Toward Improved Sustainability of the Exascale Computing Project Software Stack

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

MAGIC Meeting, August 4, 2021

Sustainability Sketch

- A product is known to be sustainable only after it has been sustained
- But can determine attributes such as “eating broccoli” for health
- Reason to sustain (why) as important as cost/approach to sustain (how)

$$\text{Sustainability} \propto \text{why} \div \text{how}$$

ECP Software Stack:
Why sustain
- Products people use
- On emerging platforms

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

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

Key ECP Software Stack Legacy:
- Portable execution on:
  - CPUs
  - 3 different GPUs
- A bridge from CPUs to GPUs

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

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

How to sustain

- Software architecture
- Quality expectations
- Access & understanding
- Plan, execute, deliver
- Better, faster, cheaper: all three

The Extreme-Scale Scientific Software Stack (E4S) and Software Development Kits (SDKs)

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, 2020.10) - 67 full release products
- Feb 2021: E4S 21.02 - 67 full release, 4 partial release
- May 2021: E4S 21.05 – 87 full release products

https://e4s.io

Lead: Sameer Shende
(U Oregon)

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

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

} 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

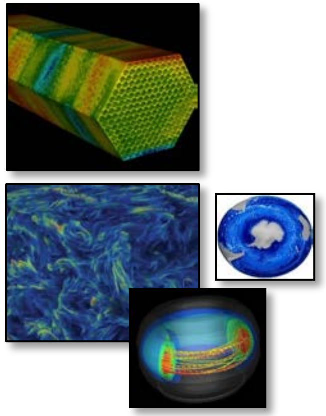
Next-generation algorithms

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

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 Community Policies

E4S Community Policies V1.0 Released

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.

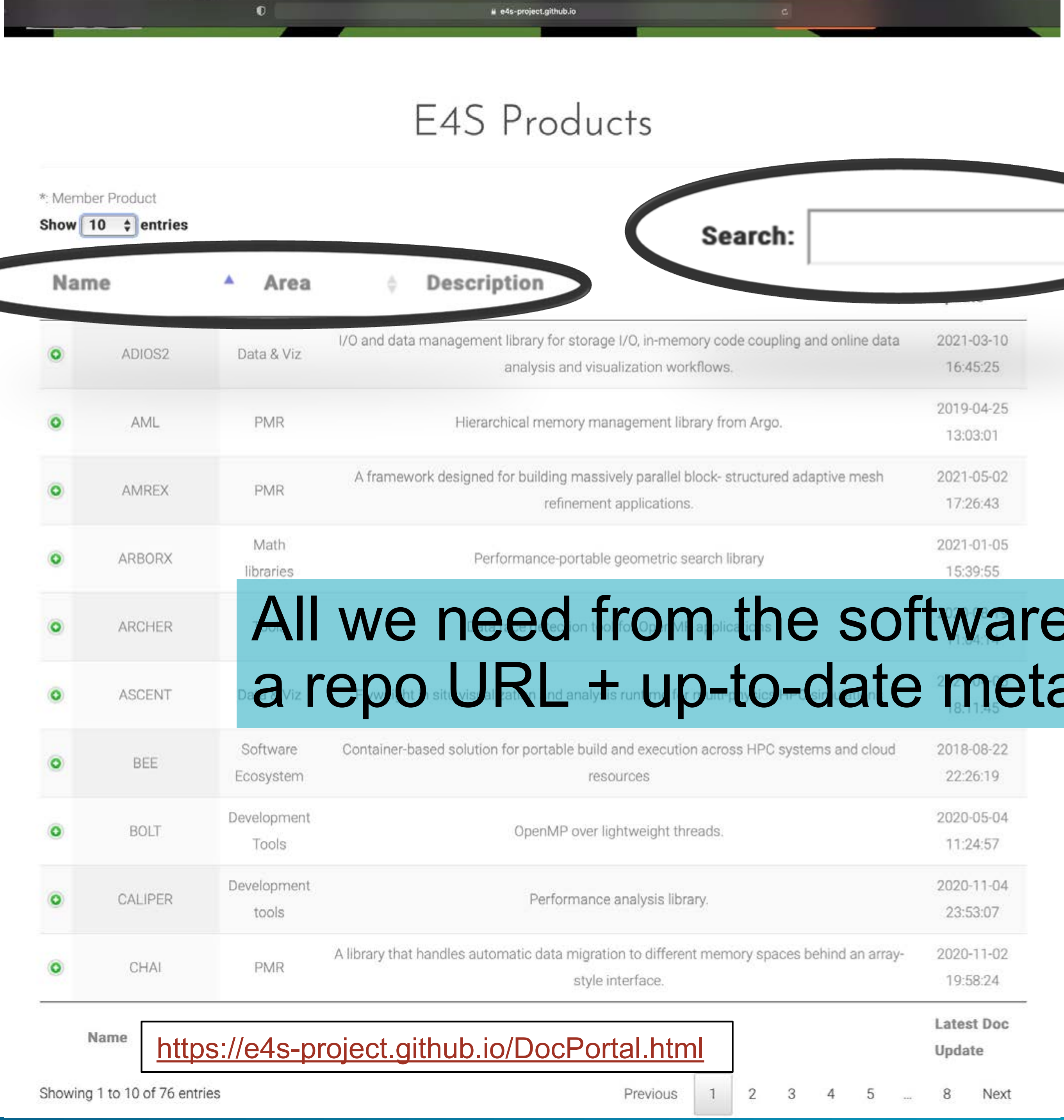
P8 *Error Handling* Each E4S member package will adopt and document a consistent system for signifying error conditions as appropriate for the language and application. For e.g., returning an error condition or throwing an exception. In the case of a command line tool, it should return a sensible exit 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.

E4S DocPortal

E4S DocPortal

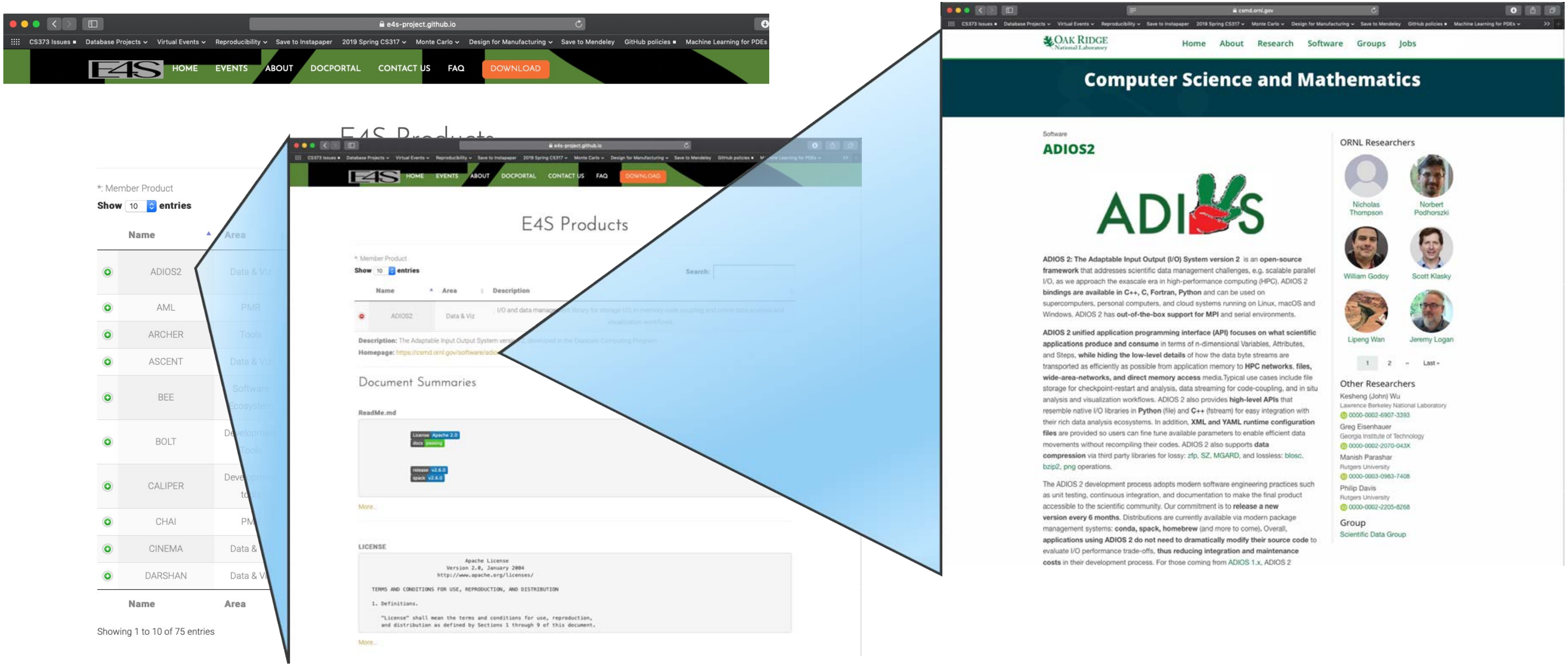
- Single point of access
- All E4S products
- Summary Info
  - Name
  - Functional Area
  - Description
  - License
- Searchable
- Sortable
- Rendered daily from repos



All we need from the software team is a repo URL + up-to-date meta-data files

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

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



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

E4S Planning, Executing, Delivering

ECP ST Planning Process: Hierarchical, three-phase, cyclical

Baseline

FY20–23 Baseline Plan
High level Definitions

- Q2 FY19 start
- FY20 Base plan
- FY21–23 planning packages

Annual Refinement

FY Refine Baseline Plan As Needed
Basic activity definitions

- 6 months prior to FY
- 4–6 P6 Activities/year
- Each activity:
  - % annual budget
  - Baseline start/end
  - High level description

Per Activity

Detailed Plan
Complete activity definitions

- 8 weeks prior to start
- High-fidelity description
- Execution strategy
- Completion criteria
- Personnel details

Two-level Review Process

Changes to Cost, Scope, and Schedule

Minor	Major
Lightweight Review in Jira, L3 and L2 leads	Change Control Board Review, ECP leadership

Variance Recorded in Jira

Proceed with Execution

KPP-3: Focus on capability integration

- **Capability:** Any significant product functionality, including existing features adapted to the pre-exascale and exascale environments, that can be integrated into a client environment.
- **Capability Integration:** Complete, sustainable integration of a significant product capability into a client environment in a pre-exascale environment (tentative score) and in an exascale environment (confirmed score).

ECP ST Lifecycle summary

Managed by P6 Activity Process

Create Annual Planning Package
- Each product has its own planning packages
- Defined for all FYs

Refine upcoming FY plan
- Complete 6 months prior to FY
- 4 or more P6 activities per product

Refine upcoming P6 activity
- Complete 8 weeks prior to activity start
- Include all details

Develop capabilities and track progress via tailored EVM

Measured by KPP-3 Process

Integrate into product
- Full testing, documentation, etc.
- Direct access for some users

Integrate into SDK
- Satisfy SDK community policies
- Direct access for some users

Integrate into E4S
- Satisfy E4S community policies
- Full ecosystem with high value

Deliver to users
- From source (spack)
- Containers, cloud

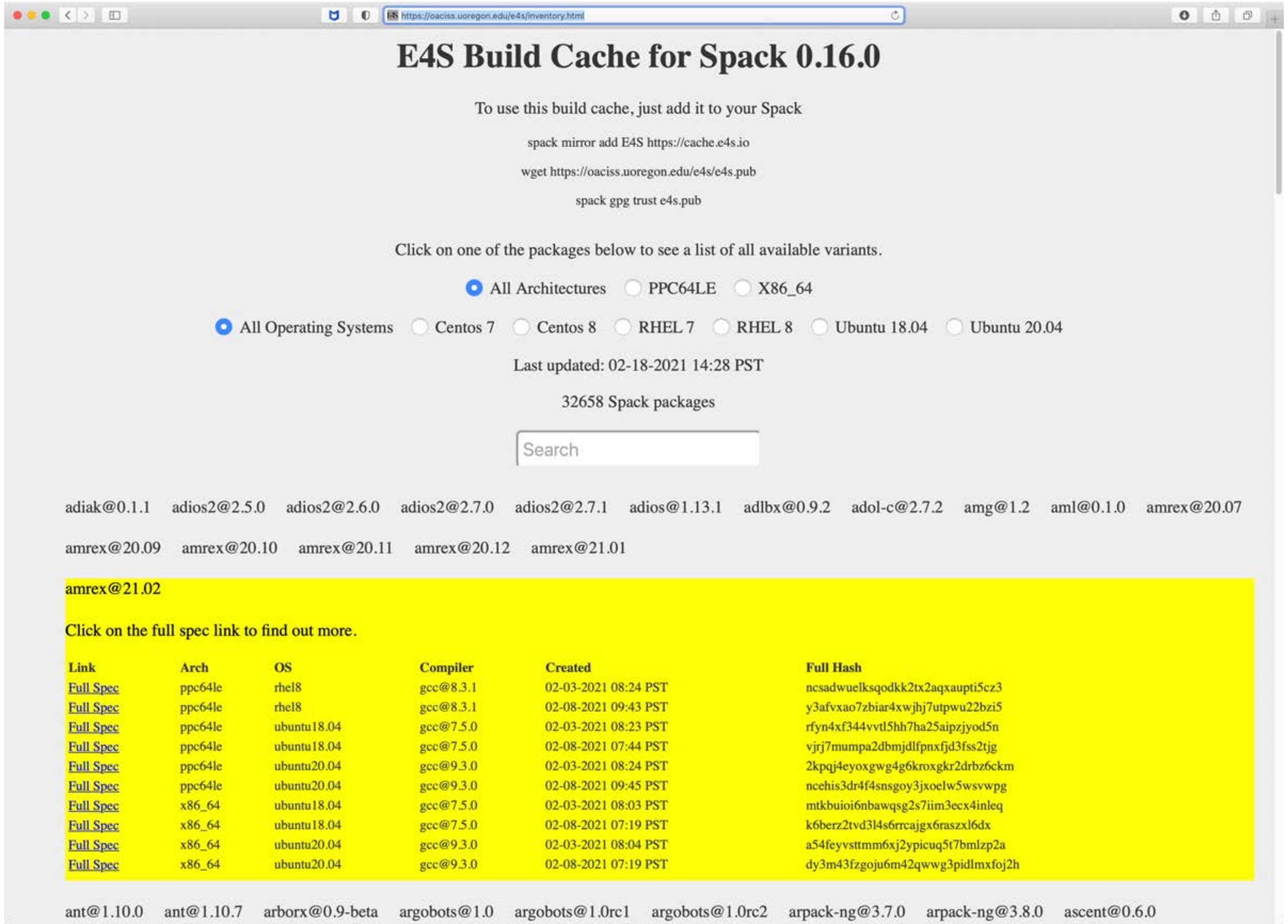
EXASCALE COMPUTING PROJECT

Using E4S

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.

E4S: Spack Build Cache



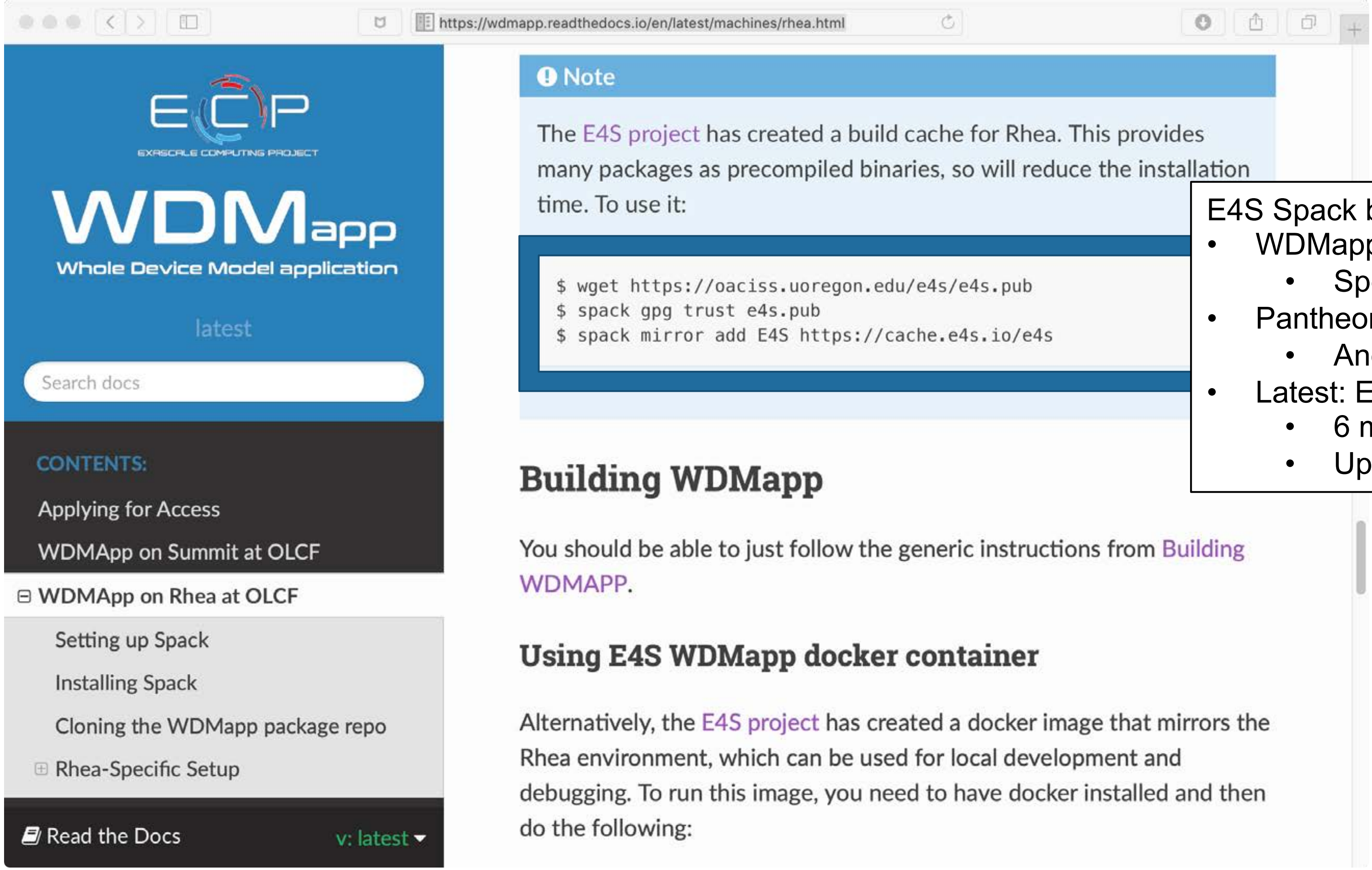
Link	Arch	OS	Compiler	Created	Full Hash
Full Spec	ppc64le	rhel8	gcc@8.3.1	02-03-2021 08:24 PST	ncsadwuelksqodkk2tx2aqxaupti5cz3
Full Spec	ppc64le	rhel8	gcc@8.3.1	02-08-2021 09:43 PST	y3afvxao7zbiar4xwjhj7utpwu22bzi5
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	02-03-2021 08:23 PST	rfyn4xf344vvtl5hh7ha25aipzjyod5n
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	02-08-2021 07:44 PST	vjrj7mumpa2dbmjdlfpnxfjd3fss2tjg
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	02-03-2021 08:24 PST	2kpqj4eyoxgwg4g6kroxgkr2drbz6ckm
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	02-08-2021 09:45 PST	ncehis3dr4f4snsgoy3jxoelw5wsvwpg
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	02-03-2021 08:03 PST	mtkbuioi6nbawqsg2s7iim3ecx4inleq
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	02-08-2021 07:19 PST	k6berz2tvd3l4s6rrcajgx6raszxl6dx
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	02-03-2021 08:04 PST	a54feyvsttmm6xj2ypicuq5t7bmlzp2a
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	02-08-2021 07:19 PST	dy3m43fzgoju6m42qwwg3pidlmxfoj2h

- 32,000+ binaries
- S3 mirror
- No need to build from source code!
- Speeds up installations 10x

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

EXASCALE COMPUTING PROJECT

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



E4S Spack build cache:

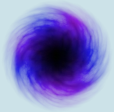
- WDMapp added E4S mirror
  - Speedup: 10X
- Pantheon: 10X
  - Another 10X via "smoother" installs
- Latest: ExaWind (Nalu-Wind)
  - 6 minutes with build cache
  - Up to 4 hours without

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

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

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

Community Policies
Commitment to software quality



DocPortal
Single portal to all E4S product info

Portfolio testing
Especially leadership platforms

Curated collection
The end of dependency 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

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.

Backup Content

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

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.

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov