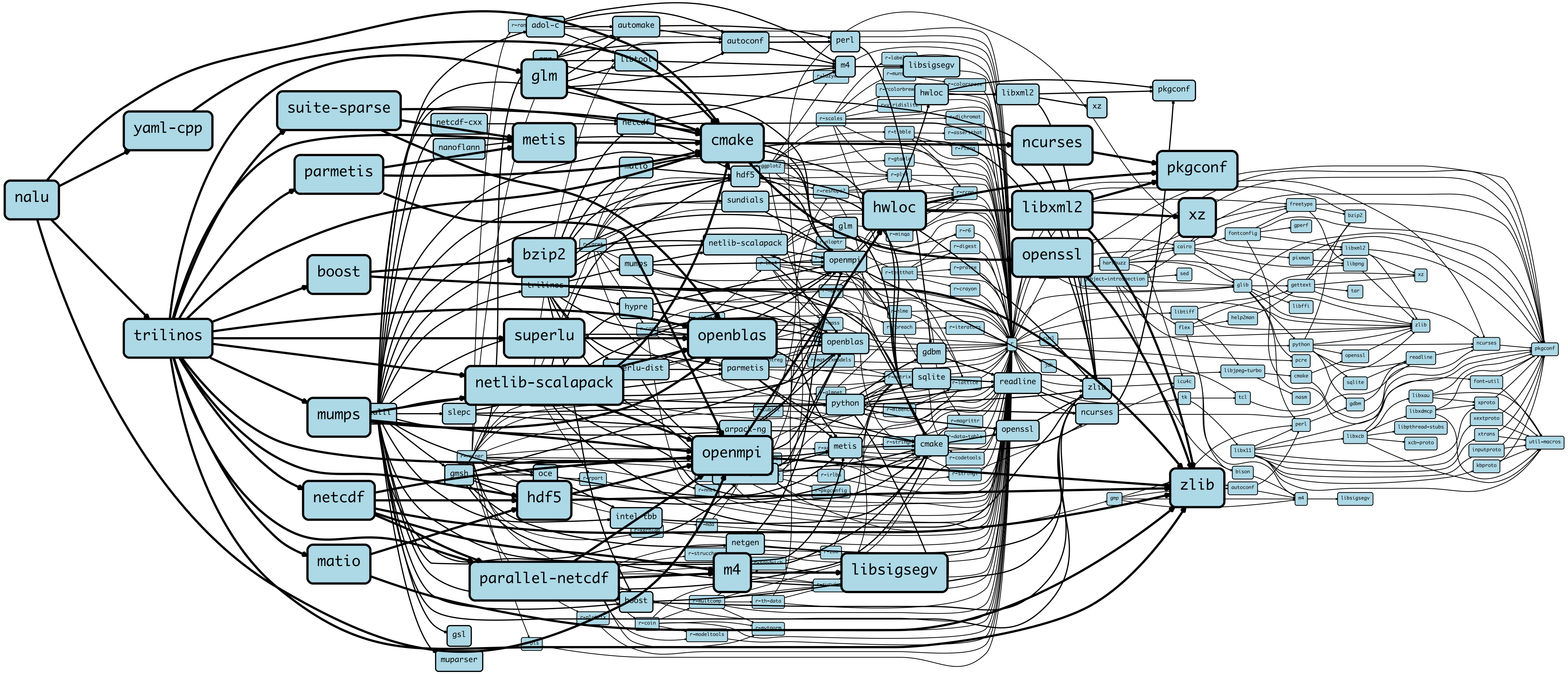
Nature Computational Science 1, 92–94(2021) | Cite this article

Metrics

Emerging exascale architectures and systems will provide a sizable increase in raw computing power for science. To ensure the full potential of these new and diverse architectures, as well as the longevity and sustainability of science applications, we need to embrace software ecosystems as first-class citizens.



Scientific software is becoming extremely complex

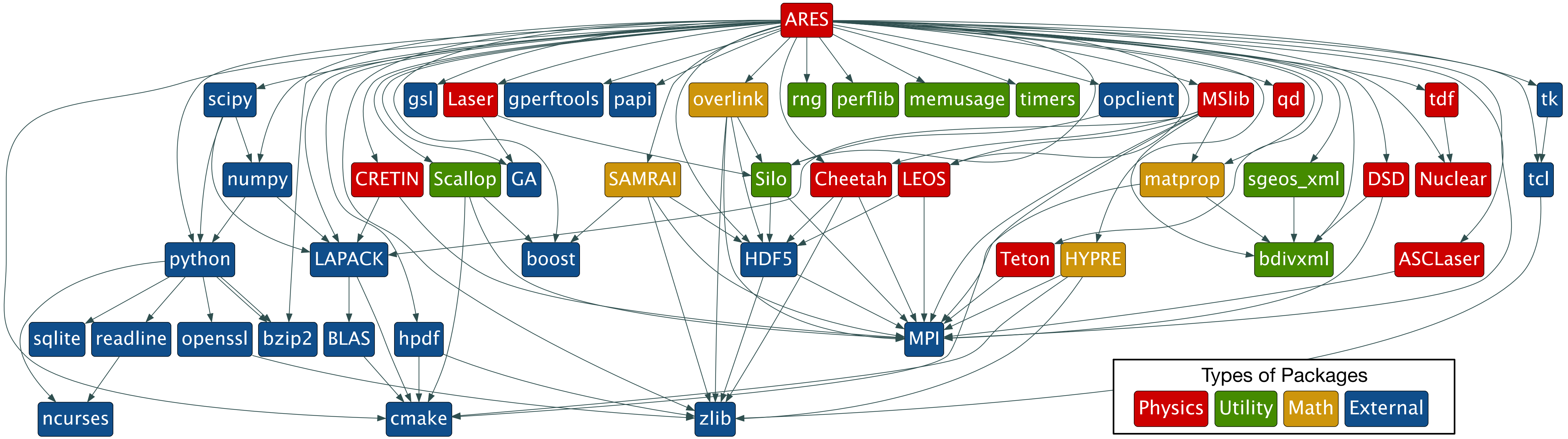


Nalu: Generalized Unstructured Massively Parallel Low Mach Flow

dealii: C++ Finite Element Library

EXASCALE COMPUTING PROJECT

Even proprietary codes are based on many open source libraries



- Half of this DAG is external (blue); *more* than half of it is open source
- Nearly *all* of it needs to be built specially for HPC to get the best performance

The Exascale Computing Project is building an entire *ecosystem*

15+ applications	x	**80+ software packages**	x	**5+ target architectures/platforms** Xeon Power KNL NVIDIA ARM Laptops?

x	**Up to 7 compilers** Intel GCC Clang XL PGI Cray NAG	x	**10+ Programming Models** OpenMPI MPICH MVAPICH OpenMP CUDA OpenACC Dharma Legion RAJA Kokkos	x	**2-3 versions of each package + external dependencies**

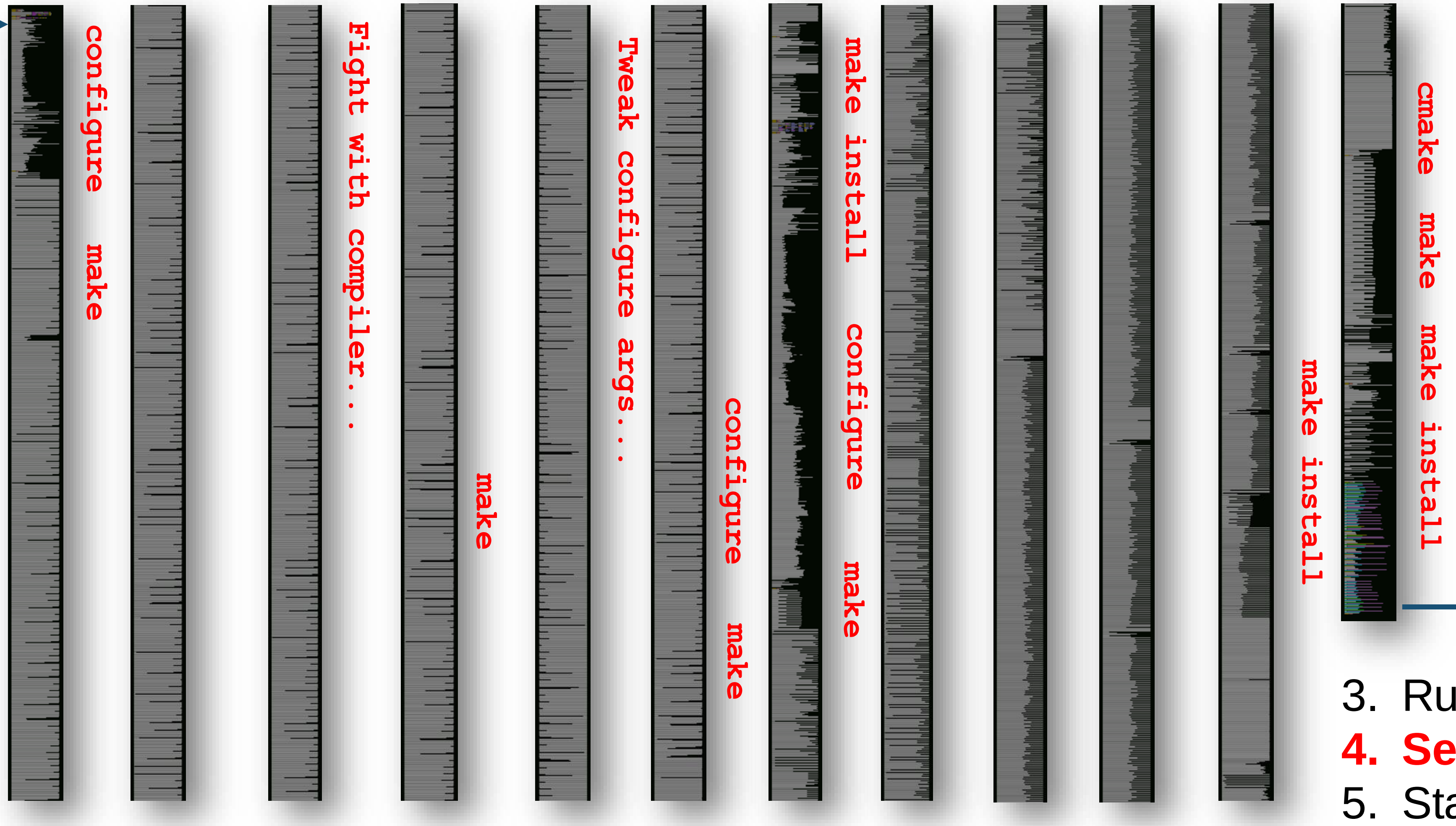
= up to **1,260,000** combinations!

- Every application has its own stack of dependencies.
- Developers, users, and facilities dedicate (many) FTEs to building & porting.
- Often trade reuse and usability for performance.

We must make it easier to rely on others' software!

How to install software on a supercomputer

1. Download all 16 tarballs you need
2. Start building!



3. Run code
4. **Segfault!?**
5. Start over…

What about modules?

- Most supercomputers deploy some form of *environment modules*
  - TCL modules (dates back to 1995) and Lmod (from TACC) are the most popular

```
$ gcc
- bash: gcc: command not found

$ module load gcc/7.0.1
$ gcc –dumpversion
7.0.1
```

- Modules don't handle installation!
  - They only modify your environment (things like PATH, LD_LIBRARY_PATH, etc.)
- Someone (likely a team of people) has already installed gcc for you!
  - Also, you can *only* `module load` the things they've installed

Spack Overview

Spack

- E4S uses the Spack package manager for software delivery
- Spack provides the ability to specify versions of software packages that are and are not interoperable.
- Spack is a build layer for not only E4S software, but also a large collection of software tools and libraries outside of ECP ST.
- Spack supports achieving and maintaining interoperability between ST software packages.
- https://spack.io

Spack is a flexible package manager for HPC

- How to install Spack (works out of the box):

```
$ git clone https://github.com/spack/spack
$ . spack/share/spack/setup-env.sh
```

- How to install a package:

```
$ spack install tau
```

- TAU and its dependencies are installed within the Spack directory.
- Unlike typical package managers, Spack can also install many variants of the same build.
  - Different compilers
  - Different MPI implementations
  - Different build options

Visit spack.io

github.com/spack/spack

@spackpm

Spack provides the *spec* syntax to describe custom configurations

```
$ git clone https://github.com/spack/spack
$ . spack/share/spack/setup-env.sh
$ spack compiler find                     # set up compilers
$ spack external find                     # set up external packages
```

```
$ spack install tau                                                          unconstrained
$ spack install tau@2.30.1                                                   @ custom version
$ spack install tau@2.30.1 %gcc@7.3.0                                        % custom compiler
$ spack install tau@2.30.1 %gcc@7.3.0 +level_zero                            +/- build option
$ spack install tau@2.30.1 %gcc@7.3.0 +mpi ^mvapich2@2.3~wrapperrpath ^ dependency information
```

- Each expression is a ***spec*** for a particular configuration
  - Each clause adds a constraint to the spec
  - Constraints are optional – specify only what you need.
  - Customize install on the command line!
- Spec syntax is recursive
  - Full control over the combinatorial build space

`spack find` shows what is installed

```
Singularity> spack find
==> 319 installed packages
-- linux-ubuntu18.04-power9le / gcc@7.3.0 -----------------------
autoconf@2.69     diffutils@3.7     libiconv@1.16       m4@1.4.18     ncurses@6.2              openssl@1.1.1g      texinfo@6.5
automake@1.16.2   findutils@4.6.0   libpciaccess@0.16   matio@1.5.17  netcdf-c@4.7.4           parmetis@4.0.3      trilinos@13.0.0
boost@1.74.0      glm@0.9.7.1       libsigsegv@2.12     metis@5.1.0   netlib-scalapack@2.1.0   perl@5.26.1         util-macros@1.19.1
bzip2@1.0.8       hdf5@1.10.7       libtool@2.4.6       mpich@3.2.1   omega-h@9.29.0           pkgconf@1.7.3       xz@5.2.5
cmake@3.18.4      hypre@2.20.0      libxml2@2.9.10      mumps@5.3.3   openblas@0.3.10          suite-sparse@5.7.2  zlib@1.2.11

-- linux-ubuntu18.04-ppc64le / gcc@7.3.0 ------------------------
adiak@0.1.1            flit@2.1.0              libpfm4@4.11.0             papyrus@develop              py-more-itertools@7.2.0      qthreads@1.14
adios@1.13.1           gasnet@2020.3.0         libpng@1.6.37              parallel-netcdf@1.12.1       py-mpi4py@3.0.3              raja@0.12.1
adios2@2.6.0           gasnet@2020.3.0         libpthread-stubs@0.4       parmetis@4.0.3               py-nbclient@0.5.0            rankstr@0.0.2
adlbx@0.9.2            gdbm@1.18.1             libquo@1.3.1               pcre@8.44                    py-nbconvert@6.0.1           readline@8.0
aml@0.1.0              gettext@0.20.2          libsigsegv@2.12            pcre2@10.35                  py-nbformat@5.0.7            redset@0.0.3
amrex@20.10            gettext@0.21            libsodium@1.0.18           pdsh@2.31                    py-nest-asyncio@1.4.0        rempi@1.1.0
arborx@0.9-beta        ginkgo@1.3.0            libtool@2.4.6              pdt@3.25.1                   py-notebook@6.1.4            scr@2.0.0
argobots@1.0           git@2.28.0              libunistring@0.9.10        perl@5.26.1                  py-numpy@1.19.2              shuffile@0.0.3
arpack-ng@3.7.0        git@2.28.0              libunwind@1.4.0            petsc@3.13.6                 py-oauthlib@3.1.0            slate@develop
ascent@develop         glm@0.9.7.1             libunwind@1.4.0            petsc@3.14.0                 py-pamela@1.0.0              slepc@3.14.0
autoconf@2.69          globalarrays@5.7        libuuid@1.0.3              pkgconf@1.7.3                py-pandocfilters@1.4.2       snappy@1.1.8
automake@1.16.2        gmake@4.2.1             libxml2@2.9.10             plasma@20.9.20               py-parso@0.6.1               sqlite@3.31.1
axl@0.3.0              gmp@6.1.2               libyogrt@1.24              precice@2.1.1                py-petsc4py@3.13.0           strumpack@5.0.0
axom@0.3.3             googletest@1.10.0       libzmq@4.3.2               pumi@2.2.2                   py-pexpect@4.7.0             suite-sparse@5.7.2
bash@5.0               gotcha@0.0.2            lmod@8.3                   py-alembic@1.0.7             py-pickleshare@0.7.5         sundials@5.4.0
binutils@2.33.1        gotcha@1.0.3            lua@5.3.5                  py-argon2-cffi@20.1.0        py-prometheus-client@0.7.1   superlu@5.2.1
bmi@develop            gperftools@2.7          lua-luafilesystem@1_7_0_2  py-asn1crypto@0.24.0         py-prompt-toolkit@2.0.9      superlu-dist@6.3.0
bolt@1.0               hdf5@1.8.21             lua-luaposix@33.4.0        py-async-generator@1.10      py-psutil@5.7.2              superlu-dist@6.3.1
boost@1.73.0           hdf5@1.8.21             lwgrp@1.0.3                py-attrs@19.3.0              py-ptyprocess@0.6.0          swig@4.0.2
boost@1.73.0           hdf5@1.10.6             lz4@1.9.2                  py-babel@2.7.0               py-py@1.8.0                  sz@1.4.12.3
boost@1.73.0           hdf5@1.10.6             lzo@2.10                   py-backcall@0.1.0            py-pycparser@2.20            sz@2.0.2.0
boost@1.73.0           hpctoolkit@2020.08.03   m4@1.4.18                  py-bleach@3.1.0              py-pyelftools@0.26           sz@2.1.10
butterflypack@1.2.0    hpx@1.5.1               magma@2.5.4                py-blinker@1.4               py-pygments@2.6.1            tar@1.32
bzip2@1.0.8            hwloc@1.11.11           margo@0.4.3                py-certifi@2020.6.20         py-pyjwt@1.7.1               tasmanian@7.3
c-blosc@1.17.0         hwloc@2.2.0             matio@1.5.17               py-certipy@0.1.3             py-pyopenssl@19.0.0          tau@2.29
caliper@2.4.0          hypre@2.18.2            mbedtls@2.16.7             py-cffi@1.14.3               py-pyrsistent@0.15.7         tcl@8.6.10
cinch@master           hypre@2.20.0            mercury@1.0.1              py-chardet@3.0.4             py-pytest-runner@5.1         texinfo@6.5
cmake@3.17.3           intel-tbb@2020.3        mercury@1.0.1              py-cryptography@2.7          py-python-dateutil@2.8.0     turbine@1.2.3
conduit@master         kokkos@3.2.00           metis@5.1.0                py-cython@0.29.21            py-python-editor@1.0.4       umap@2.1.0
conduit@master         kokkos-kernels@3.2.00   mfem@4.1.0                 py-decorator@4.4.2           py-python-oauth2@1.1.1       umpire@4.0.1
cuda@10.2.89           kvtree@1.0.2            mpark-variant@1.4.0        py-defusedxml@0.6.0          py-pytz@2020.1               umpire@4.0.1
curl@7.72.0            legion@20.03.0          mpich@3.2.1                py-entrypoints@0.3           py-pyzmq@18.1.0              unifyfs@0.9.0
darshan-runtime@3.2.1  leveldb@1.22            mpifileutils@develop       py-idna@2.8                  py-requests@2.24.0           unzip@6.0
da[illegible]3.2.1     libarchive@3.4.1        mumps@5.3.3                py-importlib-metadata@2.0.0  py-send2trash@1.5.0          upcxx@2020.3.0
di[illegible]s@3.7     libbsd@0.10.0           ncurses@6.2                py-ipykernel@5.3.4           py-setuptools@50.1.0         util-macros@1.19.1
```

- All the versions coexist!
  - Multiple versions of same package are ok.
- Packages are installed to automatically find correct dependencies.
- Binaries work *regardless of user's environment*.
- Spack also generates module files.
  - Don't *have* to use them.

The Spack community is growing rapidly

- **Spack simplifies HPC software for:**
  - Users
  - Developers
  - Cluster installations
  - The largest HPC facilities
- **Spack is central to ECP's software strategy**
  - Enable software reuse for developers and users
  - Allow the facilities to consume the entire ECP stack
- **The roadmap is packed with new features:**
  - Building the ECP software distribution
  - Better workflows for building containers
  - Stacks for facilities
  - Chains for rapid dev workflow
  - Optimized binaries
  - Better dependency resolution

Visit spack.io

github.com/spack/spack

@spackpm

ECP SW Technology Software Architecture – SDKs

ECP EXASCALE COMPUTING PROJECT

ECP applications rely on ST products across all technical areas

24 ECP applications: National security, energy, Earth systems, economic security, materials, data
6 co-design centers: machine learning, graph analytics, mesh refinement, PDE discretization, particles, online data analytics

Consider ECP software technologies needed by 5 ECP applications:

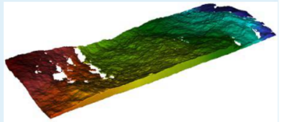
ExaWind: Turbine Wind Plant Efficiency

Harden wind plant design and layout against energy loss susceptibility; higher penetration of wind energy

Lead: NREL
DOE EERE

Subsurface: Carbon Capture, Fossil Fuel Extraction, Waste Disposal

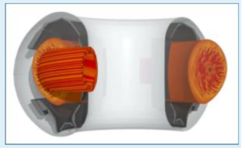
Reliably guide safe long-term consequential decisions about storage, sequestration, and exploration



Lead: LBNL
DOE BES, EERE, FE, NE

WDMApp: High-Fidelity Whole Device Modeling of Magnetically Confined Fusion Plasmas

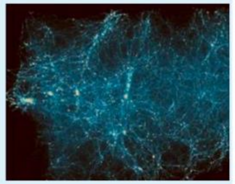
Prepare for ITER experiments and increase ROI of validation data and understanding; prepare for beyond-ITER devices



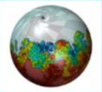
Lead: PPPL
DOE FES

ExaSky: Cosmological Probe of the Standard Model of Particle Physics

Unravel key unknowns in the dynamics of the Universe: dark energy, dark matter, and inflation



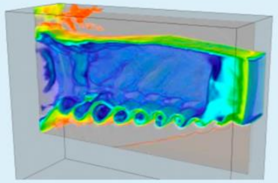
Lead: ANL
DOE HEP



The MARBL Multi-physics Code

Multi-physics simulations of high energy-density physics and focused experiments driven by high-explosive, magnetic or laser based energy sources

- Magneto-radiation-hydrodynamics at the exascale
- Next-generation pulsed power / ICF modeling
- High-order numerical methods



Lead: LLNL

ECP applications require consistency across the software stack

ECP Applications: C Cosmology (ExaSky) · F Fusion Energy (WDMApp) · N National Security (MAPP) · S Subsurface Flow · W Wind Farm (ExaWind) · … and more

24 apps, 6 co-design centers

Shown are 36 ST products (used or being considered by the 5 apps above)

ST overall has 71 unique software products used by 24 apps and 6 co-design centers

Selected ECP Software Technologies

Programming Models and Runtimes

- MPI — C F N S W
- Kokkos — F W
- RAJA — N S
- CHAI — N S
- Umpire — N S
- … and more

Tools and Technology

- TAU — F W
- HPCToolkit — C F N S
- Flux — N
- Caliper — N S
- PAPI — C F S

Compilers and Support

- LLVM — C F N S W
- OpenMP — C F N W
- … and more

Math Libraries (xSDK)

- hypre — N S W
- PETSc/TAO — F N S
- STRUMPACK — F
- SuperLU — F S W
- Trilinos — F S W
- SUNDIALS — N S
- ArborX — C W
- FFT — C W
- BLAS, LAPACK — F W
- MFEM — N
- … and more

Visualization Analysis and Reduction

- ALPINE — C N S W
- Cinema — C N
- VTK-m — C F N S W
- SZ — C
- zfp — F N
- SPOT — N
- … and more

Data Mgmt, I/O, Checkpoint Restart

- SCR — N
- MPI-IO — F
- HDF5 — C F N S W
- PnetCDF — W
- ADIOS — F W
- UnifyFS — N
- VeloC — C
- … and more

Ecosystem: E4S at large

- Spack — F N W
- … and more

Legend

- Prog Models & Runtimes
- Tools
- Math Libraries
- Data and Viz
- Ecosystems and Delivery

See E4S.io for more ST products

AID
AML
BEE
Darshan
DTK
Dyninst
FleCSI
ForTrilinios
GASNet
Ginkgo
Kokkoskernels
Legion
libEnsemble
MarFS
NRM
OpenACC
Papyrus
PaRSEC
PDT
PowerStack
ScaLAPACK
SCR
SICM
SLATE
SWIG
Tasmanian
Umap
UPC++

ECP apps rely on multiple software technologies; some software products contribute to multiple distinctly developed components of a multiphysics app (such as fusion energy modeling) that must run within a single executable.

ECP EXASCALE COMPUTING PROJECT

ECP ST SDKs will span all technology areas

Motivation: Properly chosen cross-team interactions will build relationships that support interoperability, usability, sustainability, quality, and productivity within ECP ST.

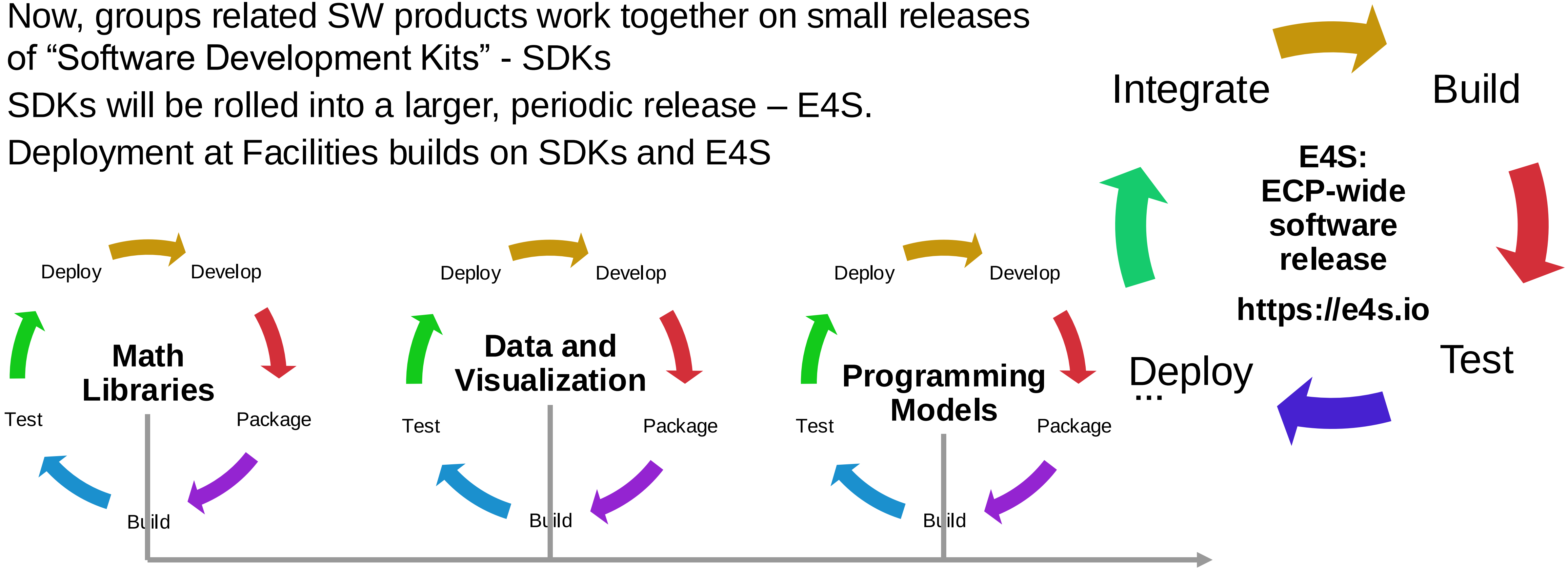
Action Plan: Identify product groupings where coordination across development teams will improve usability and practices, and foster community growth among teams that develop similar and complementary capabilities.

PMR Core (17)	Compilers and Support (7)	Tools and Technology (11)	xSDK (16)	Visualization Analysis and Reduction (9)	Data mgmt, I/O Services, Checkpoint restart (12)	Ecosystem/E4S at-large (12)
QUO	openarc	TAU	hypre	ParaView	SCR	mpiFileUtils
Papyrus	Kitsune	HPCToolkit	FleSCI	Catalyst	FAODEL	TriBITS
SICM	LLVM	Dyninst Binary Tools	MFEM	VTK-m	ROMIO	MarFS
Legion	CHiLL autotuning comp	Gotcha	Kokkoskernels	SZ	Mercury (Mochi suite)	GUFI
Kokkos (support)	LLVM openMP comp	Caliper	Trilinos	zfp	HDF5	Intel GEOPM
RAJA	OpenMP V & V	PAPI	SUNDIALS	VisIt	Parallel netCDF	BEE
CHAI	Flang/LLVM Fortran comp	Program Database Toolkit	PETSc/TAO	ASCENT	ADIOS	FSEFI
PaRSEC*		Search (random forests)	libEnsemble	Cinema	Darshan	Kitten Lightweight Kernel
DARMA		Siboka	STRUMPACK	ROVER	UnifyCR	COOLR
GASNet-EX		C2C	SuperLU		VeloC	NRM
Qthreads		Sonar	ForTrilinos		IOSS	ArgoContainers
BOLT			SLATE		HXHIM	Spack
UPC++			MAGMA			
MPICH			DTK			
Open MPI			Tasmanian			
Umpire			TuckerMPI			
AML						

Legend
- PMR
- Tools
- Math Libraries
- Data and Vis
- Ecosystems and delivery

ECP is working towards a periodic, hierarchical release process

- In ECP, teams increasingly need to ensure that their libraries and components work together
  - Historically, HPC codes used very few dependencies
- Now, groups related SW products work together on small releases of “Software Development Kits” - SDKs
- SDKs will be rolled into a larger, periodic release – E4S.
- Deployment at Facilities builds on SDKs and E4S



xSDK: Primary delivery mechanism for ECP math libraries' continual advancements toward predictive science

xSDK release 0.6.0 (Nov 2020)

- hypre
- PETSc/TAO
- SuperLU
- Trilinos
- AMReX
- ButterflyPACK
- DTK
- Ginkgo
- heFFTe
- libEnsemble
- MAGMA
- MFEM
- Omega_h
- PLASMA
- PUMI
- SLATE
- Tasmanian
- SUNDIALS
- Strumpack
- Alquimia
- PFLOTRAN
- deal.II
- preCICE
- PHIST
- SLEPc

(deal.II, preCICE, PHIST, SLEPc: from the broader community)

As motivated and validated by the needs of ECP applications:

ECP Math libraries

- Performance on new node architectures
- Extreme strong scalability
- Advanced, coupled multiphysics, multiscale
- Optimization, UQ, solvers, discretizations
- Interoperability, complementarity: xSDK

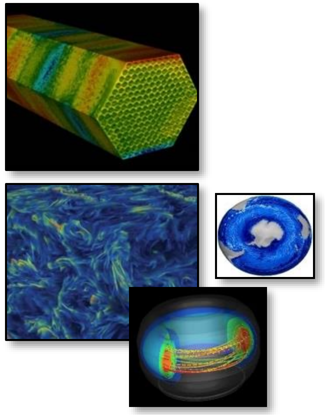
Next-generation algorithms

Advances in data structures for new node architectures

Improving library quality, sustainability, interoperability

Toward predictive scientific simulations

Increasing performance, portability, productivity



Timeline: xSDK release 1 → xSDK release 2 → ...... → xSDK release n

Ref: xSDK: Building an Ecosystem of Highly Efficient Math Libraries for Exascale, **SIAM News**, Jan 2021

The Extreme-Scale Scientific Software Stack (E4S)

EXASCALE COMPUTING PROJECT

E4S: Extreme-scale Scientific Software Stack

- Curated, Spack based software distribution
- Spack binary build caches for bare-metal installs
  - x86_64, ppc64le (IBM Power 9), and aarch64 (ARM64)
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products
- Base images and full featured containers (with GPU support)
- GitHub recipes for creating custom images from base images
- GitLab integration for building E4S images
- E4S validation test suite on GitHub
- E4S-cl container launcher tool for MPI substitution in applications using MPICH ABI
- E4S VirtualBox image with support for container runtimes
  - Docker
  - Singularity
  - Shifter
  - Charliecloud
- AWS and GCP images to deploy E4S

https://e4s.io

E4S Components

- E4S is a curated release of ECP ST products based on Spack [http://spack.io].
- E4S Spack cache to support bare-metal installs at facilities and custom container builds:
  - x86_64, ppc64le, and aarch64
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products.
- Base images and full featured containers with support for GPUs.
- GitHub recipes for creating custom images from base images.
- e4s-cl for container launch and for replacing MPI in application with system MPI libraries.
- Validation test suite on GitHub provides automated build and run tests.
- Automates build process via GitLab Continuous Integration to ensure packages can be built.
- E4S Doc Portal aggregates and summarizes documentation and metadata by raking product repos.
- E4S VirtualBox image with support for Docker, Shifter, Singularity, and Charliecloud runtimes.
- AWS image to deploy E4S on EC2.
- GCP image to deploy E4S on GCP.

https://e4s.io

Core questions E4S is addressing

How can new ECP software capabilities be effectively and efficiently integrated and sustained?	How can E4S build upon, leverage and extend existing capabilities and activities?	How can E4S become a sustainable, open, collaborative software ecosystem for HPC?
• ECP success requires development, delivery and use of new GPU capabilities in 70 products • Requires coordination of versioning, integration, testing, debugging, interaction with vendors and facilities • Requires access to new documentation • Requires focus on high quality	• Using Spack for product installation, leveraging growing Spack capabilities • Making E4S available via containers, cloud platforms • Providing integration pathways to multiple destinations: from-source, LLVM, vendor stacks, facilities, etc	• Hierarchical, open architecture to accept and manage community contributions • Defined processes for community engagement within DOE, with other US agencies, industry, international partners • Delivering the value proposition of the ecosystem vs each app managing its dependencies

Extreme-scale Scientific Software Stack (E4S)

- E4S: HPC Software Ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source**, **containers**, **cloud, binary caches**
- Leverages and enhances SDK interoperability thrust
- Not a commercial product – an open resource for all
- Oct 2018: E4S 0.1 - 24 full, 24 partial release products
- Jan 2019: E4S 0.2 - 37 full, 10 partial release products
- Nov 2019: E4S 1.0 - 50 full, 5 partial release products
- Feb 2020: E4S 1.1 - 61 full release products
- Nov 2020: E4S 1.2 (aka, 20.10) - 67 full release products
- Feb 2021: E4S 21.02 - 67 full release, 4 partial release
- May 2021: E4S 21.05 - 76 full release products

Spack

E4S

https://e4s.io

Lead: Sameer Shende (U Oregon)

Also include other products .e.g.,
AI: PyTorch, TensorFlow, Horovod
Co-Design: AMReX, Cabana, MFEM

21.05 Release: 76 Official Products + dependencies

```
1: adios2       /opt/spack/opt/spack/l:39: mpifileutils       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mpifileutils-0.11-r6bpcpm6phpuln2mibtdkryjcfwosf4m
2: aml  /opt/spack/opt/spack/linux-ubu40: ninja       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ninja-1.10.2-yve7omxowuaybwm2wsxouxnl2b5ryfb5
3: amrex        /opt/spack/opt/spack/l:41: nrm         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/nrm-0.1.0-776u3axt7iuffpe323ge7a4nvx6ljkyw
4: arborx       /opt/spack/opt/spack/l:42: omega-h     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/omega-h-9.32.5-ee2aexiee5z7fvuouny6gwpndea64xex
5: archer       /opt/spack/opt/spack/l:43: openpmd-api        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openpmd-api-0.13.4-rvqsurlwpelzfv5ge7vgu36k45af5azs
6: argobots     /opt/spack/opt/spack/l:44: openmpi     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openmpi-4.0.5-k2cwgujjufavvkche3aso2fifckc4qfk
7: ascent       /opt/spack/opt/spack/l:45: papi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papi-6.0.0.1-lbnepjda4cajdibucrhe2nwk55ukfxzk
8: axom         /opt/spack/opt/spack/l:46: papyrus     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papyrus-1.0.1-vnpujpcs733g2blalzcj6sxxggb6qmo2
9: bolt         /opt/spack/opt/spack/l:47: parallel-netcdf    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/parallel-netcdf-1.12.2-3s3fwdbb6mldmj6phplpy22myhawtbr5
10: cabana      /opt/spack/opt/spack/l:48: pdt         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pdt-3.25.1-kvi5wuu5y72fypijti3nxqvdn7zpj6ni
11: caliper     /opt/spack/opt/spack/l:49: petsc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/petsc-3.15.0-dxbnwqckqnpw3up26wbdi6w5dmousldl
12: chai        /opt/spack/opt/spack/l:50: plasma      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/plasma-20.9.20-y5bbn7xgtuvncfk5dsgbufiumanvdd2f
13: conduit     /opt/spack/opt/spack/l:51: precice     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/precice-2.2.1-jejwk7wd3ht5vogj6v43auxhbro4vrgb
14: darshan-runtime     /opt/spack/opt,52: pumi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pumi-2.2.5-o3nghxtqhqdyj6ywwq5dyckma7hp3z34
15: dyninst     /opt/spack/opt/spack/l:53: py-jupyterhub      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-jupyterhub-1.0.0-iukma66u3enhfuco42igy6xmoxueqz5w
16: faodel      /opt/spack/opt/spack/l:54: py-libensemble     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-libensemble-0.7.2-zfuxwhvakzffkxjad4xp77k7kkloll2r
17: flecsi      /opt/spack/opt/spack/l:55: qthreads    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/qthreads-1.16-p5py3iqacg5ygvfbfcxyroo7olzbkegk
18: flit        /opt/spack/opt/spack/l:56: raja        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/raja-0.13.0-2asfc6h4ae524nwl2tdsrrm2ule7e2t5
19: fortrilinos         /opt/spack/opt,57: rempi       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/rempi-1.1.0-azga72bteazyjplxkrovvpr5xmfsahxv
20: gasnet      /opt/spack/opt/spack/l:58: scr         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/scr-3.0rc1-ah3lzth5hym7yz4sebfgvzhe7e4i5xio
21: ginkgo      /opt/spack/opt/spack/l:59: slate       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slate-2021.05.02-54hoalx6fyhuxv5adq23qvqdtpvt2yxi
22: globalarrays        /opt/spack/opt,60: slepc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slepc-3.15.0-hrjvfc5jrrv5lsoukxizgl2mw7tnapzh
23: gotcha      /opt/spack/opt/spack/l:61: strumpack   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/strumpack-5.1.1-vp2z3pw7vj373o4vezjohynohrg2cc4d
24: hdf5        /opt/spack/opt/spack/l:62: sundials    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sundials-5.7.0-5fvsk4mjkzia4gk6kgw5ewxzinjgglof
25: hpctoolkit  /opt/spack/opt/spack/l:63: superlu-dist       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/superlu-dist-6.4.0-iucmfeueabiuotrxweidst6jjiedjgy7
26: hpx         /opt/spack/opt/spack/l:64: stc         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/stc-0.9.0-xumzpy3kigoayk3tjafw4fgzziwx4mjn
27: hypre       /opt/spack/opt/spack/l:65: swig        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/swig-4.0.2-fortran-nhtva752i5uwrxddcmcrtz7xypzmg6bn
28: kokkos      /opt/spack/opt/spack/l:66: sz  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sz-2.1.11.1-6odx7fth3tirf6a77eqo4z4xdnqv73g5
29: kokkos-kernels      /opt/spack/opt,67: tasmanian   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tasmanian-7.5-h6c57bz4sh32q5mzeerweuk2oiideeid
30: legion      /opt/spack/opt/spack/l:68: tau         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tau-2.30.1-2takiod6dciqrzrxauj5kmmuad7wlpto
31: libnrm      /opt/spack/opt/spack/l:69: trilinos    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/trilinos-13.0.1-j3xepw77pbfziv5zi6llqathgd2rsvpd
32: libquo      /opt/spack/opt/spack/l:70: turbine     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/turbine-1.3.0-3hmjjd3d7tmer6hfncyndsfpoti6zsx5
33: llvm-doe    /opt/spack/opt/spack/l:71: umap        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umap-2.1.0-r2kjvkblf2muxddcekd3mnrioeibekyz
34: magma       /opt/spack/opt/spack/l:72: umpire      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umpire-4.1.2-mehd645kgrzsihe642q7hd64fcsomkod
35: mercury     /opt/spack/opt/spack/l:73: unifyfs     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/unifyfs-0.9.1-kd5rl2cp3clvaz2kuzh2mtasqduacaxq
36: metall      /opt/spack/opt/spack/l:74: upcxx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/upcxx-2021.3.0-hstxnkb7hhre5gy6h4ib7zgp2ttdbdi7
37: mfem        /opt/spack/opt/spack/l:75: warpx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/warpx-21.05-qnzleiwscixmmxoeg6lrkwd2omm2o4hk
38: mpich       /opt/spack/opt/spack/l:76: zfp         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/zfp-0.5.5-iprubqak5qwzcfd6jk3vkxgtouqvrznc
```

EXASCALE COMPUTING PROJECT

Delivering an open, hierarchical software ecosystem

More than a collection of individual products

Levels of Integration	Product	Source and Delivery
• Build all SDKs • Build complete stack • Assure core policies • Build, integrate, test	E4S	**Source:** ECP E4S team; Non-ECP Products (all dependencies) **Delivery:** spack install e4s; containers; CI Testing
• Group similar products • Make interoperable • Assure policy compliant • Include external products	SDKs	**Source:** SDK teams; Non-ECP teams (policy compliant, spackified) **Delivery:** Apps directly; spack install sdk; future: vendor/facility
• Standard workflow • Existed before ECP	ST Products	**Source:** ECP L4 teams; Non-ECP Developers; Standards Groups **Delivery:** Apps directly; spack; vendor stack; facility stack

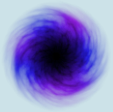
ECP ST Open Product Integration Architecture

ECP ST Individual Products

E4S: Better quality, documentation, testing, integration, delivery, building & use

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

Quality Commitment
Community policies, improvement



DocPortal
Single portal to all E4S product info

Portfolio testing
Especially leadership platforms

Curated collection
The end of dependency hell

Quarterly releases
Release 1.2 – November

Build caches
10X build time improvement

Turnkey stack
A new user experience

https://e4s.io

E4S Strategy Group
US agencies, industry, international

EXASCALE COMPUTING PROJECT

E4S Commitment to Quality
- Community Policies
- Practice Improvement

EXASCALE COMPUTING PROJECT

E4S Community Policies V1.0 Released

What is E4S?

The Extreme-scale Scientific Software Stack (E4S) is a community effort to provide open source software packages for developing, deploying and running scientific applications on high-performance computing (HPC) platforms. E4S provides from-source builds and containers of a broad collection of HPC software packages.

Purpose

E4S exists to accelerate the development, deployment and use of HPC software, lowering the barriers for HPC users. E4S provides containers and turn-key, from-source builds of more than 80 popular HPC products in programming models, such as MPI; development tools such as HPCToolkit, TAU and PAPI; math libraries such as PETSc and Trilinos; and Data and Viz tools such as HDF5 and Paraview.

Approach

By using Spack as the meta-build tool and providing containers of pre-built binaries for Docker, Singularity, Shifter and CharlieCloud, E4S enables the flexible use and testing of a large collection of reusable HPC software packages.

E4S Community Policies Version 1

A Commitment to Quality Improvement

- Will serve as membership criteria for E4S
  - Membership is not required for *inclusion* in E4S
  - Also includes forward-looking draft policies
- Purpose: enhance sustainability and interoperability
- Topics cover building, testing, documentation, accessibility, error handling and more
- Multi-year effort led by SDK team
  - Included representation from across ST
  - Multiple rounds of feedback incorporated from ST leadership and membership
- Modeled after xSDK Community Policies
- https://e4s-project.github.io/policies.html

P1 *Spack-based Build and Installation* Each E4S member package supports a scriptable *Spack* build and production-quality installation in a way that is compatible with other E4S member packages in the same environment. When E4S build, test, or installation issues arise, there is an expectation that teams will collaboratively resolve those issues.

P2 *Minimal Validation Testing* Each E4S member package has at least one test that is executable through the E4S validation test suite (*https://github.com/E4S-Project/testsuite*). This will be a post-installation test that validates the usability of the package. The E4S validation test suite provides basic confidence that a user can compile, install and run every E4S member package. The E4S team can actively participate in the addition of new packages to the suite upon request.

P3 *Sustainability* All E4S compatibility changes will be sustainable in that the changes go into the regular development and release versions of the package and should not be in a private release/branch that is provided only for E4S releases.

P4 *Documentation* Each E4S member package should have sufficient documentation to support installation and use.

P5 *Product Metadata* Each E4S member package team will provide key product information via metadata that is organized in the *E4S DocPortal* format. Depending on the filenames where the metadata is located, this may require *minimal setup*.

P6 *Public Repository* Each E4S member package will have a public repository, for example at GitHub or Bitbucket, where the development version of the package is available and pull requests can be submitted.

P7 *Imported Software* If an E4S member package imports software that is externally developed and maintained, then it must allow installing, building, and linking against a functionally equivalent outside copy of that software. Acceptable ways to accomplish this include (1) forsaking the internal copied version and using an externally-provided implementation or (2) changing the file names and namespaces of all global symbols to allow the internal copy and the external copy to coexist in the same downstream libraries and programs. This pertains primarily to third party support libraries and does not apply to key components of the package that may be independent packages but are also integral components to the package itself.

P8 *Error Handling* Each E4S member package will adopt and document a consistent system for signifying error conditions as appropriate for the language and application. For e.g., returning an error condition or throwing an exception. In the case of a command line tool, it should return a sensible exit status on success/failure, so the package can be safely run from within a script.

P9 *Test Suite* Each E4S member package will provide a test suite that does not require special system privileges or the purchase of commercial software. This test suite should grow in its comprehensiveness over time. That is, new and modified features should be included in the suite.

EXASCALE COMPUTING PROJECT

https://ideas-productivity.org

IDEAS-ECP team works with the ECP community to improve developer productivity and software sustainability as key aspects of increasing overall scientific productivity.

1 Customize and curate methodologies

- Target scientific software productivity and sustainability
- Use workflow for best practices content development

2 Incrementally and iteratively improve software practices

- Determine high-priority topics for improvement and track progress
- *Productivity and Sustainability Improvement Planning (PSIP)*

3 Establish software communities

- Determine community policies to improve software quality and compatibility
- Create Software Development Kits (SDKs) to facilitate the combined use of complementary libraries and tools

4 Engage in community outreach

- Broad community partnerships
- Collaboration with computing facilities
- Webinars, tutorials, events
- *WhatIs* and *HowTo* docs
- Better Scientific Software site (https://bssw.io)

Productivity and Sustainability Improvement Planning (PSIP)

psip

A lightweight iterative workflow, where teams identify their most urgent software bottlenecks and track progress to overcome them.

Snapshot of PSIP Progress Tracking Card (PTC)

Developers of HDF5 used PSIP to:

- Modernize processes for handling documentation (PTC)
- Move HDF5 from a THG managed Bitbucket instance to GitHub (PTC)
- Define and adopt a set of consistent coding standards (PTC)

Target

Create end-user documentation for the HDF5 library

User Story

As an HDF5 stakeholder, I want an easy-to-use reference manual, so that I can have access to accurate information about the HDF5 library.

Card

Items in blue are in works now.

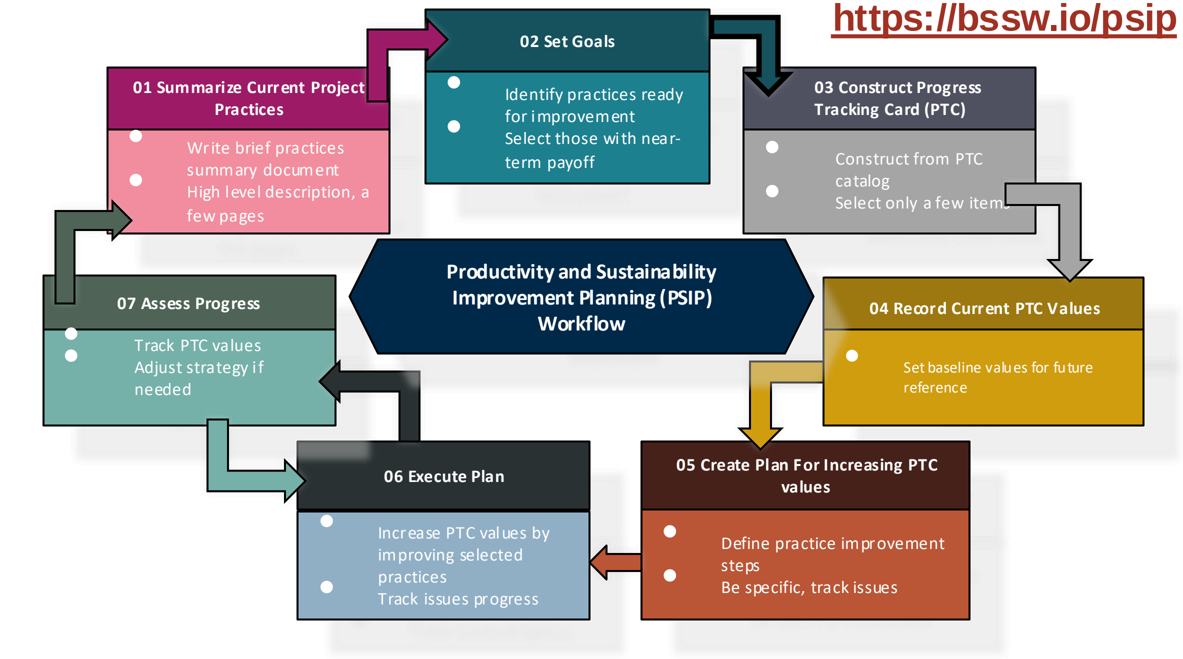
Score	Description
0	~~Multiple RM versions based on different technologies exist. Stakeholders and requirements are not documented and partially unknown.~~
1	~~Stakeholder requirements are identified~~ and documented. The requirements list is prioritized.
2	~~Potential technical solutions have been researched~~ and the pros & cons of each have been evaluated.
3	~~A technical solution has been selected and an outline of the end product vetted by stakeholders.~~
4	Existing artifacts have been converted to the new system.
5	The process for updating the API and the accompanying RM entries is documented and followed.
6	An RM maintenance plan is created.
7	The process for reviewing and improving RM entries is documented and incorporated into the product life cycle.

Comments

- The conversion of existing artifacts will take place incrementally, e.g., 30% / 60% / 100%.
- Opportunities for conversion automation will be explored.
- Opportunities for automation of RM updates in response to API changes will be explored.

EXASCALE COMPUTING PROJECT

https://bssw.io/psip



"The PSIP project had an immediate impact on our community. With the GitHub move we see increasing amount of small but very valuable contributions to make HDF5 code and documentation better." Elena Pourmal, Director of Engineering, The HDF Group

better scientific software

https://bssw.io/blog_posts/recent-successes-with-psip-on-hdf5

Better Scientific Software (BSSw) Fellowship Program

Meet Our Fellows

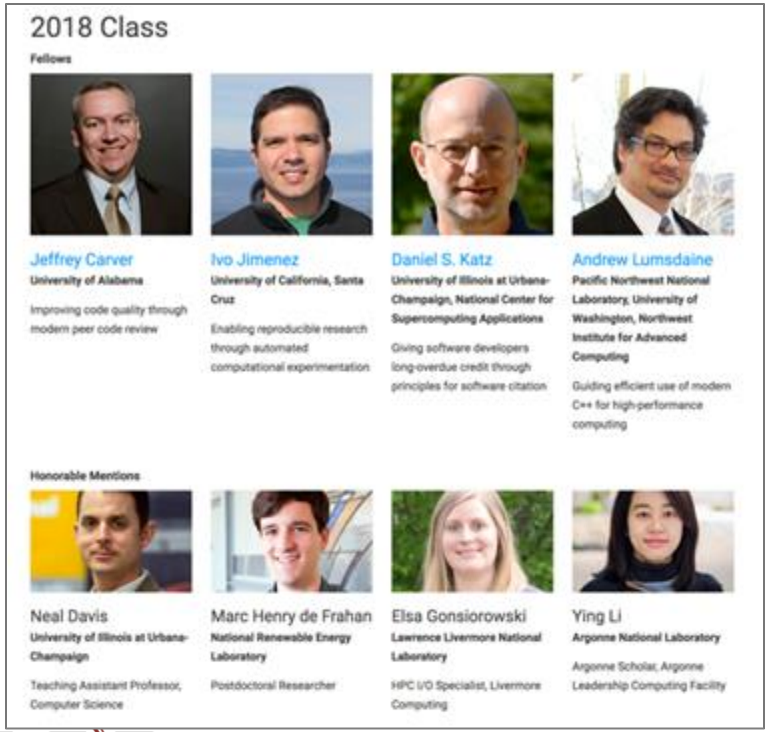
The BSSw Fellowship program gives recognition and funding to leaders and advocates of high-quality scientific software. Meet the Fellows and Honorable Mentions and learn more about how they impact Better Scientific Software.

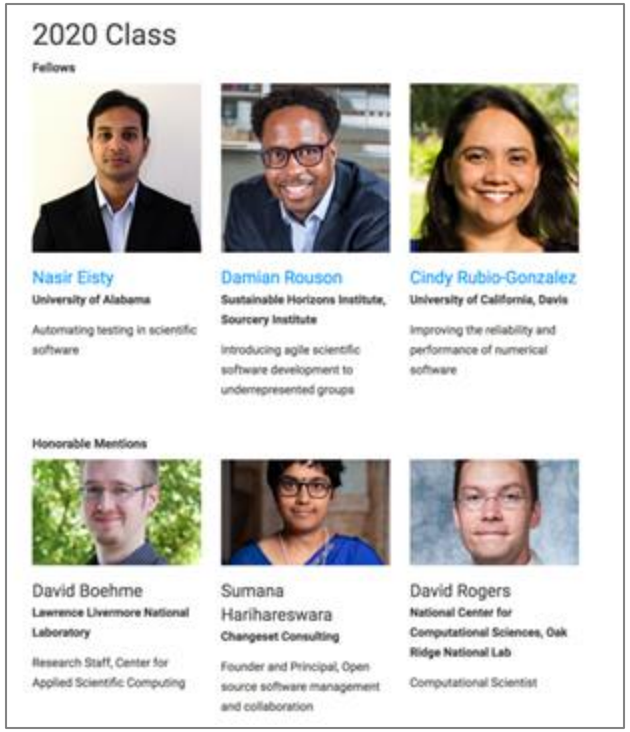
Fellowships Overview | Apply | Meet Our Fellows | BSSw Fellowship FAQ

Goal: Foster and promote practices, processes, and tools to improve developer productivity and software sustainability of scientific codes.

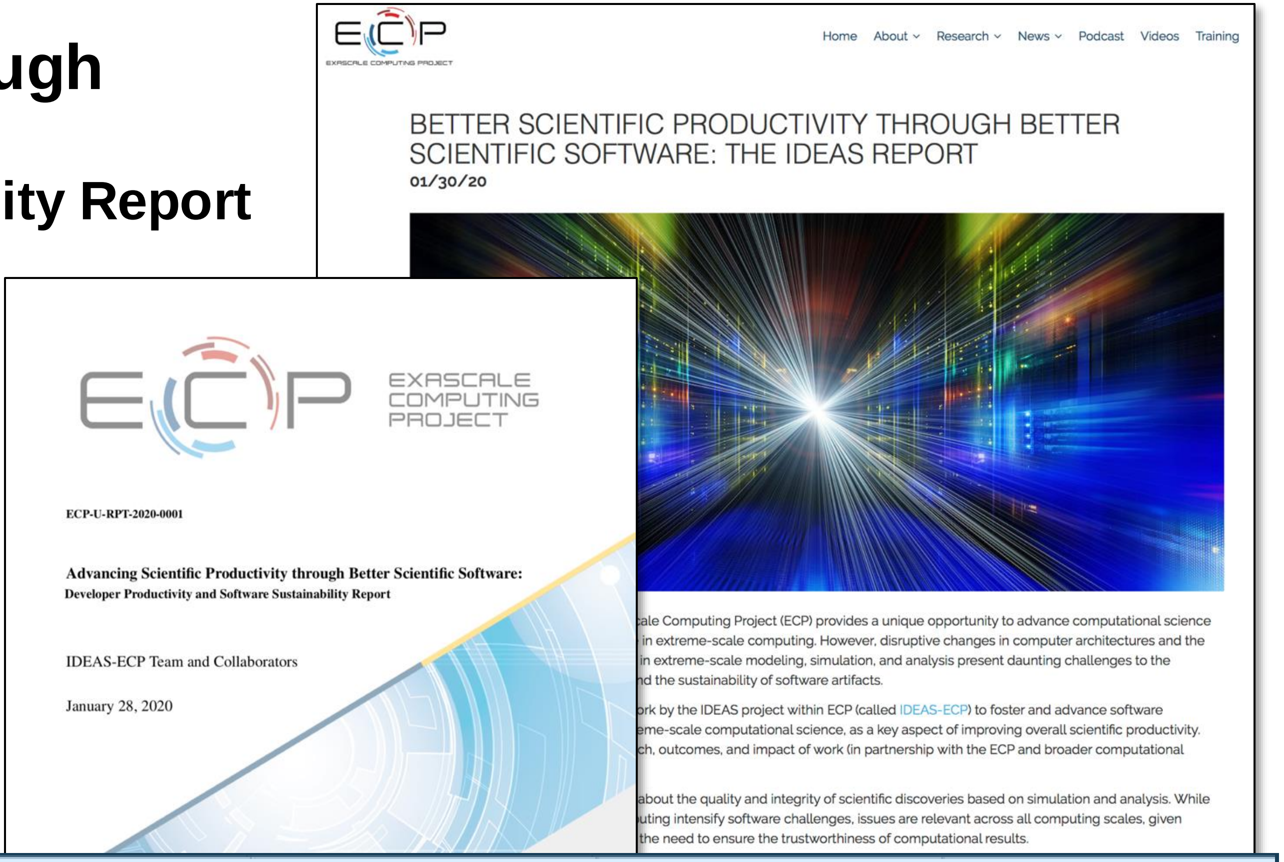
2018 - 2020

Coming soon: announcement of 2021 Fellows and HMs





Advancing Scientific Productivity through Better Scientific Software:
Developer Productivity & Software Sustainability Report



Disruptive changes in computer architectures and the complexities of tackling new frontiers in extreme-scale modeling, simulation, and analysis present daunting challenges to software productivity and sustainability.

This report explains the IDEAS approach, outcomes, and impact of work (in partnership with the ECP and broader computational science community).

Target readers are all those who care about the quality and integrity of scientific discoveries based on simulation and analysis. While the difficulties of extreme-scale computing intensify software challenges, issues are relevant across all computing scales, given universal increases in complexity and the need to ensure the trustworthiness of computational results.

Preparing public report update to be released in Jan 2021: adding topics that are completely new this year (Performance Portability Panel Series, Strategies for Working Remotely Panel Series), topics not previously discussed in depth (Collegeville Workshop Series on Scientific Software, tools for analysis), as well as an update on Productivity and Sustainability Improvement Planning. Info provided in internal ECP report, Sept 2020.

https://exascaleproject.org/better-scientific-productivity-through-better-scientific-software-the-ideas-report

E4S DocPortal

E4S DocPortal

- Single point of access
- All E4S products
- Summary Info
  - Name
  - Functional Area
  - Description
  - License
- Searchable
- Sortable
- Rendered daily from repos

e4s-project.github.io

E4S Products

*: Member Product

Show 10 entries

Search:

Name	Area	Description	Latest Doc Update
ADIOS2	Data & Viz	I/O and data management library for storage I/O, in-memory code coupling and online data analysis and visualization workflows.	2021-03-10 16:45:25
AML	PMR	Hierarchical memory management library from Argo.	2019-04-25 13:03:01
AMREX	PMR	A framework designed for building massively parallel block- structured adaptive mesh refinement applications.	2021-05-02 17:26:43
ARBORX	Math libraries	Performance-portable geometric search library	2021-01-05 15:39:55
ARCHER	Tools	Data race detection tool for OpenMP applications	2020-08-19 11:04:14
ASCENT	Data & Viz	Flyweight in situ visualization and analysis runtime for multi-physics HPC simulations	2021-04-05 18:11:45
BEE	Software Ecosystem	Container-based solution for portable build and execution across HPC systems and cloud resources	2018-08-22 22:26:19
BOLT	Development Tools	OpenMP over lightweight threads.	2020-05-04 11:24:57
CALIPER	Development tools	Performance analysis library.	2020-11-04 23:53:07
CHAI	PMR	A library that handles automatic data migration to different memory spaces behind an array-style interface.	2020-11-02 19:58:24
Name			Latest Doc Update

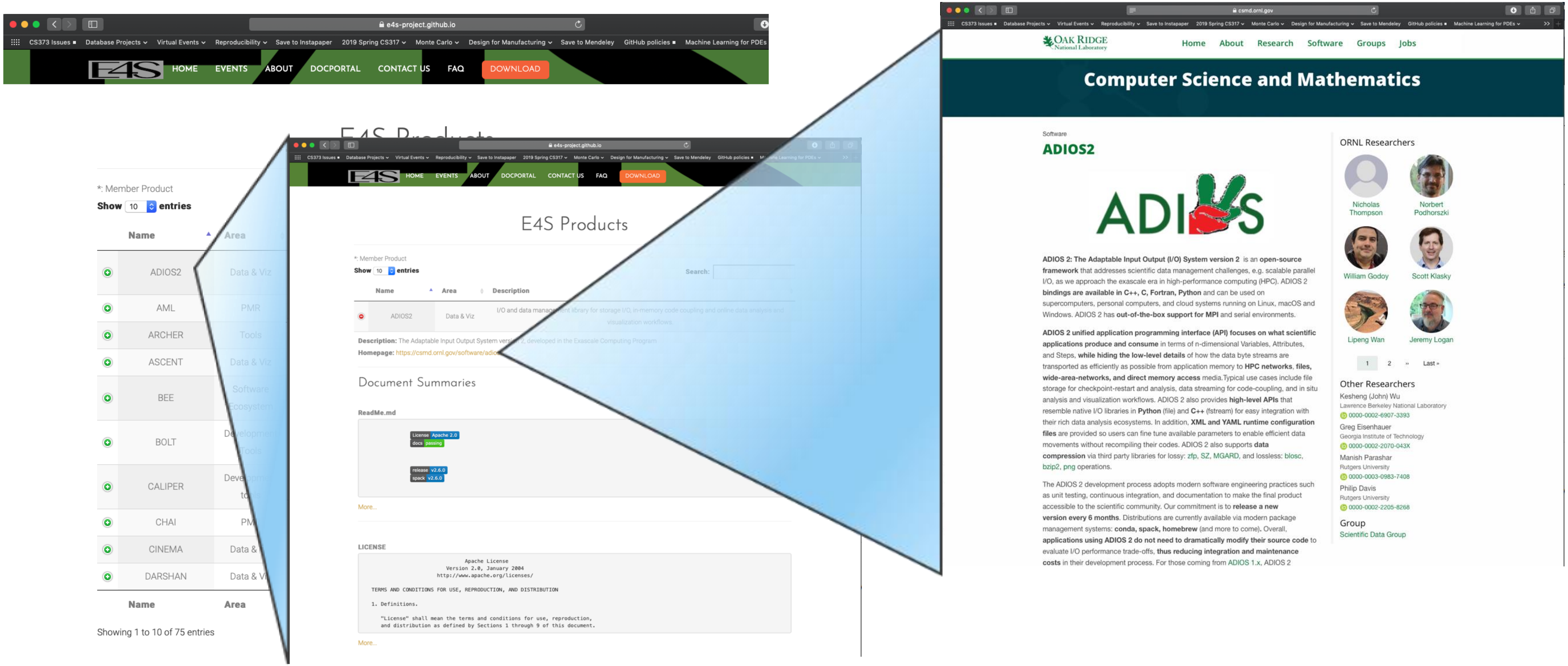
https://e4s-project.github.io/DocPortal.html

Showing 1 to 10 of 76 entries

Previous 1 2 3 4 5 … 8 Next

EXASCALE COMPUTING PROJECT

Goal: All E4S product documentation accessible from single portal on E4S.io (working mock webpage below)



https://e4s-project.github.io/DocPortal.html

Using E4S: From source using Spack and build caches

E4S Spack environment spack.yaml

https://github.com/UO-OACISS/e4s/blob/master/docker-recipes/ubuntu18.04-e4s-x86_64/spack

Search or jump to... Pull requests Issues Marketplace Explore

UO-OACISS / e4s

Unwatch 5 Star 11 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master e4s / docker-recipes / ubuntu18.04-e4s-x86_64 / spack.yaml Go to file

eugeneswalker remove old recipes Latest commit 096d2ec on Mar 23, 2020 History

1 contributor

201 lines (201 sloc) 3.04 KB Raw Blame

```
spack:
  concretization: separately
  definitions:
  - e4s:
    - openpmd-api
    - slate
    - magma
    - stc
    - py-libensemble^python@3.7.3
    - flecsi+cinch
    - hypre
    - mfem
    - trilinos@12.14.1+dtk+intrepid2+shards
    - sundials
    - petsc
    - strumpack
    - superlu-dist
    - superlu
    - tasmanian
    - openblas threads=openmp
    - faodel
    - mercury
    - hdf5
    - adios2
    - adios
    - darshan-runtime
    - darshan-util
    - veloc
    - scr
```

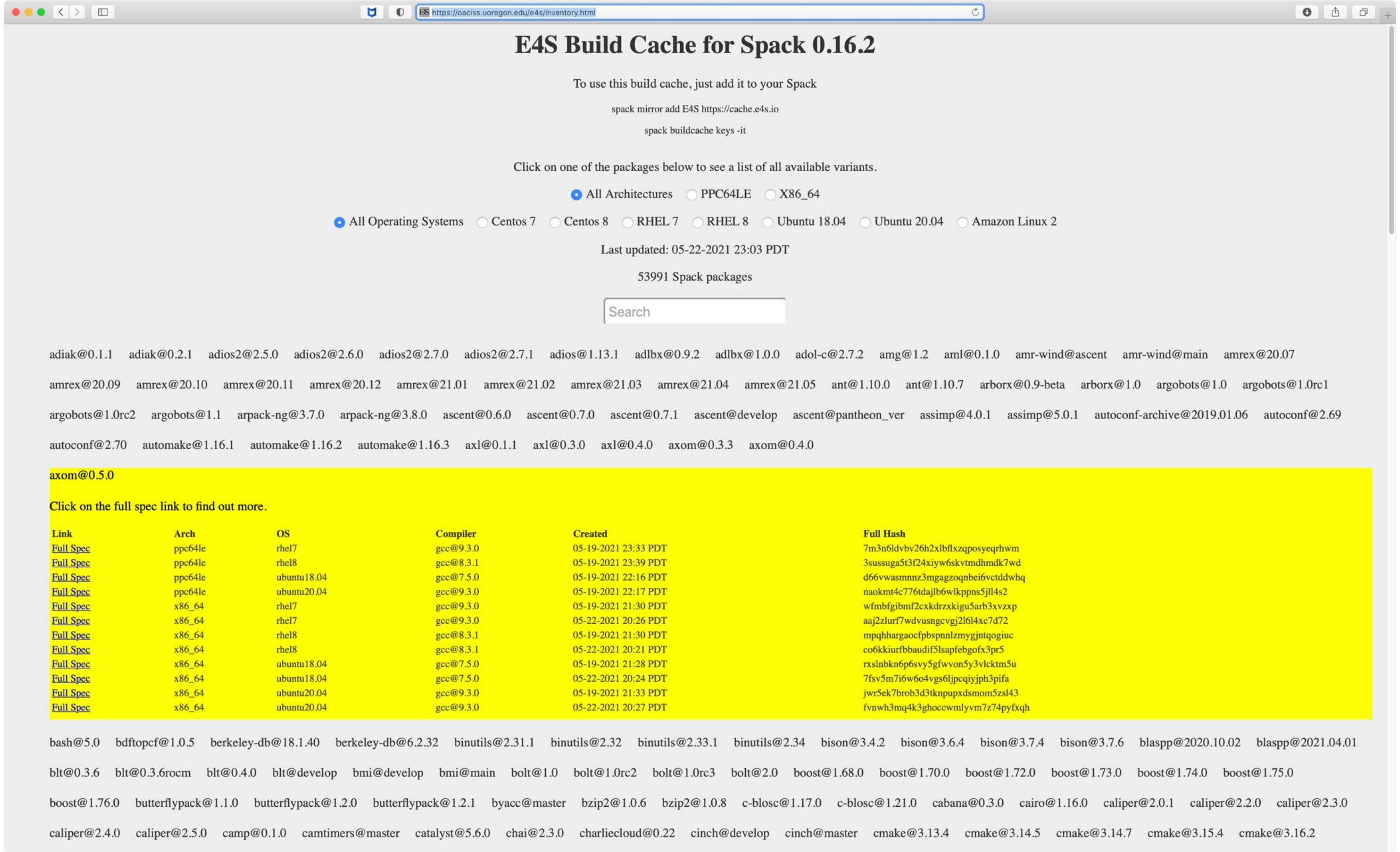
- Bare-metal install
  % cat spack.yaml
  % spack -e . install

- Docker build:

Executable File | 2 lines (2 sloc) | 78 Bytes

```
#!/bin/bash -x
docker build --no-cache -t ecpe4s/ubuntu18.04-e4s-x86_64:1.2 .
```

E4S: Spack Build Cache at U. Oregon



https://oaciss.uoregon.edu/e4s/inventory.html

E4S Build Cache for Spack 0.16.2

To use this build cache, just add it to your Spack

spack mirror add E4S https://cache.e4s.io

spack buildcache keys -it

Click on one of the packages below to see a list of all available variants.

All Architectures | PPC64LE | X86_64

All Operating Systems | Centos 7 | Centos 8 | RHEL 7 | RHEL 8 | Ubuntu 18.04 | Ubuntu 20.04 | Amazon Linux 2

Last updated: 05-22-2021 23:03 PDT

53991 Spack packages

Search

adiak@0.1.1 adiak@0.2.1 adios2@2.5.0 adios2@2.6.0 adios2@2.7.0 adios2@2.7.1 adios@1.13.1 adlbx@0.9.2 adlbx@1.0.0 adol-c@2.7.2 amg@1.2 aml@0.1.0 amr-wind@ascent amr-wind@main amrex@20.07

amrex@20.09 amrex@20.10 amrex@20.11 amrex@20.12 amrex@21.01 amrex@21.02 amrex@21.03 amrex@21.04 amrex@21.05 ant@1.10.0 ant@1.10.7 arborx@0.9-beta arborx@1.0 argobots@1.0 argobots@1.0rc1

argobots@1.0rc2 argobots@1.1 arpack-ng@3.7.0 arpack-ng@3.8.0 ascent@0.6.0 ascent@0.7.0 ascent@0.7.1 ascent@develop ascent@pantheon_ver assimp@4.0.1 assimp@5.0.1 autoconf-archive@2019.01.06 autoconf@2.69

autoconf@2.70 automake@1.16.1 automake@1.16.2 automake@1.16.3 axl@0.1.1 axl@0.3.0 axl@0.4.0 axom@0.3.3 axom@0.4.0

axom@0.5.0

Click on the full spec link to find out more.

Link	Arch	OS	Compiler	Created	Full Hash
Full Spec	ppc64le	rhel7	gcc@9.3.0	05-19-2021 23:33 PDT	7m3n6ldvbv26h2xlbflxzqposyeqrhwm
Full Spec	ppc64le	rhel8	gcc@8.3.1	05-19-2021 23:39 PDT	3sussuga5t3f24xiyw6skvtmdhmdk7wd
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	05-19-2021 22:16 PDT	d66vwasmnnz3mgagzoqnbei6vctddwhq
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	05-19-2021 22:17 PDT	naokmt4c776tdajlb6wlkppns5jll4s2
Full Spec	x86_64	rhel7	gcc@9.3.0	05-19-2021 21:30 PDT	wfmbfgibmf2cxkdrzxkigu5arb3xvzxp
Full Spec	x86_64	rhel7	gcc@9.3.0	05-22-2021 20:26 PDT	aaj2zlurf7wdvusngcvgj2l6l4xc7d72
Full Spec	x86_64	rhel8	gcc@8.3.1	05-19-2021 21:30 PDT	mpqhhargaocfpbspnnlzmygjntqogiuc
Full Spec	x86_64	rhel8	gcc@8.3.1	05-22-2021 20:21 PDT	co6kkiurfbbaudif5lsapfebgofx3pr5
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	05-19-2021 21:28 PDT	rxslnbkn6p6svy5gfwvon5y3vlcktm5u
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	05-22-2021 20:24 PDT	7fsv5m7i6w6o4vgs6ljpcqiyjph3pifa
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	05-19-2021 21:33 PDT	jwr5ek7brob3d3tknpupxdsmom5zsl43
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	05-22-2021 20:27 PDT	fvnwh3mq4k3ghoccwmlyvm7z74pyfxqh

bash@5.0 bdftopcf@1.0.5 berkeley-db@18.1.40 berkeley-db@6.2.32 binutils@2.31.1 binutils@2.32 binutils@2.33.1 binutils@2.34 bison@3.4.2 bison@3.6.4 bison@3.7.4 bison@3.7.6 blaspp@2020.10.02 blaspp@2021.04.01

blt@0.3.6 blt@0.3.6rocm blt@0.4.0 blt@develop bmi@develop bmi@main bolt@1.0 bolt@1.0rc2 bolt@1.0rc3 bolt@2.0 boost@1.68.0 boost@1.70.0 boost@1.72.0 boost@1.73.0 boost@1.74.0 boost@1.75.0

boost@1.76.0 butterflypack@1.1.0 butterflypack@1.2.0 butterflypack@1.2.1 byacc@master bzip2@1.0.6 bzip2@1.0.8 c-blosc@1.17.0 c-blosc@1.21.0 cabana@0.3.0 cairo@1.16.0 caliper@2.0.1 caliper@2.2.0 caliper@2.3.0

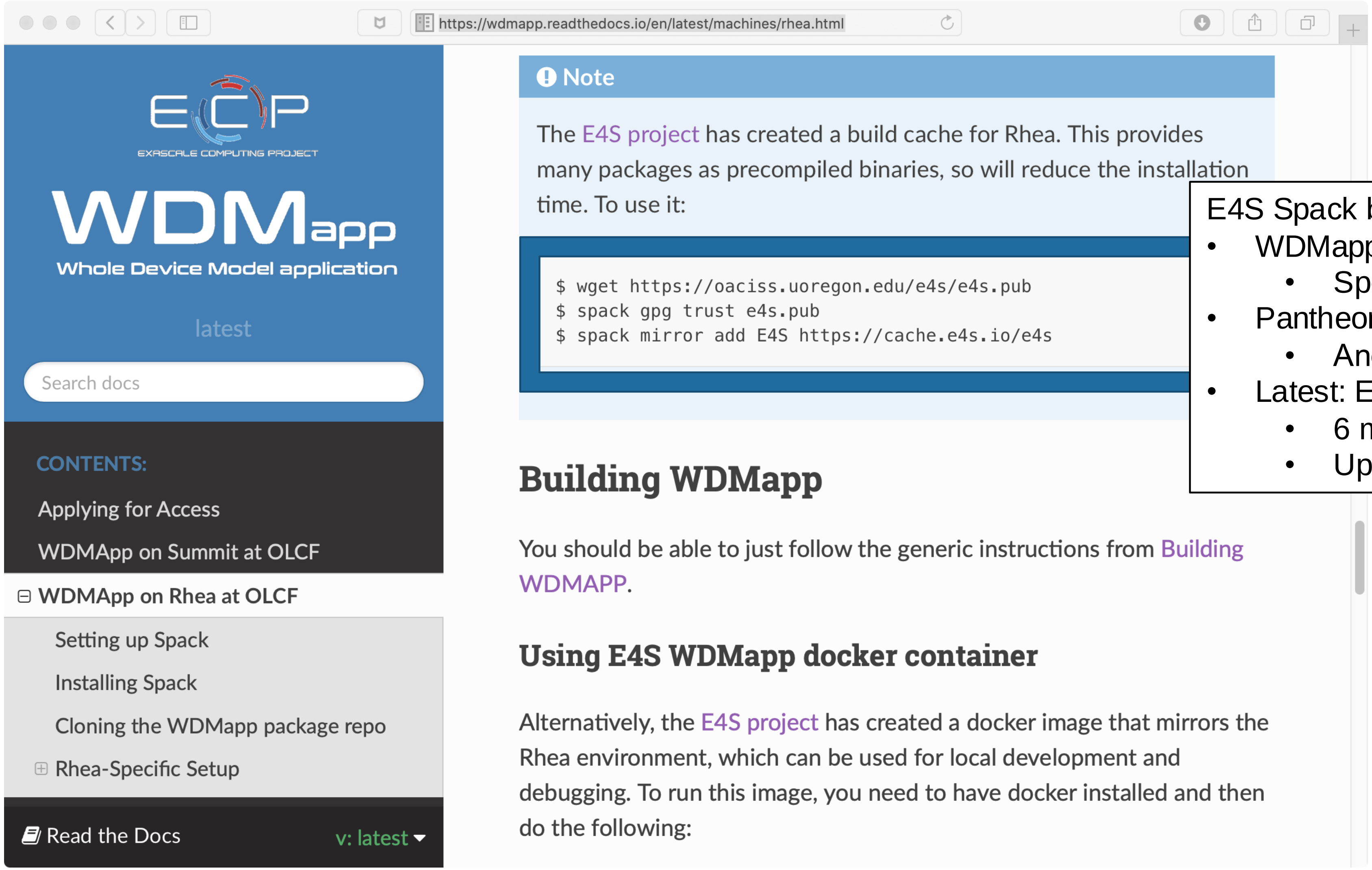
caliper@2.4.0 caliper@2.5.0 camp@0.1.0 camtimers@master catalyst@5.6.0 chai@2.3.0 charliecloud@0.22 cinch@develop cinch@master cmake@3.13.4 cmake@3.14.5 cmake@3.14.7 cmake@3.15.4 cmake@3.16.2

- 53,000+ binaries
- S3 mirror
- No need to build from source code!

- https://oaciss.uoregon.edu/e4s/inventory.html

EXASCALE COMPUTING PROJECT

WDMApp: Speeding up bare-metal installs using E4S build cache



E4S Spack build cache:

- WDMapp added E4S mirror
  - Speedup: 10X
- Pantheon: 10X
  - Another 10X via "smoother" installs
- Latest: ExaWind (Nalu-Wind)
  - 6 minutes with build cache
  - Up to 4 hours without

- https://wdmapp.readthedocs.io/en/latest/machines/rhea.html

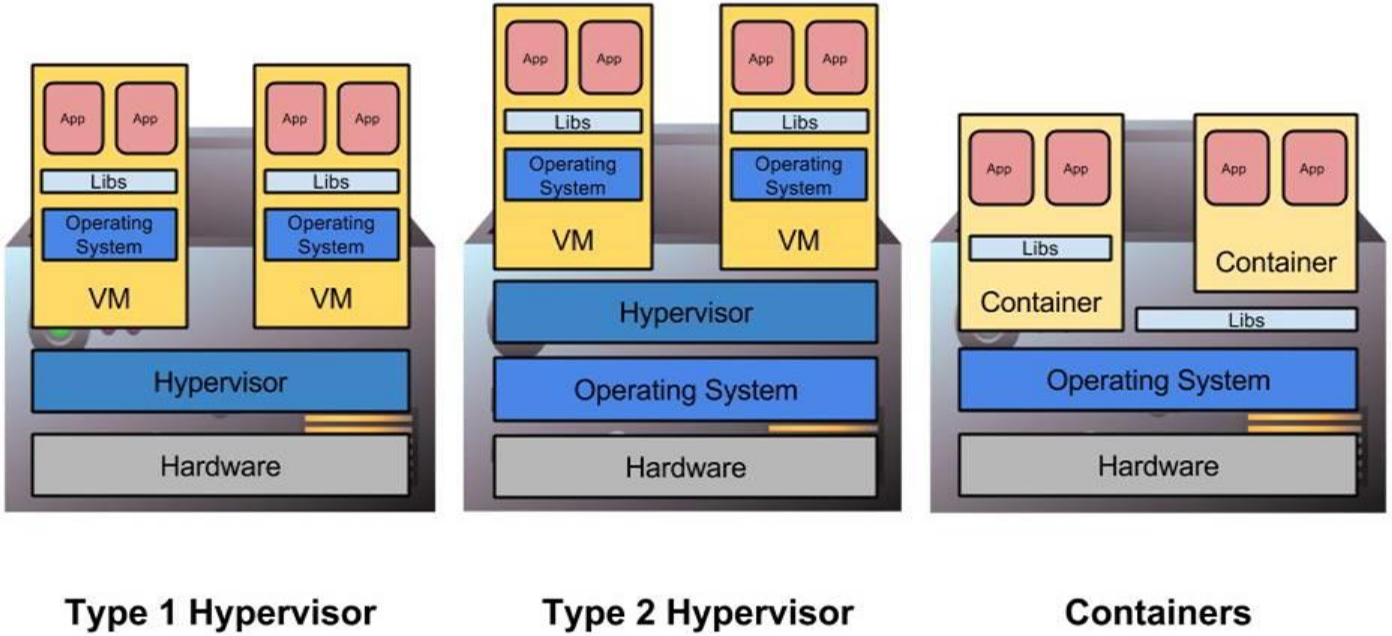
Using E4S with containers

What are containers

- A lightweight collection of executable software that encapsulates everything needed to run a single specific task
  - Minus the OS kernel
  - Based on Linux only
- Processes and all user-level software is isolated
- Creates a portable* software ecosystem
- Think `chroot` on steroids
- Docker most common tool today
  - Available on all major platforms
  - Widely used in industry
  - Integrated container registry via Dockerhub

Hypervisors and Containers

- Type 1 hypervisors insert layer below host OS
- Type 2 hypervisors work as or within the host OS
- Containers do not abstract hardware, instead provide “enhanced chroot” to create isolated environment
- Location of abstraction can have impact on performance
- All enable custom software stacks on existing hardware



Download E4S 2021-02 GPU Container Image

https://e4s-project.github.io/download.html

```
# docker pull ecpe4s/ubuntu18.04-e4s-gpu
```

RHEL 7	Ubuntu 18.04	CentOS 7
SPACK MINIMAL	**E4S GPU IMAGE**	**SPACK MINIMAL**
ecpe4s/rhel7-spack docker GitHub	• ecpe4s/ubuntu18.04-e4s-gpu docker GitHub x86_64 version: CUDA and ROCM ppc64le version: CUDA	ecpe4s/centos7-spack docker GitHub
E4S COMPREHENSIVE	**SPACK MINIMAL**	**E4S COMPREHENSIVE**
ecpe4s/rhel7-e4s docker GitHub	ecpe4s/ubuntu18.04-spack docker GitHub	ecpe4s/centos7-e4s docker GitHub
CUSTOM	**E4S COMPREHENSIVE**	**CUSTOM**
ecpe4s/superlu_sc docker GitHub	ecpe4s/ubuntu18.04-e4s docker GitHub	- - - -

https://e4s.io

E4S v2021-02 GPU Release for x86_64

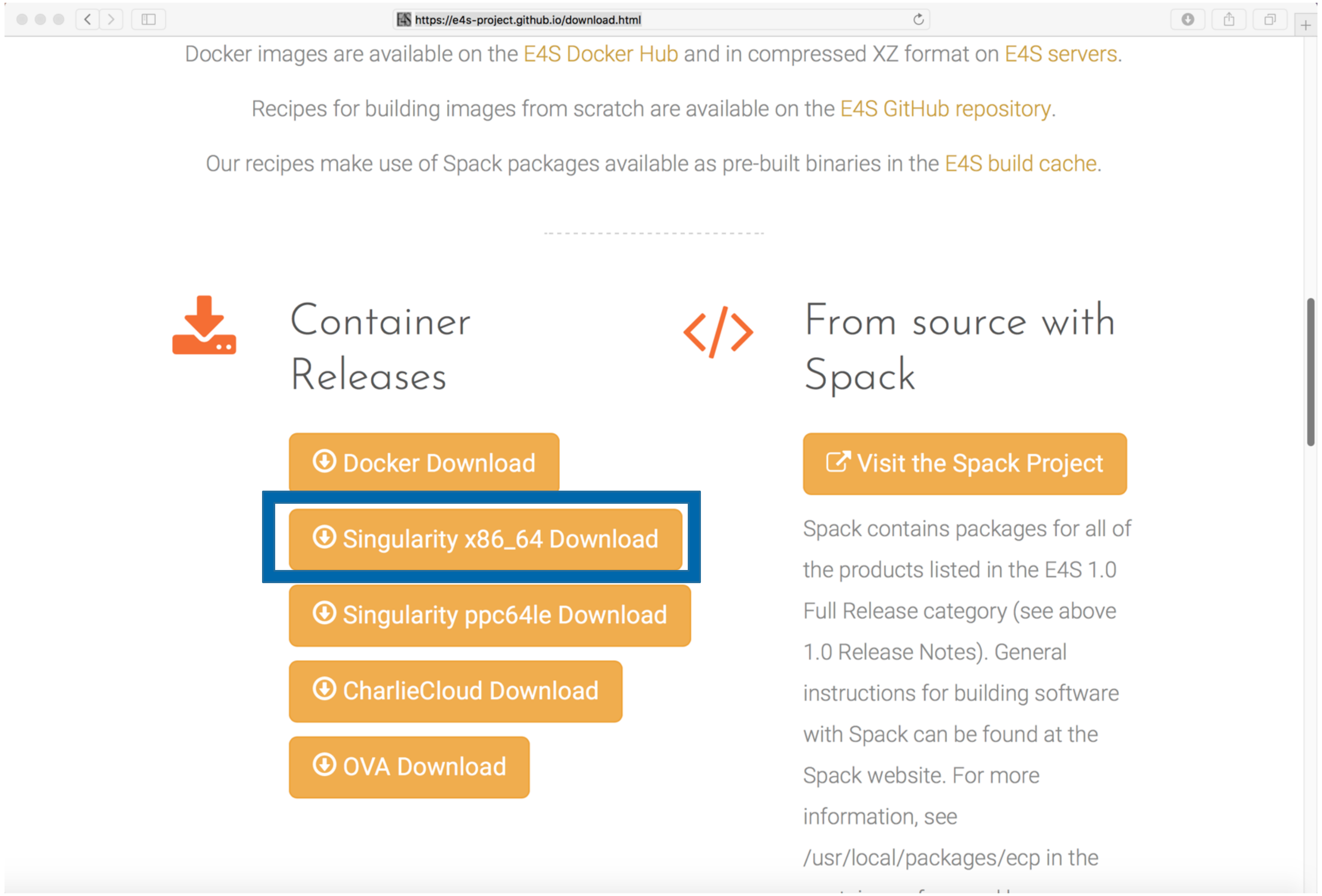
```
1: adios2       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/adios2-2.6.0-nkp24j7enorn3dt7626chuqm3pbkrvfe
2: aml  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/aml-0.1.0-3mwyb6cf6ervfnruqb5u33v46buyuqth
3: arborx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/arborx-0.9-beta-qjzxlkcgplto6pnpjwejoh5xpoik3adr
4: argobots     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/argobots-1.0-yoafg2slps7kp4dkmb6pzu5z2a37sgs4
5: ascent       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ascent-develop-ciwqg6lh6unww3hjsnu47wr7cpqptqgy
6: axom         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/axom-0.3.3-tzyejxpy3p3ekaev35k2bhpkr74cnuhh
7: bolt         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/bolt-1.0-uxku5w5qdfnpa4atgzcbraq7wop7lunc
8: caliper      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/caliper-2.4.0-lfdx3gc6qodg2abbpovib3thdsmsamnn
9: darshan-runtime      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/darshan-runtime-3.2.1-jqugqxx2uunyaduoe3owhd2snves6mlr
10: dyninst     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/dyninst-10.2.1-xad3v6rvosm6qfai5fc7d4nn33svtzzf
11: faodel      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/faodel-1.1906.1-ijilel2vjionmj56mscqkw2hpecfsuym
12: flecsi      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/flecsi-1-c7sevlnc2ak4pf2jgq6wh3mwictch5l2
13: flit        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/flit-2.1.0-yvvog7kmax22ei2yyrwfxj3heinmz5am
14: gasnet      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/gasnet-2020.3.0-ufrq5hym67eq3jsg4jtttjjqgo4i6hnq
15: ginkgo      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ginkgo-1.2.0-r6lorgchpr5qrcwyqqxtewqdhtpi4rmt
16: globalarrays        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/globalarrays-5.7-bow6d32j63j6gusotzjuityznwqvv64b
17: gotcha      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/gotcha-1.0.3-7n7bjnzsnf5w5tnihiok3otbaewdhjmu
18: hdf5        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hdf5-1.10.6-k74avubedd5knvlc73dr3ib5oyw6bcwn
19: hpctoolkit  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hpctoolkit-2020.08.03-wck4g3h3jhfvzvxorelxqunbe3xsesry
20: hpx         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hpx-1.5.0-pynmocntkmuwkowyo5jxtycvg34w6kue
21: hypre       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hypre-2.19.0-vqo72wn6ei7ruitpg7drkje2rdbdfguo
22: kokkos      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/kokkos-3.2.00-pqv3uugd6cv3qftyur3rx6dm2gao2tg3
23: kokkos-kernels      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/kokkos-kernels-3.1.00-y4veufypftworlbehxusg4yzh6n7anhp
24: legion      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/legion-20.03.0-zkbz7h2wuze4dgbwcbo4w5fvqltugmog
25: libnrm      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/libnrm-0.1.0-kp5jb7o4kow25rnggiditwtmdbeebojs
26: libquo      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/libquo-1.3.1-w45wcw6dqbiajeeauj3ryaeskqu7bzx6
27: magma       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/magma-2.5.3-yksxthffslhjrhzwgcx7smz2tca6ojfn
28: mercury     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mercury-1.0.1-ppledsr3drk2upciytfsuawfxrtjp73q
29: mfem        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mfem-4.1.0-kivaike2qintplgufwp5yf2mj3n36ay3
30: mpich       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mpich-3.2.1-kgwtpelzobpkrvg24ct6padfbhw7nene
31: mpifileutils        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mpifileutils-develop-djje5g7ts55g3yic3bms426c2zi7gqsj
32: ninja       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ninja-1.10.1-7zbbtuslw25nmqo4ur6abyyf3tchnqvv
33: omega-h     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/omega-h-9.29.0-eln73w7ytpvgqtkmkqyjm4gsabsu2w4p
```

- 67 ECP ST products
- Ubuntu v18.04 x86_64
- AI/ML package support
  - TensorFlow 2.3.5
  - PyTorch 1.8
  - Horovod
- Support for GPUs
  - AMD ROCm 3.8
  - NVIDIA CUDA 10.2, 11
  - Intel OneAPI 2021.1
- Kokkos with support for AMD GPUs!

E4S v2021-02 GPU Release for x86_64

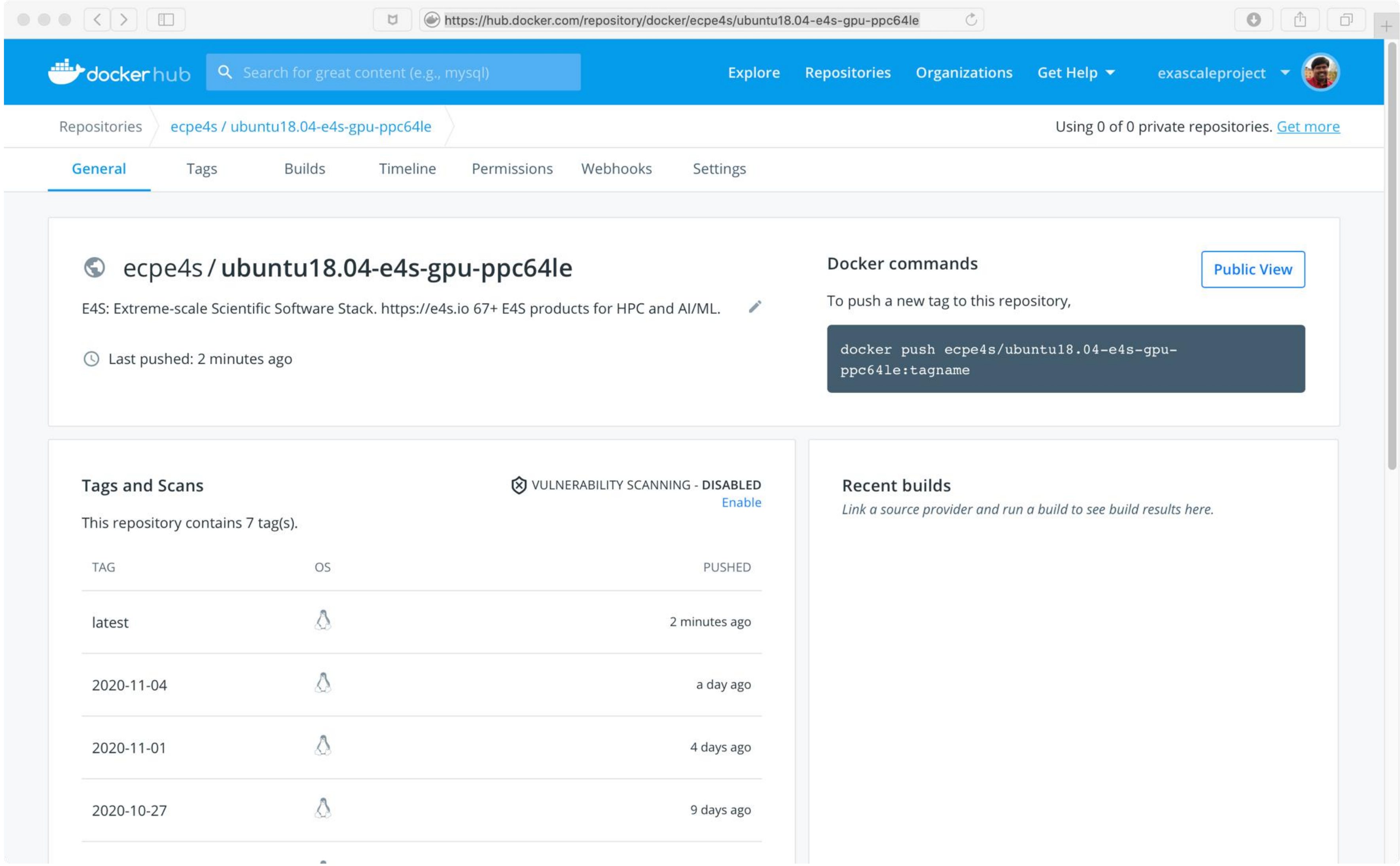
```
34: openpmd-api         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openpmd-api-0.12.0-4myph6pbjnupgupxdlvbxvqqeqx6atyp
35: openmpi     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openmpi-3.1.6-6yqtoym56as6xso2pdgkmn4bcsoyufku
36: papi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papi-6.0.0.1-gorrfrvrik575lldzgq46qmmu63kxl7x
37: papyrus     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papyrus-develop-iu3dgpmwwyykgv5mpw2dwcrol4wbwbai
38: parallel-netcdf     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/parallel-netcdf-1.12.1-tmmkzibn43xr7su76msxxusyzrphdtn5
39: pdt         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pdt-3.25.1-kvi5wuu5y72fypijti3nxqvdn7zpj6ni
40: petsc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/petsc-3.13.4-llg3u4rrt5axrqlim75tt73epewxu4fb
41: plasma      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/plasma-19.8.1-tji7bojb5ne5hqj2mwn5bqq2tfkm23ke
42: precice     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/precice-2.1.0-ozdmbat2hlivccha3nklbeahikgynewu
43: pumi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pumi-2.2.2-52czzdbxeg7pmjkd55nub5jgxzodcprh
44: py-jupyterhub       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-jupyterhub-1.0.0-tr3wcolaij3kbzb6xm4mbbvakcstwsw3
45: py-libensemble      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-libensemble-0.7.0-mxvqxhiiblqnmhlfepbxboyiskqyvbej
46: qthreads    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/qthreads-1.14-neshsclplh7ttkebm34grztaijqohnxt
47: raja        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/raja-0.11.0-w25bj2dys6cjqn7isgcjfyvte3tuulev
48: rempi       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/rempi-1.1.0-sideqdbiik2yseshs3loh4sictbis3t6
49: scr         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/scr-2.0.0-yh3chyq5gayuk6r4juejjiye6zg3rh3u
50: slate       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slate-develop-jnysy2rh5vxhwua5ubtvq4bsfd3py7d5
51: slepc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slepc-3.13.4-q3lalpbqoshiyvjjgrnhb2iqiisvnfrp
52: strumpack   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/strumpack-4.0.0-rlbti5eqc5rjhfisxv2uxevj6m3fn5gg
53: sundials    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sundials-5.3.0-3g52gh4a6h4ohucqart5i4m6pi66woj6
54: superlu-dist        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/superlu-dist-6.3.1-o2hkund66coxn2rrbtlalda2vq35uu7j
55: stc         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/stc-0.8.3-oxfik7nsmgufoqyy7xilzsrct7it63ej
56: swig        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/swig-4.0.1-htxmzjd5sed5yfibw6j7jn5cx6p7g72x
57: sz  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sz-2.1.9-tcatyiuzh6quctrgd2g3dcli7xa7gvtj
58: tasmanian   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tasmanian-7.1-quo3grs5kb2xrvjufpi7vn66cpjfnadv
59: tau         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tau-2.29-ijw2nbphmlfkt42ubwz7g5a5yru22ikn
60: trilinos    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/trilinos-13.0.0-6xfnp44g5xm7gpn2en6gkwzfceykfd3x
61: turbine     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/turbine-1.2.3-q4qjvgxjl3cbuyquo6zrurb4mwfn6wkp
62: umap        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umap-2.0.0-5tob3exzrmwoitudu5pstbb2dms3xnto
63: umpire      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umpire-3.0.0-6woo2uuvazcucxikc6xad6g3zksu2ygi
64: unifyfs     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/unifyfs-0.9.0-be7mqbng7kdeewdlgvlhdm4jkxnquiil
65: upcxx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/upcxx-2020.3.0-pshe62qyvmnrvesqa4pkj6bdq3fxxucf
66: veloc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/veloc-1.4-gk3iwfjhmgkwlawp7rmxf2eh37rqpqm2
67: zfp         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/zfp-0.5.5-6r6yaco7gga5w4gbuvid3zt2iohrnepj
```

E4S Support for Singularity Container Runtime [Sylabs.io]



- wget http://tau.uoregon.edu/ecp.simg; singularity run ./ecp.simg
- singularity run ecp.simg
- Supports Intel OneAPI, CUDA, and ROCm
- spack find

E4S v2021-02 Release: GPU, ppc64le for Docker Containers



- 67 ECP Products
- Support for GPUs
  - NVIDIA (CUDA 10.2)
  - ppc64le and x86_64

% docker pull ecpe4s/ubuntu18.04-e4s-gpu

E4S v2021-02 GPU Release: 67 E4S Products (ppc64le)

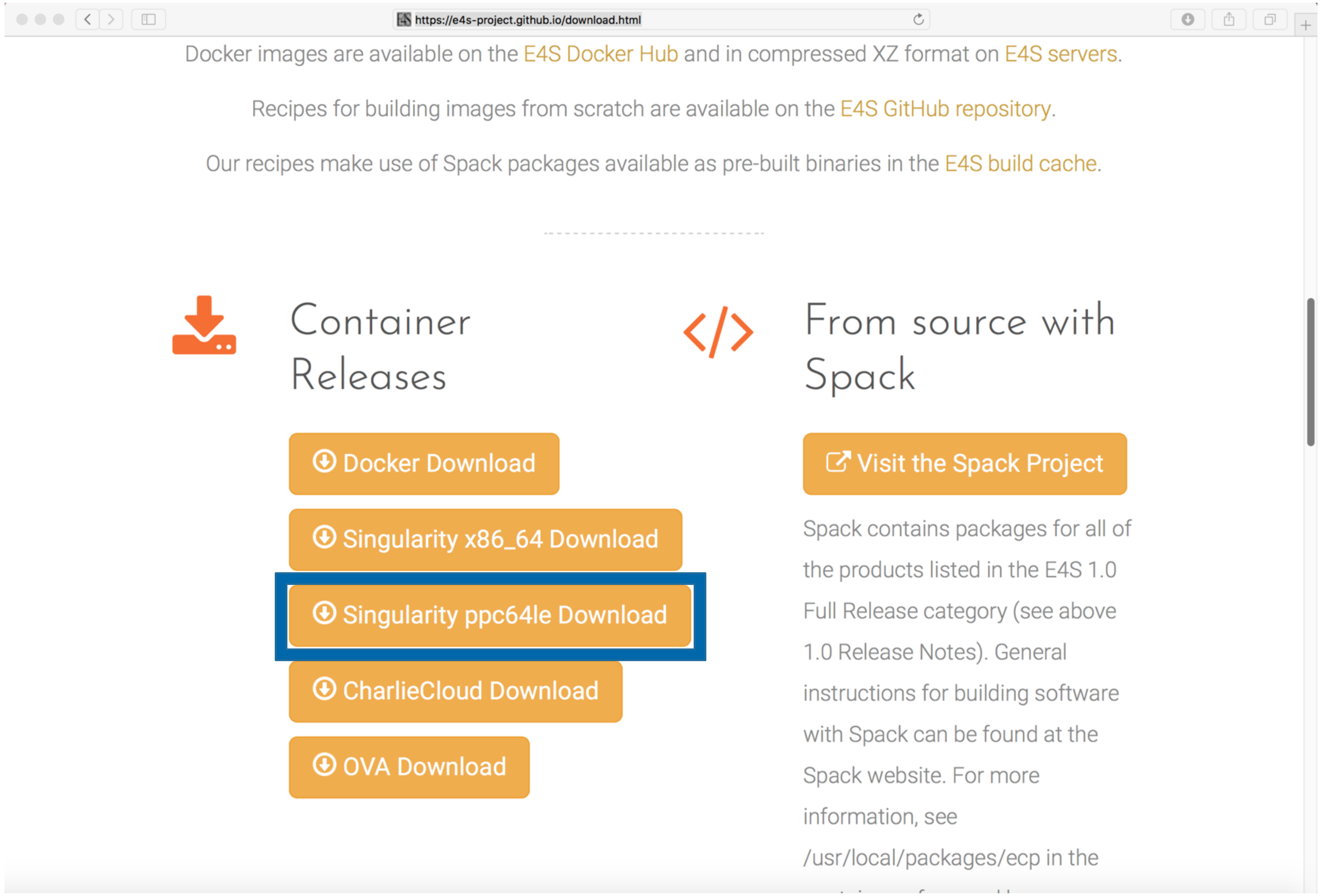
```
1: adios2       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/adios2-2.6.0-veoqi5iqkx4kbeddhxoroggvxqqbtvos
2: aml  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/aml-0.1.0-ftizegmvpbweuyzg75g3ndzhdyjx37op
3: amrex        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/amrex-20.10-4z5quvlqt3fbzv5n6rrjv5byq7472emy
4: arborx       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/arborx-0.9-beta-p7lw7eobsrdpqwhb7ispxgphng2tn4nt
5: ascent       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ascent-develop-7ktzsmvluqvd4xzoop7hjwddyjetn2ai
6: axom         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/axom-0.3.3-zfgqs6qa6vxlodjnaojeffmyl26czmp5
7: argobots     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/argobots-1.0-qra2gqxuisqqlbdfrhwm5mvq2iga3l3l
8: bolt         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/bolt-1.0-ojy67rk47pcbqpcvuq6a4c7g7qysvndv
9: caliper      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/caliper-2.4.0-6xzehuxs2updvdrl2tdvcym3n6nf3y3l
10: darshan-runtime     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/darshan-runtime-3.2.1-6uzihv7v75yu47c2jca4qpxqdtgptn2g
11: dyninst     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/dyninst-10.2.1-jvqx4j3ehuh73pp67b4vdy4co3kivma5
12: faodel      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/faodel-1.1906.1-r77asm5xkb256omn4trg5hnxc3e376uy
13: flecsi      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/flecsi-1-2kxukdrijujvbmsabmmj3um54ukhrayk
14: flit        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/flit-2.1.0-tepzltg6kmefdg4eo2rbzwmjeca56bmc
15: gasnet      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/gasnet-2020.3.0-uynuhs6itzczkfpgbnlm2xgotvqmmeb6
16: ginkgo      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ginkgo-1.3.0-dodvdbixjpdg5ci5xrgomjeqybiob33i
17: globalarrays        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/globalarrays-5.7-3zbsvrakwto5jc454jl3l36rpvray25h
18: gotcha      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/gotcha-1.0.3-pvjdzcg3fggpagjcsorwidsllflmomnz
19: hdf5        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hdf5-1.10.6-arkkhmy4auglzqndt7xraupyvgkrpv7o
20: hpctoolkit  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hpctoolkit-2020.08.03-yqayfprp2aleaxtzq543c75lcvcviso7
21: hpx         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hpx-1.5.1-tzfs3nkglsacequjxflokigwgjzabybk
22: hypre       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hypre-2.20.0-ewmv445dkzmju4upg4rregq7apgkcdbu
23: kokkos      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/kokkos-3.2.00-3qzjrzoxl5lpqgtaq4atid6ylgkko3uk
24: kokkos-kernels      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/kokkos-kernels-3.2.00-n4trpqubmexqahdy4tolj6nhfml5j4v6
25: legion      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/legion-20.03.0-xsotehq7eg77hcguvqx5qymfhimgtuic
26: libnrm      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/libnrm-0.1.0-q67khfosljacbl3djdg5jeh4thsl5p5f
27: libquo      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/libquo-1.3.1-syjf6c3adia34wlwneacynrwkhh72i3u
28: magma       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/magma-2.5.4-fzeektdrkybbuo6i6niikzglcwlnt2jx
29: mercury     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mercury-1.0.1-ufxkkvb7osjnwgbfevdhtrmtuoj6dfbz
30: mfem        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mfem-4.1.0-qrepufdzopbphsyuyc6npn7k2tpprd5w
31: mpich       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mpich-3.2.1-5m7ofmtvtov45hcudrm3qvd2dyheunyv
32: mpifileutils        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mpifileutils-develop-rd5xj2pmx5vdd7fddrhbrvn2uykg4uay
33: ninja       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ninja-1.10.1-cr2ada5fjgvkvdtmxel4zj6venfiif5e
```

EXASCALE COMPUTING PROJECT

E4S v2021-02 GPU Release: 67 E4S Products (ppc64le)

```
34: omega-h     /opt/spack/opt/spack/linux-ubuntu18.04-power9le/gcc-7.3.0/omega-h-9.29.0-ziz55mnp5r7l4kuhx4zqmjp2imjdvrk5
35: openmpi     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/openmpi-3.1.6-utceq6uech6rgnabxevau4lhtzrwbaol
36: openpmd-api         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/openpmd-api-0.12.0-szt65gmfb76iwdbcfkhryfztg5jwjd7g
37: papi        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/papi-6.0.0.1-xu35qtffffq2ofyjic3fafmj6yeijoih
38: papyrus     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/papyrus-develop-2zopf6p3ha4v7ijxslxskrf2qyhpt3py
39: parallel-netcdf     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/parallel-netcdf-1.12.1-svuejkorgi2bzvhgq4wts72bcjfn426r
40: pdt         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/pdt-3.25.1-opxwliyf5vqgt3hbla7qspf3laaqbt74
41: petsc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/petsc-3.14.0-phaqc52ryvhcib37qqjg2lmqdebql2uo
42: plasma      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/plasma-20.9.20-vc4olrzgwsvqx7mevom2j7mhsgb6ynam
43: precice     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/precice-2.1.1-qlitin5qdhtz3n7rg4jjzxkdss4qocvn
44: pumi        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/pumi-2.2.2-m4uipa7yh632dftix4kzyxcz3pm3fasv
45: py-jupyterhub       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-jupyterhub-1.0.0-gzzlya6f4gr2xgsgpndmbp2pkffm3tuc
46: py-libensemble      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-libensemble-0.7.1-oee4zlxigkjc5nnkr6fyu7thzsnftvvu
47: py-petsc4py         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-petsc4py-3.13.0-g2rp2v37qbpx5fo5fmg6c4xtrj6shsbz
48: qthreads    /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/qthreads-1.14-bdxplr2gf7knpek4vo5sjvzh5py5fdaf
49: raja        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/raja-0.12.1-q32nuxmeowavkwzmoiwx6f5md246tw66
50: rempi       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/rempi-1.1.0-h3x5q2rwwsv34v7e4ricjw65wcd5mvkg
51: scr         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/scr-2.0.0-2okrlxki5b63gzakjy2x4sbovrmegmcx
52: slate       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/slate-develop-2jp7v35nifhyucbf4vmi3mjsernm5t26
53: slepc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/slepc-3.14.0-7qn6k5qxzf32tc2cnuk2mknlvqvv6hfw
54: strumpack   /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/strumpack-5.0.0-gy5opc36suubh6uoiqy4l223psdyrilg
55: sundials    /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/sundials-5.4.0-wonraynurs6xhyv6m6bc7o4grlwchlnp
56: superlu-dist        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/superlu-dist-6.3.1-poufv43kq7tw2rw6upldbpcpabkpbdta
57: swig        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/swig-4.0.2-3bddrfojvkrowa43v5so3ongbmhzxx5s
58: sz  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/sz-2.1.10-fld5xazn2spjg46yaaaam5gftgyb5loa
59: tasmanian   /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/tasmanian-7.3-zbz26kn2yabritfi2wsbqv5raexgi4p3
60: tau         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/tau-2.29-zqbkmoraislptbdny6fw4pakoipm3cbv
61: trilinos    /opt/spack/opt/spack/linux-ubuntu18.04-power9le/gcc-7.3.0/trilinos-13.0.0-olf4mdmym4sjbgue66gx42k7dbeb6z27
62: turbine     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/turbine-1.2.3-jy42tjmn7rd2ofwwb3jaanlri2hnte65
63: umpire      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/umpire-4.0.1-ynagdhefpcujnpeybxtasoqecr2p7bxj
64: unifyfs     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/unifyfs-0.9.0-sxswh3b5upcys4bxc5wdzczvwxvn6emg
65: upcxx       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/upcxx-2020.3.0-i6hf7mat23um3fz5wexqswvn6mm4o7zp
66: veloc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/veloc-1.4-7ygadmpwv2zr26ec6opicysts4mxkwym
67: zfp         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/zfp-0.5.5-3r4a4s3qdeqbdabvlwlswrgig62yc6yj
```

E4S Support for Singularity Container Runtime [Sylabs.io]



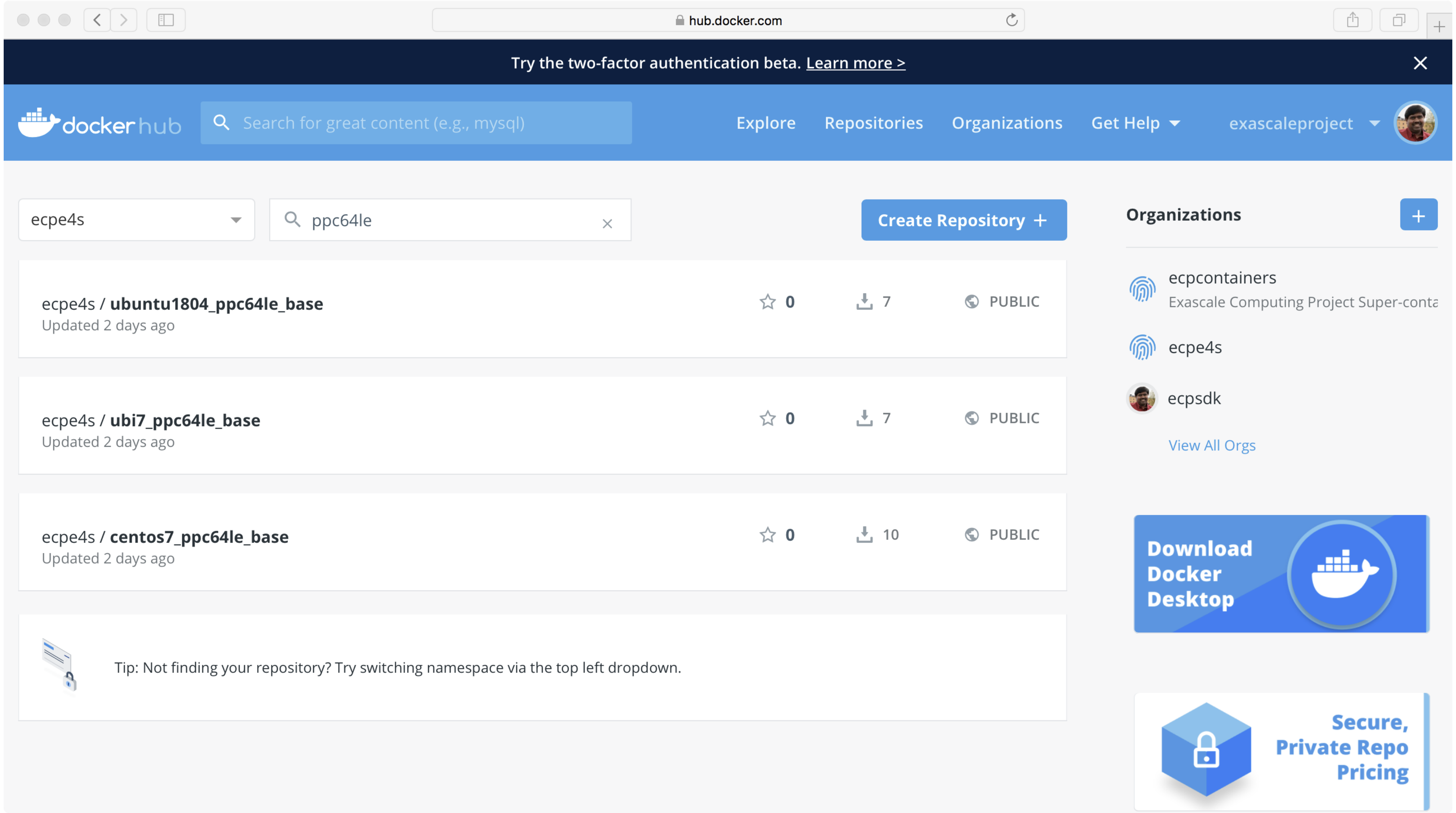
- wget http://oaciss.uoregon.edu/e4s/images/ubuntu18.04-e4s-gpu-ppc64le_1.2.simg
- singularity exec --nv ubuntu18.04-e4s-gpu-ppc64le_1.2.simg /bin/bash --rcfile /etc/bashrc
- spack find; module avail

EXASCALE COMPUTING PROJECT

E4S v2021-02 GPU Support

```
alias runsi='singularity exec --nv /home/users/sameer/images/ubuntu18.04-e4s-gpu-ppc64le_1.2.simg /bin/bash --rcfile /etc/bashrc'
[sameer@gorgon ~]$ runsi
Singularity> python
Python 3.6.10 |Anaconda, Inc.| (default, Jan  7 2020, 21:47:07)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
>>> import torch
>>> import cv2
>>> import matplotlib
>>> import numpy
>>> tensorflow.test.is_gpu_available()
2020-11-05 17:09:35.705979: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0004:04:00.0
totalMemory: 31.75GiB freeMemory: 12.35GiB
2020-11-05 17:09:35.778351: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 1 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0004:05:00.0
totalMemory: 31.75GiB freeMemory: 31.44GiB
2020-11-05 17:09:35.907371: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 2 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0035:03:00.0
totalMemory: 31.75GiB freeMemory: 883.50MiB
2020-11-05 17:09:35.989499: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 3 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0035:04:00.0
totalMemory: 31.75GiB freeMemory: 31.44GiB
2020-11-05 17:09:35.989594: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0, 1, 2, 3
2020-11-05 17:09:45.948104: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 ed
ge matrix:
2020-11-05 17:09:45.948182: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990]      0 1 2 3
2020-11-05 17:09:45.948199: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0:   N Y Y Y
2020-11-05 17:09:45.948210: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 1:   Y N Y Y
2020-11-05 17:09:45.948222: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 2:   Y Y N Y
2020-11-05 17:09:45.948232: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 3:   Y Y Y N
2020-11-05 17:09:45.950552: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/device:GPU:0 with 11587
MB  Snapz Pro X  physical GPU (device: 0, name: Tesla V100-SXM2-32GB, pci bus id: 0004:04:00.0, compute capability: 7.0)
```

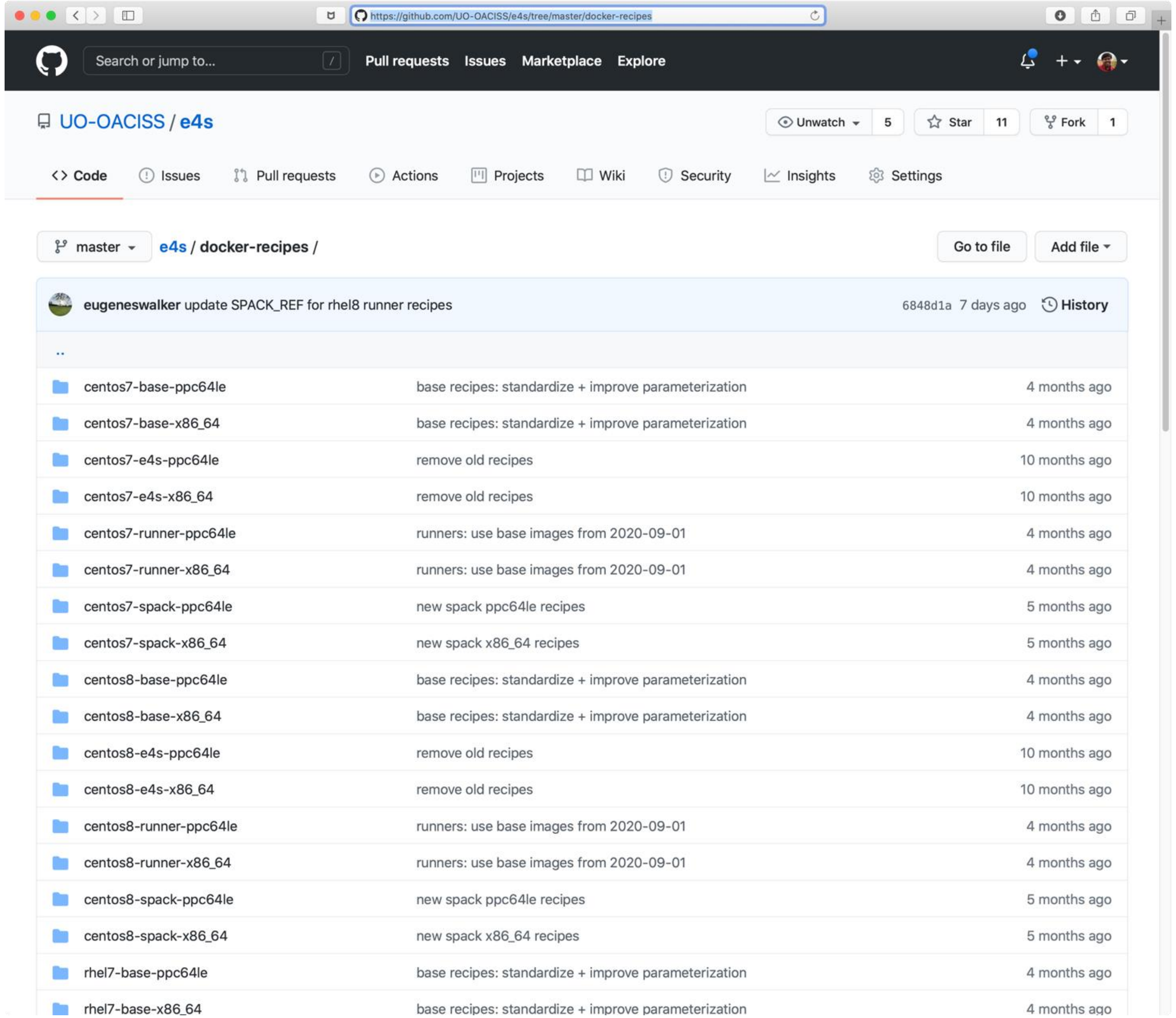
E4S: ppc64le Base Container Images



- Hub.docker.com
- ecpe4s

- Ubuntu 18.04
- RHEL/UBI 7.6
- Centos 7.6

Multi-platform E4S Docker Recipes

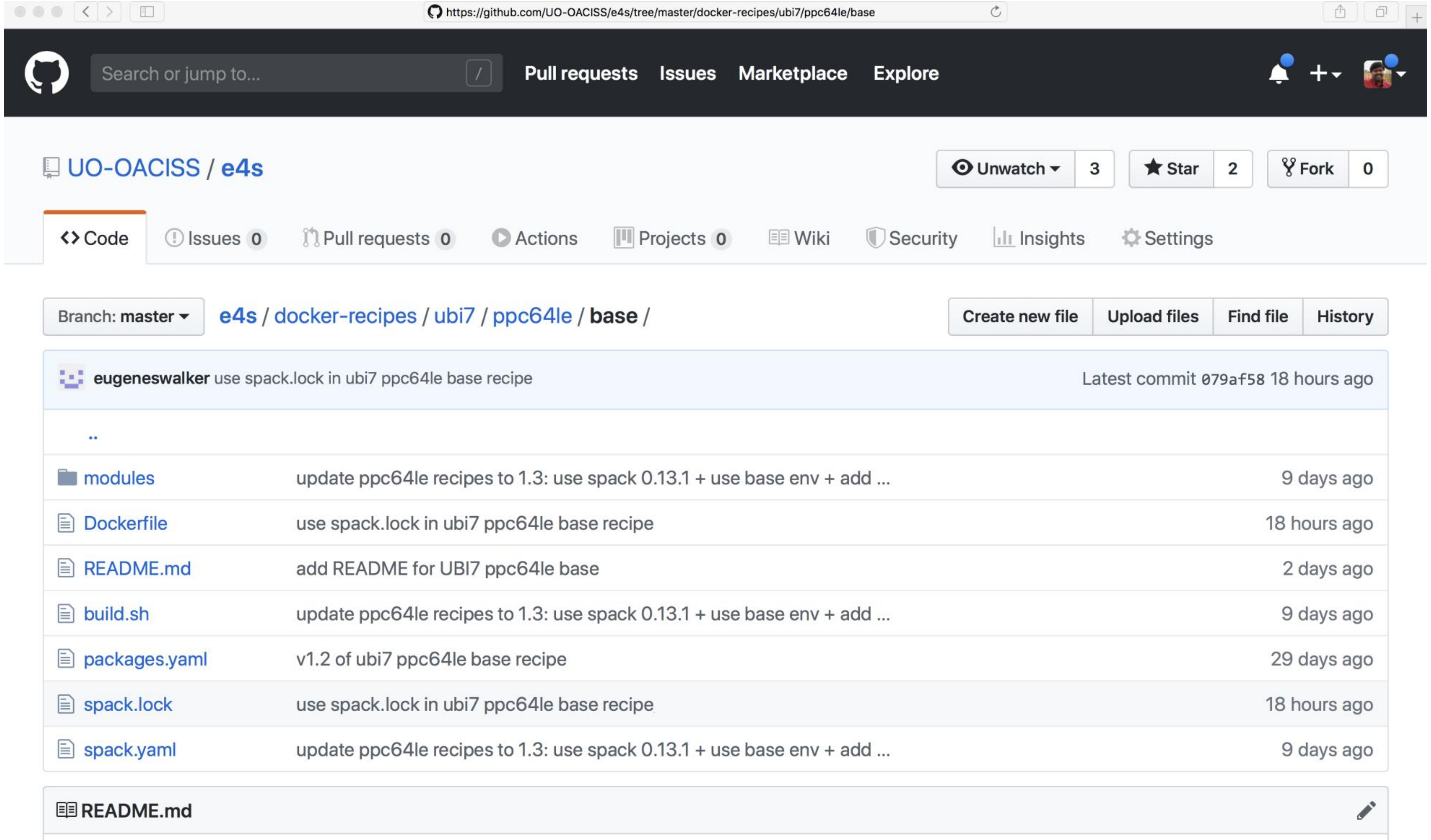


10 lines (6 sloc) | 178 Bytes

```
FROM ecpe4s/ubuntu18.04-spack-x86_64:0.14.1

WORKDIR /e4s-env

COPY /spack.yaml .

RUN spack install --cache-only \
 && spack clean -a && rm -rf /tmp/root/spack-stage

WORKDIR /
```

EXASCALE COMPUTING PROJECT

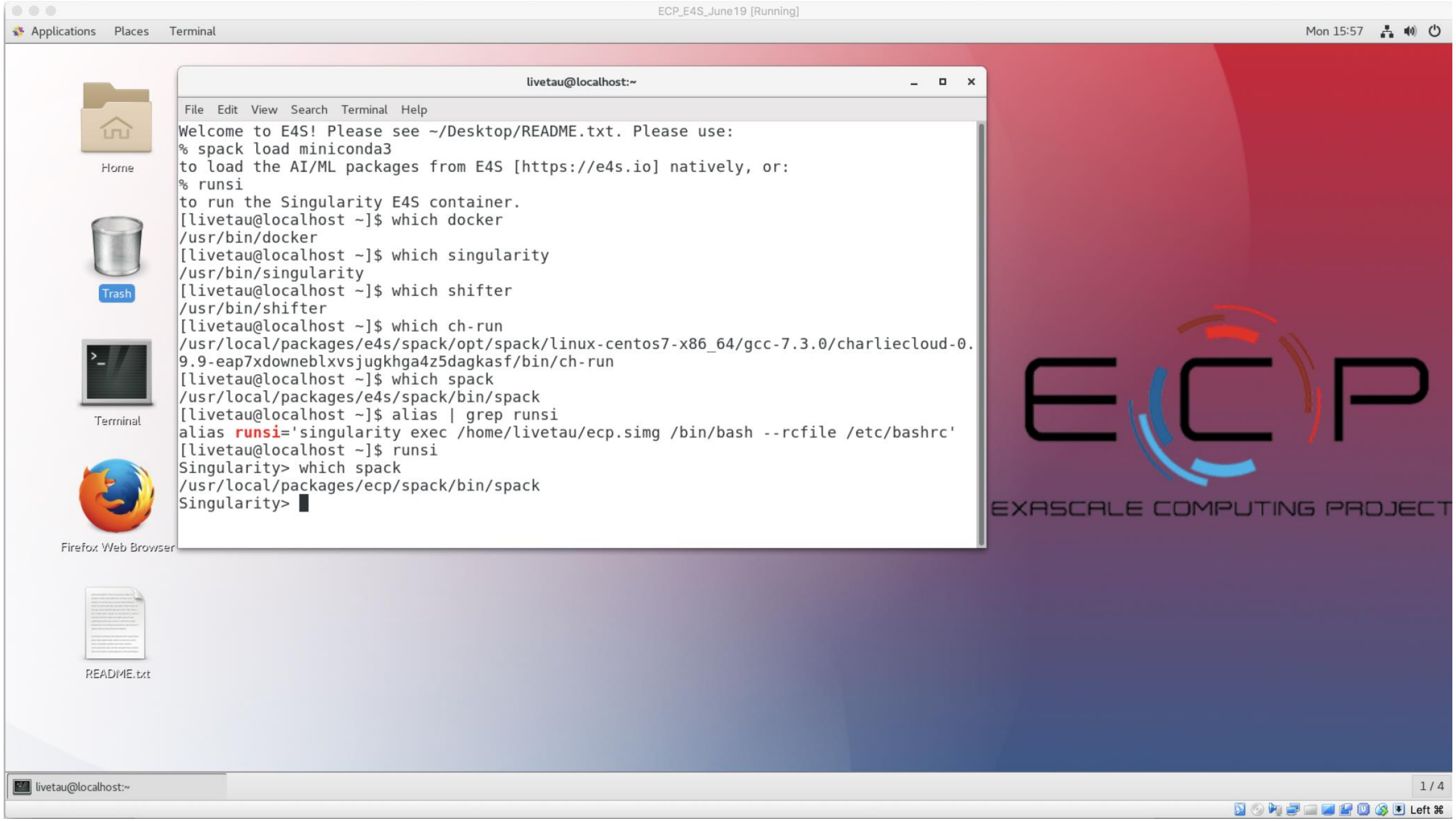
E4S: Multi-platform Reproducible Docker Recipes



E4S
- x86_64
- ppc64le
- aarch64

https://e4s.io

E4S VirtualBox Image



Container Runtimes

- Docker
- Shifter
- Singularity
- Charliecloud

https://e4s.io

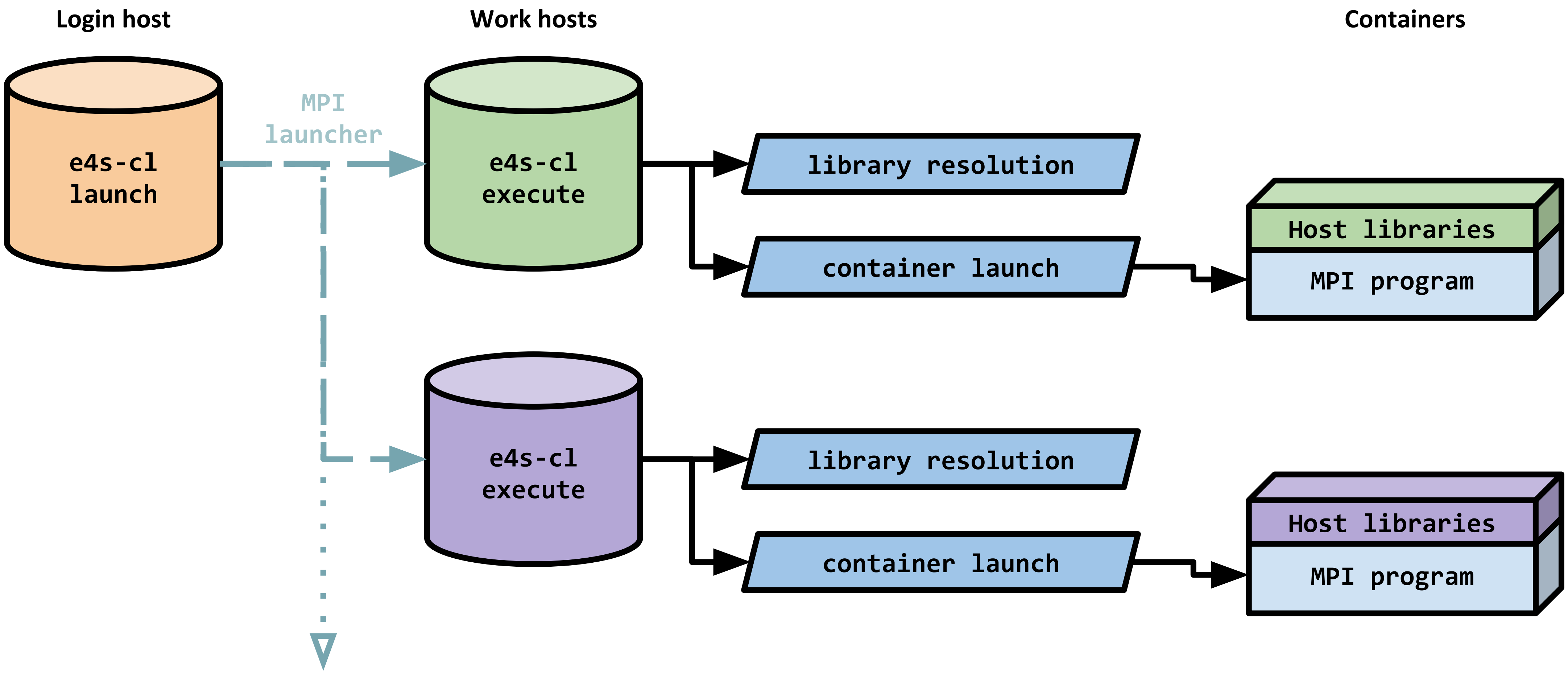
e4s-cl: A tool to simplify the launch of MPI jobs in E4S containers

- E4S containers support replacement of MPI libraries using MPICH ABI compatibility layer.
- Applications binaries built using E4S can be launched with Singularity using MPI library substitution for efficient inter-node communications.
- e4s-cl is a new tool that simplifies the launch and MPI replacement.
- Usage:

  1. `e4s-cl init …`
  2. `e4s-cl mpirun -np <> -hosts <> <command>`

https://github.com/E4S-Project/e4s-cl

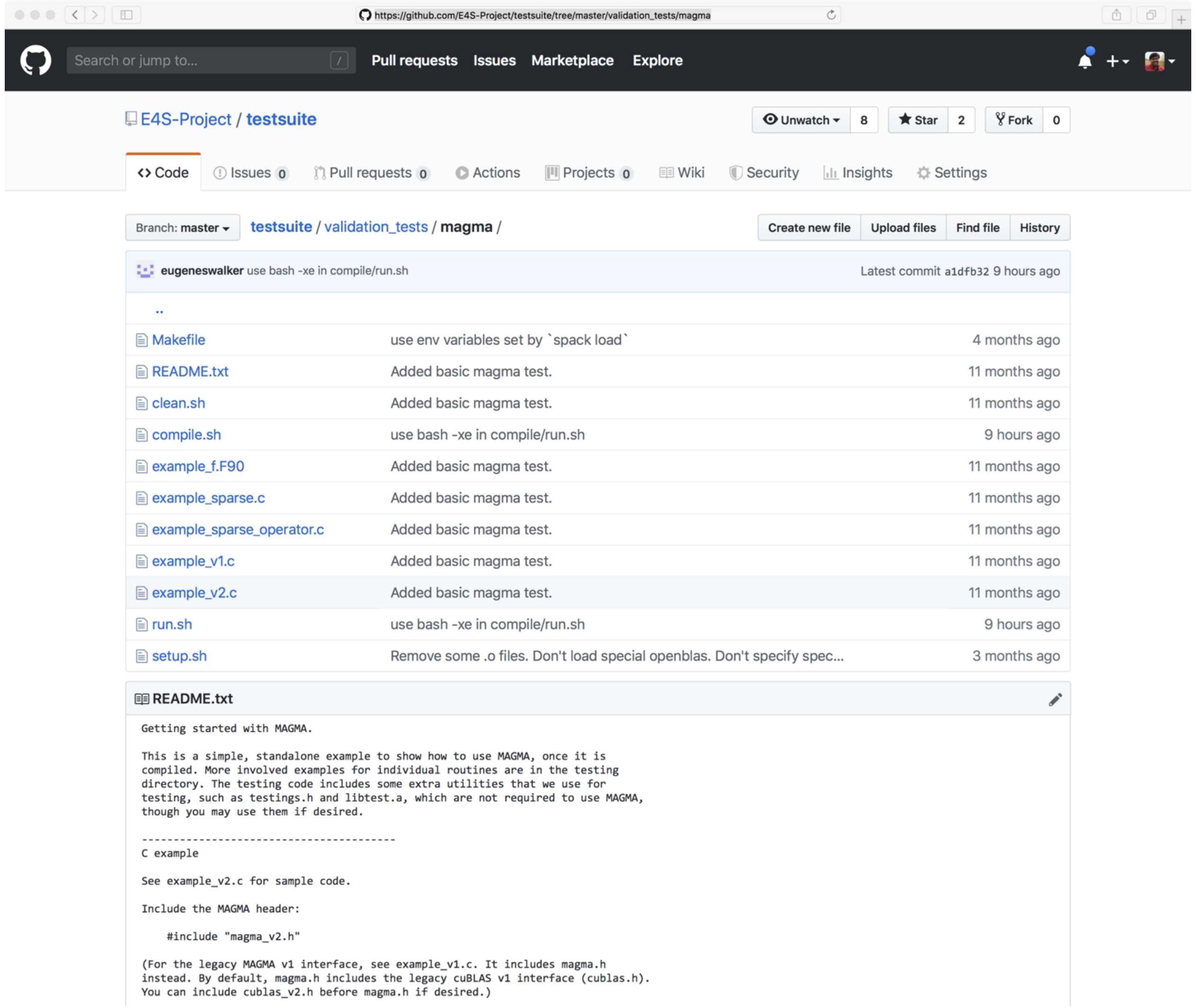
e4s-cl Container Launcher



E4S Continuous Integration Testing

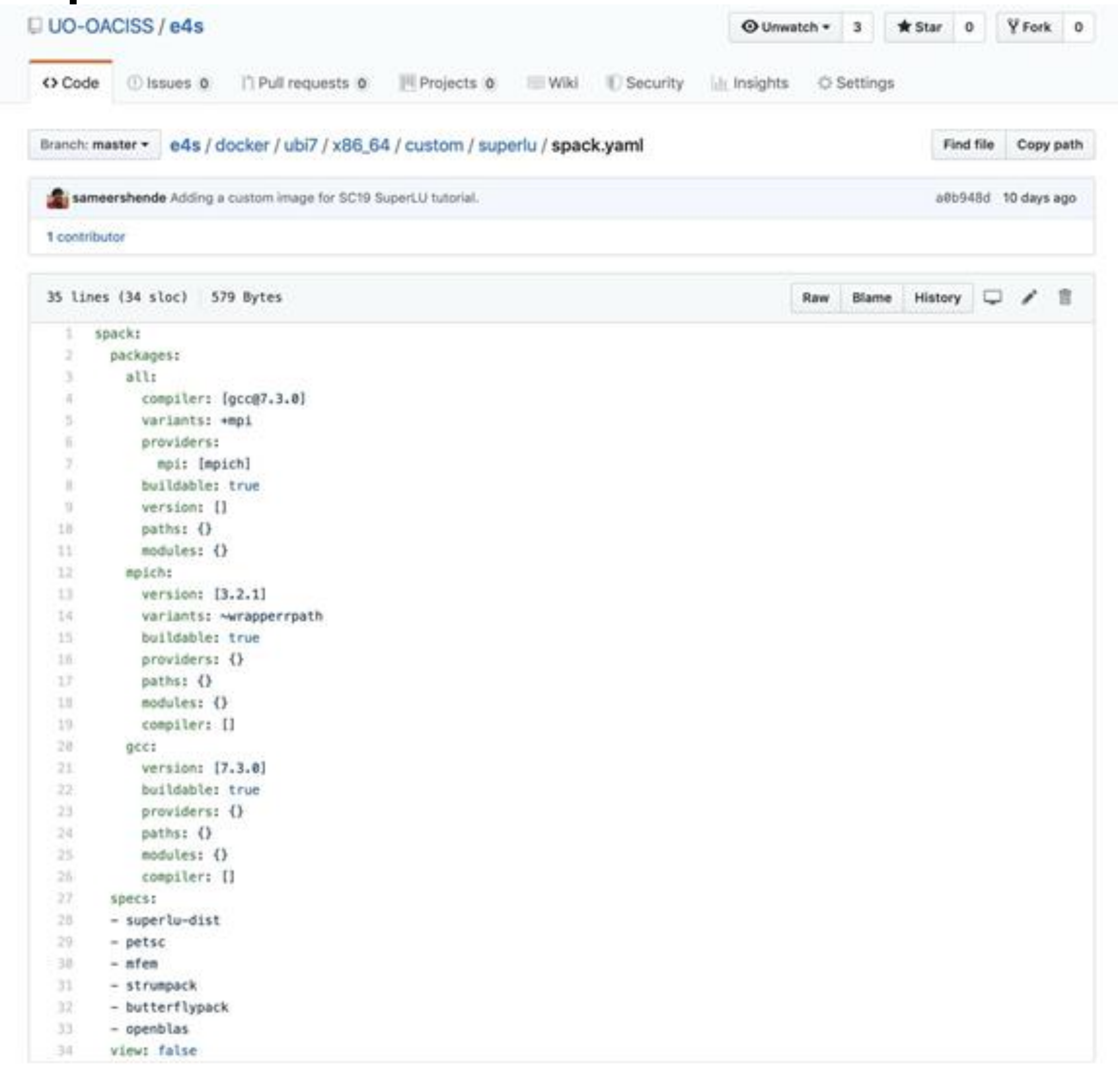
E4S Validation Test Suite

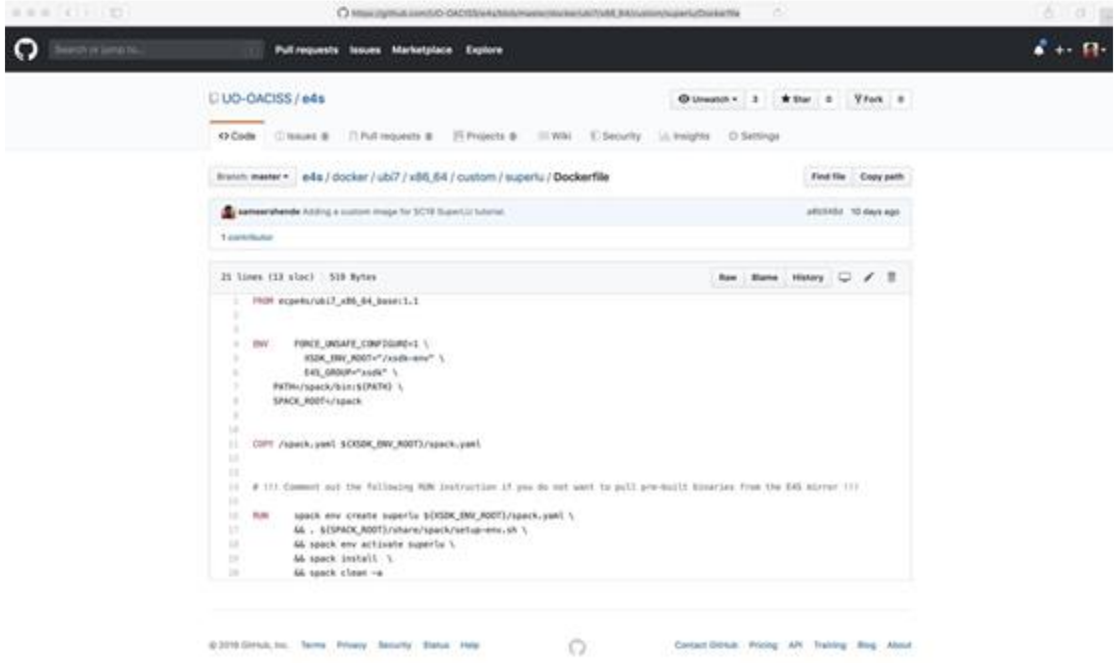
- Provides automated build and run tests
- Validate container environments and products
- New LLVM validation test suite for DOE LLVM



- git clone https://github.com/E4S-Project/testsuite.git

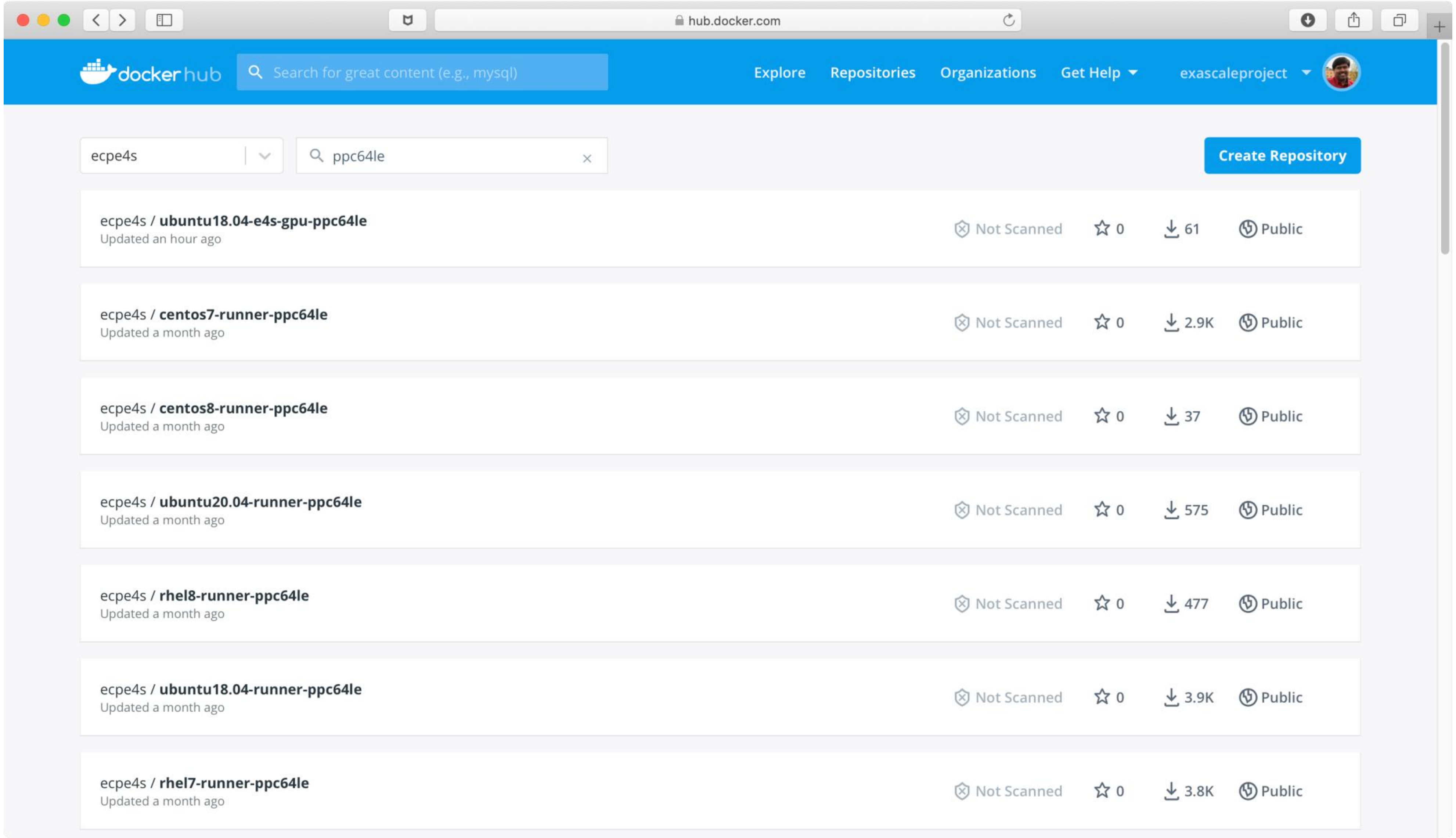
Reproducible Container Builds using E4S Base Images





- PMR SDK base image has Spack build cache mirror and GPG key installed.
- Base image has GCC and MPICH configured for MPICH ABI level replacement (with system MPI).
- Customized container build using binaries from E4S Spack build cache for fast deployment.
- No need to rebuild packages from the source code.
- Same recipe for container and native bare-metal builds with Spack!

E4S: GitLab Runner Images



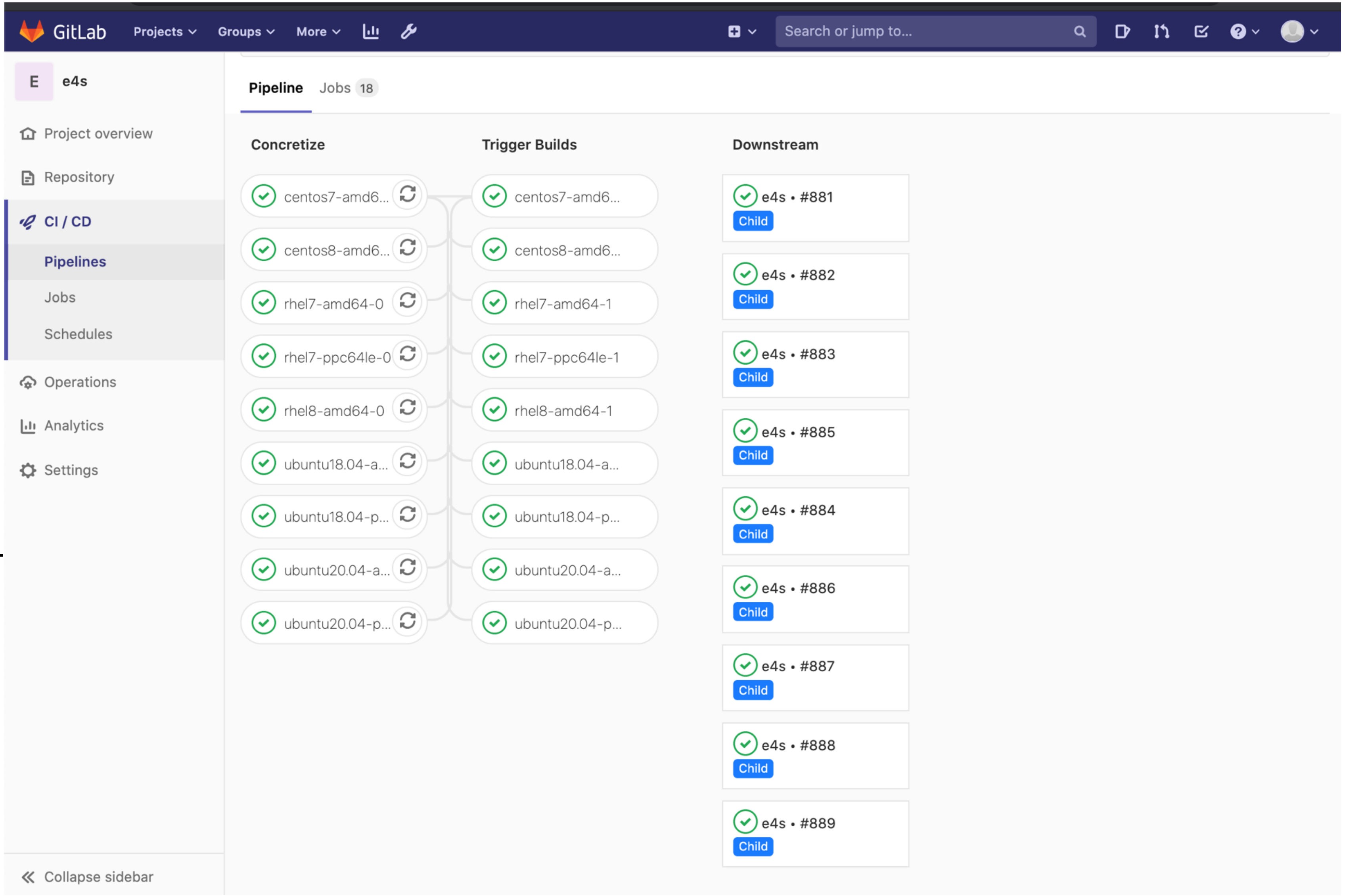
- Dockerhub
- Bare-bones
- Multi-platfrom
- Build E4S

EXASCALE COMPUTING PROJECT

University of Oregon GitLab CI

E4S Builds:
- Ubuntu 18.04
- Ubuntu 20.04
- RHEL 7.6
- RHEL 8
- CentOS 7
- CentOS 8

Architectures:
ppc64le and x86_



- https://gitlab.e4s.io

EXASCALE COMPUTING PROJECT

GitLab GPU Runners on Frank, U. Oregon

Recent searches | uo-illyad | Created date

Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
shared locked paused	9yUY6sjq	uo-illyad	13.9.0	192.168.165.1...	n/a	0	a100 avx avx2 large medium nvidia-gpu public small spack uo x86_64 xlarge	3 minutes ago

A100 NVIDIA GPU

Recent searches | uo-jupiter | Created date

Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
specific locked paused	E_zdx2ry	uo-jupiter	13.9.0	192.168.165.1...	2	105	avx avx2 cooper-lake dg1 huge large medium public small spack uo x86_64 xlarge	4 minutes ago

DG1 Intel GPU

Recent searches | uo-instinct | Created date

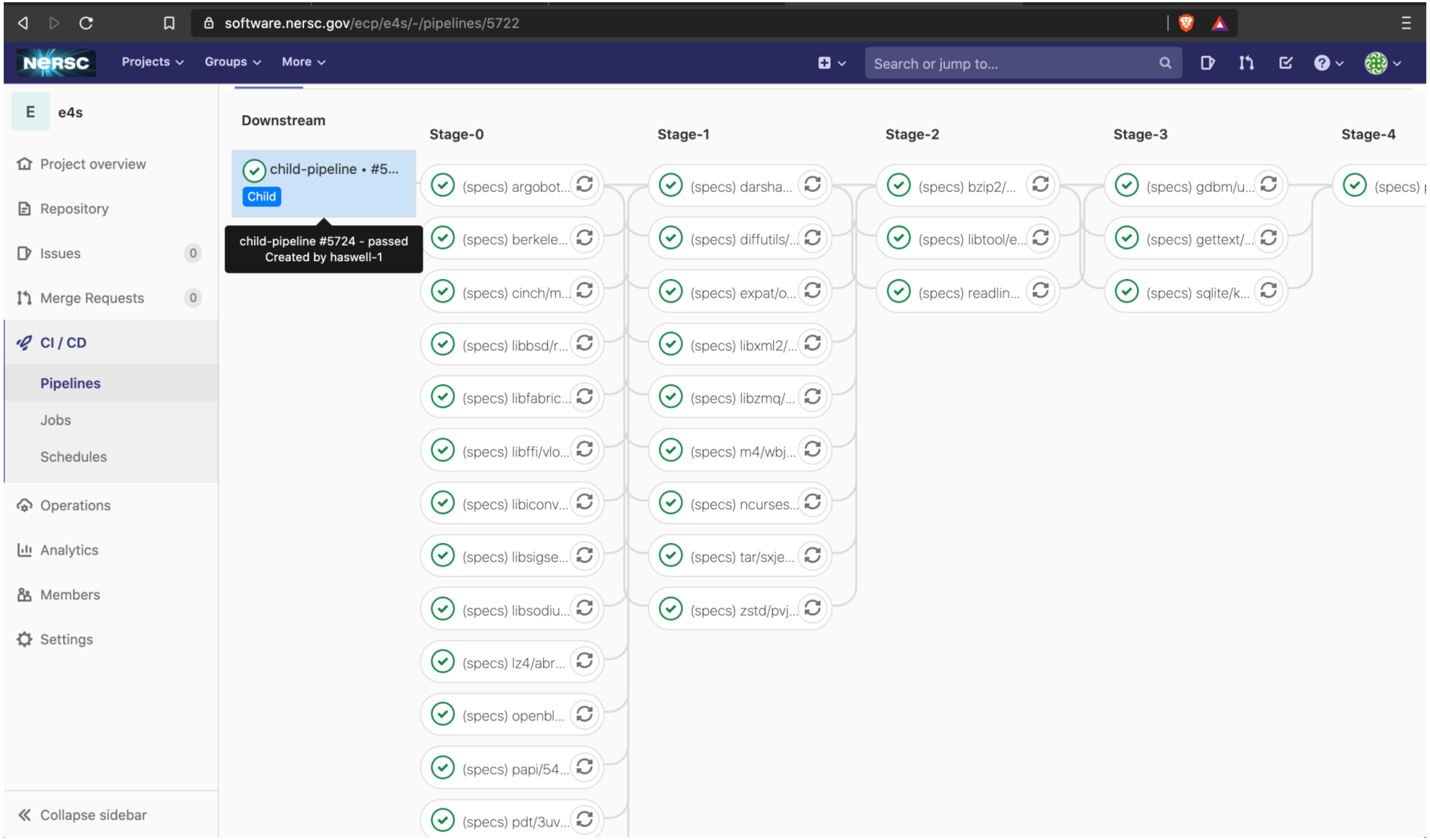
Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
shared locked paused	xPVjy9oY	uo-instinct	13.9.0	192.168.165.1...	n/a	0	amd-gpu avx avx2 huge large medium mi50 public small spack uo x86_64 xlarge	5 minutes ago

MI50 AMD GPU

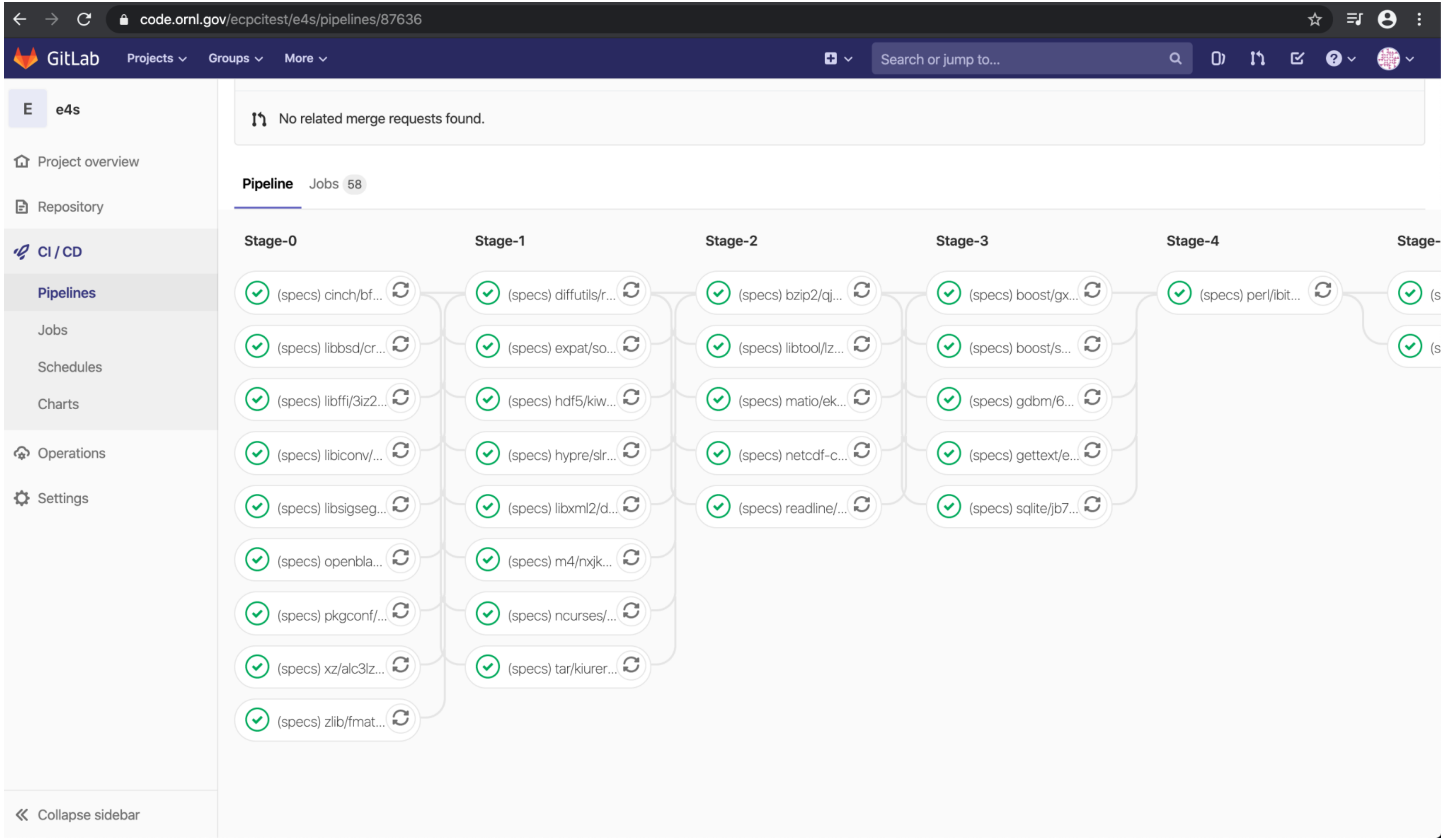
EXASCALE COMPUTING PROJECT

Multi-stage E4S CI Build Pipeline on Cori, NERSC



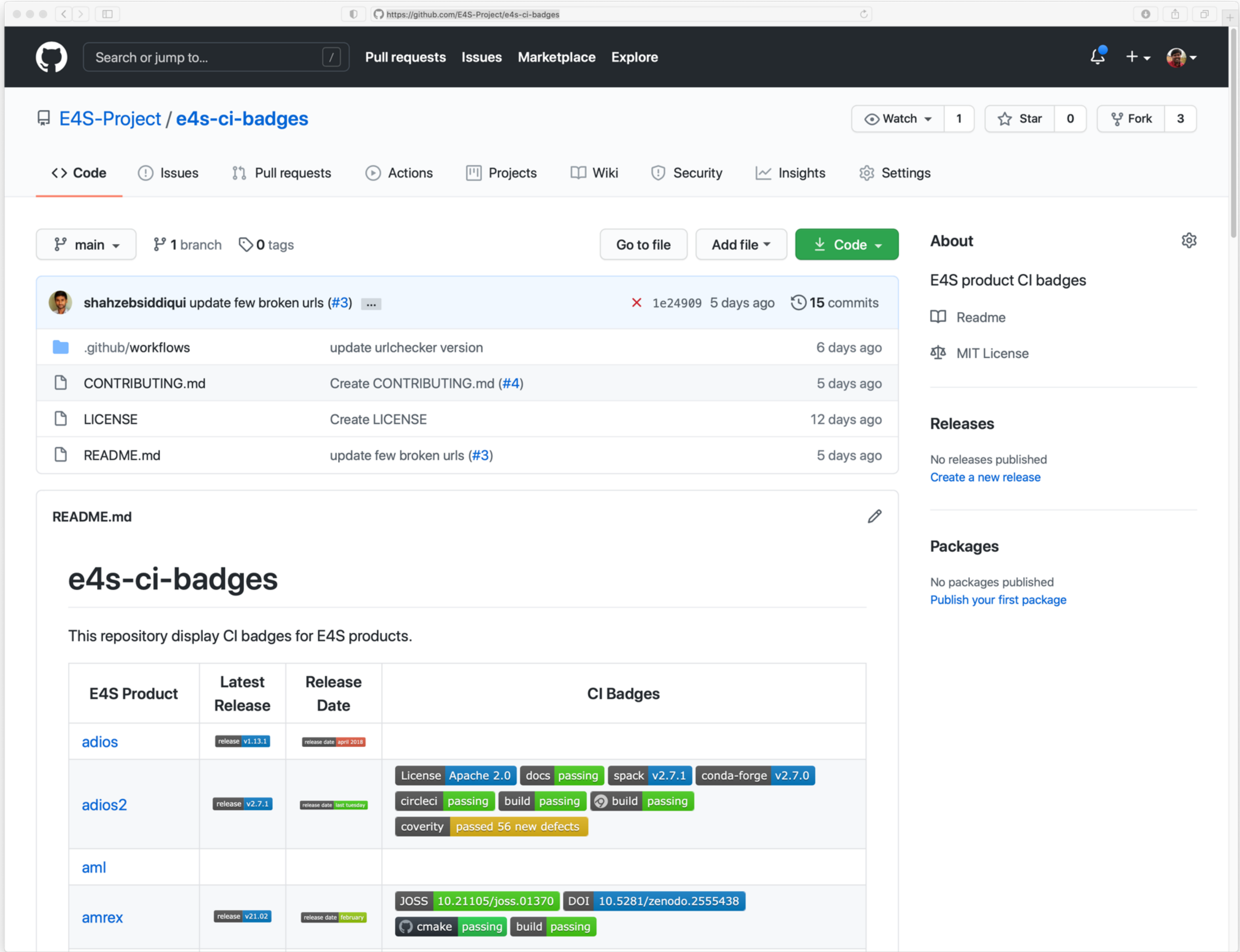
EXASCALE COMPUTING PROJECT

ORNL GitLab Build Pipeline for E4S Spack Build Cache



- ppc64le (Ascent @ ORNL)
- Reproducible container builds

E4S CI Badges



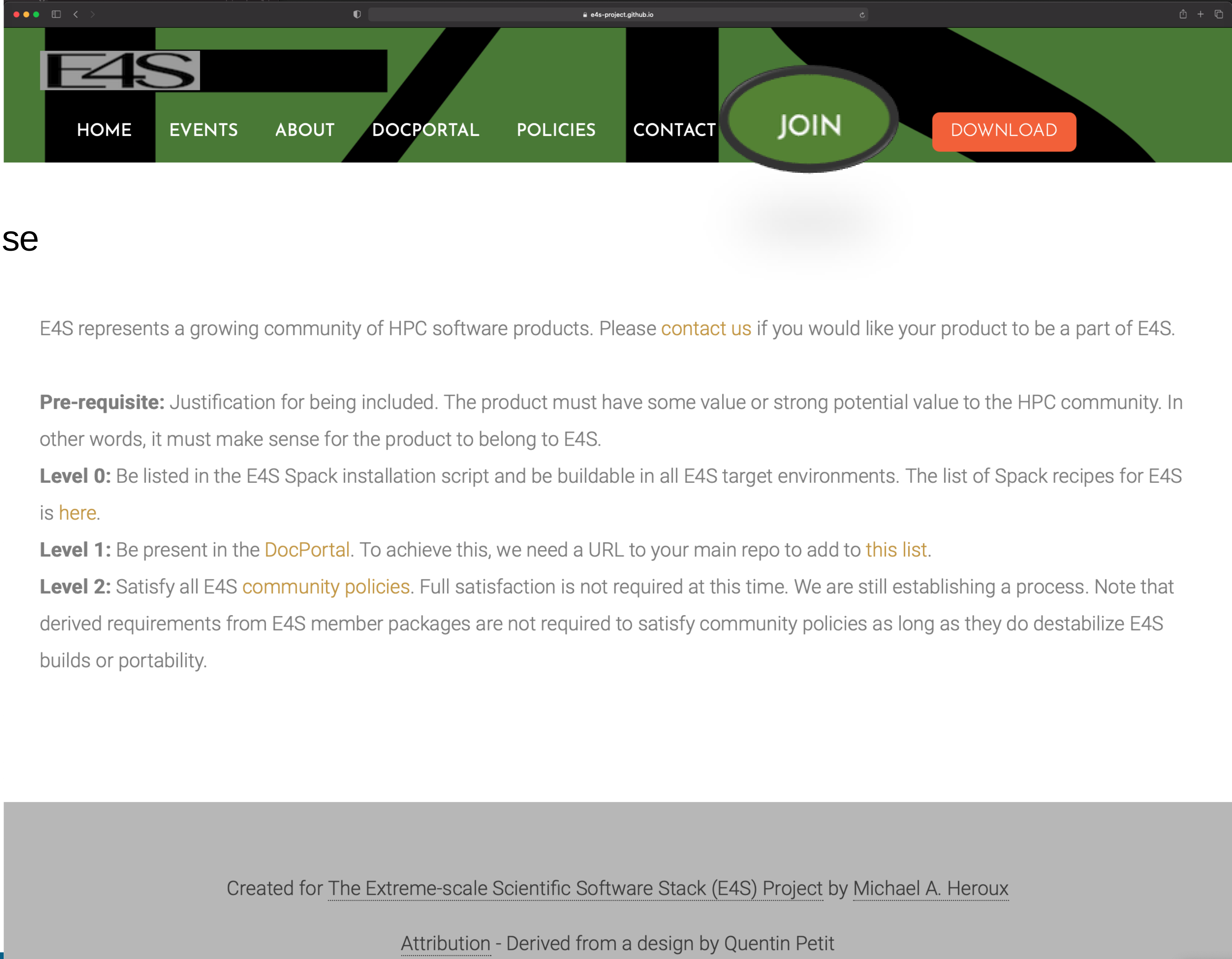
E4S Community Engagement

Opportunities via E4S

- E4S enables portfolio strategy for ASCR R&D software delivery:
  - Facilities: Robust planning, delivery, integration and testing at Facilities
  - Community: MPI Forum, C++, OpenMP, LLVM
  - Vendor: Coordinated integration into vendor software stacks
  - Users: Turnkey delivery of capabilities to DOE program offices, US agencies, industry, international partners
- E4S provides incentives and support for high-quality research software products
  - Community policies: Drives quality by explicit expectations and clear view of gaps
  - SDKs for community interaction: Build awareness and collaboration across independent teams
  - Transparency: E4S DocPortal, build, test, integration shows quality (good or poor) of a product
- E4S provides direct path for software teams to reach users and other stakeholders
  - Example: ArborX is brand new geometric search library
    - Part of E4S, available at DocPortal, tested regularly on many platforms
    - Installed anywhere E4S is installed, users can count on it being there
    - Without E4S: ArborX would take years to become visible and available
  - Availability and adoption timeline reduced from years (or never) to months

Joining E4S

- Process:
- Pre-req: Must make sense
- L0: E4S Spackified
- L1: Listed in DocPortal
- L2: Satisfy policies



ECP EXASCALE COMPUTING PROJECT

Broader Community Engagement

The Second Extreme-scale Scientific Software Stack Forum (E4S Forum)
September 24th, 2020, Workshop at EuroMPI/USA'20

- Presenters from 11 institutions, 6 non-DOE
- 70 participants
  - DOE Labs, NASA
  - AMD
  - HLRS, CSCS

- E4S: The Extreme-scale Scientific Software Stack for Collaborative Open Source Software, Michael Heroux, Sandia National Laboratories
- Title: Practical Performance Portability at CSCS, **Ben Cumming, CSCS**
- Title: An Overview of High Performance Computing and Computational Fluid Dynamics at NASA, **Eric Nielsen, NASA Langley**
- Towards An Integrated and Resource-Aware Software Stack for the EU Exascale Systems, **Martin Schulz, Technische Universität München**
- Spack and E4S, Todd Gamblin, LLNL
- Rocks and Hard Places – Deploying E4S at Supercomputing Facilities, Ryan Adamson, Oak Ridge Leadership Computing Facility
- Advances in, and Opportunities for, LLVM for Exascale, Hal Finkel, Argonne National Laboratory
- Kokkos: Building an Open Source Community, Christian Trott, SNL
- Experiences in Designing, Developing, Packaging, and Deploying the MVAPICH2 Libraries in Spack, **Hari Subramoni, Ohio State University**
- Software Needs for Frontera and the NSF Leadership Class Computing Facility – the Extreme Software Stack at the Texas Advanced Computing Center, **Dan Stanzione, TACC**
- Building an effective ecosystem of math libraries for exascale, Ulrike Yang
- Towards Containerized HPC Applications at Exascale, Andrew Younge, Sandia
- E4S Overview and Demo, Sameer Shende, University of Oregon
- The Supercomputer "Fugaku" and Software, programming models and tools, **Mitsuhisa Sato, RIKEN Center for Computational Science (R-CCS), Japan**

E4S provides a natural collaboration vehicle for interacting within DOE, with other US agencies, industry and international partners

E4S summary

What E4S is not	What E4S is
• A closed system taking contributions only from DOE software development teams.	• Extensible, open architecture software ecosystem accepting contributions from US and international teams. • Framework for collaborative open-source product integration for ECP & beyond, including AI and Quantum.
• A monolithic, take-it-or-leave-it software behemoth.	• Full collection if compatible software capabilities **and** • Manifest of a la carte selectable software capabilities.
• A commercial product.	• Vehicle for delivering high-quality reusable software products in collaboration with others. • New entity in the HPC ecosystem enabling first-of-a-kind relationships with Facilities, vendors, other DOE program offices, other agencies, industry & international partners.
• A simple packaging of existing software.	• Hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations. • Conduit for future leading edge HPC software targeting scalable computing platforms.

Looking Forward

Lessons learned from E4S/ECP ST to carry forward

- Deliver DOE reusable software as a portfolio
  - E4S value is already more than the sum of its parts
  - Community policies drive quality, membership
  - DocPortal, testing, containerization, cloud, build caches, modules, etc., greatly improve access & usability
  - Poor performing products are ID'ed, then improved or removed
- E4S is ready to extend to next-generation software and hardware needs
  - AI/ML products already in portfolio, ready for any new products
  - Quantum, FPGA, neuromorphic devices likely to be accelerators
    - From a macro software architecture, similar to GPUs
    - Software for these devices can and should be part of the same stack for holistic HPC environment
- DOE software as a portfolio is a first-class entity in the ecosystem
  - E4S planning, executing, tracking, assessing is peer collaboration with Facilities, program offices, vendors, etc
  - E4S can become a perennial asset for DOE/ASCR as part of its mission impact within and beyond DOE

E4S sustainability

Challenges

- ECP has a robust tailored 413.3b project management infrastructure
- Transitioning & adapting this infrastructure is essential for post-ECP success
- Funding models, portfolio management, org structure are particularly critical

Opportunities

- A sustainable software ecosystem for HPC software from DOE & broader community
- Payoff if done right: better, faster and cheaper – get all three

E4S Expansion – Base Scope & Gaps

Within base scope

Making a high-quality HPC product portfolio through tools, processes, and transparency

Community policies: Improve product quality upstream, shepherd membership growth

DocPortal: Provide easy access to product documentation

Portfolio testing: Protecte against regressions, prepare for new platforms

Curated collection: Maintain version compatibility across products

Turnkey stack via quarterly releases: Provide functionality via Spack, containers, clouds

Gaps not in base

Features that are a significant departure from core mission needs

Sustained support of new customers (without specific collaborative funding)

Activities related to commercial software enterprise

Ongoing support of a maintenance-only product (no longer funded for R&D)

Need: Business models for the gaps

Final points

- E4S is a curated software stack with quality improvement incentives, moving toward turnkey use
- With DOE program managers ECP is starting
  - Software ecosystem sustainability planning
  - E4S strategic plan (will include monthly townhalls)
- We believe
  - E4S has reduced important gaps that limit usefulness of DOE software for industry
  - But some gaps remain
- Next steps:
  - Better characterize these gaps
  - Explore models to further reduce and close gaps
  - Plan and execute toward sustainability

Some opportunities for interactions

- E4S is ready for app teams to use now
  - Curated, version-managed collection of many libraries & tools app teams use
  - Turnkey builds, containers & cloud builds, Spack build cache: Can dramatically improve productivity
  - Full E4S suite available for non-GPU platforms (CPU-based clusters)
  - Many E4S product work on Nvidia GPUs, growing set of capabilities for Intel, AMD GPUs, some Arm/SVE
- Would love to engage new software teams
- Another opportunity:
  - 2021 Collegeville Workshop on Scientific Software – Software Teams
  - https://collegeville.github.io/CW21/
- Thank you!

EXASCALE COMPUTING PROJECT

ST Capability Assessment Report (CAR)

- Tiered discussion of ECP Software Technology structure, strategy, status and plans
- From high-level overview to details about each team's activities and next steps
- Produced about twice a year
- Includes gap analyses
- E4S scope updated for emerging needs


EXASCALE COMPUTING PROJECT

ECP-RPT-ST-0002-2020–Public

ECP Software Technology Capability Assessment Report–Public

Michael A. Heroux, Director ECP ST
Lois Curfman McInnes, Deputy Director ECP ST
Rajeev Thakur, Programming Models & Runtimes Lead
Jeffrey S. Vetter, Development Tools Lead
Sherry Li, Mathematical Libraries Lead
James Ahrens, Data & Visualization Lead
Todd Munson, Software Ecosystem & Delivery Lead
Kathryn Mohror, NNSA ST Lead

November 19, 2020

https://www.exascaleproject.org/wp-content/uploads/2021/01/ECP-ST-CAR-v2.5.pdf

Thank you

https://www.exascaleproject.org

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.

EXASCALE COMPUTING PROJECT

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.