How a Community Software Ecosystem Perspective Helps to Advance Science Goals in the Exascale Computing Project

Lois Curfman McInnes, Argonne National Laboratory
in collaboration with the ECP community

12th Joint Laboratory for Extreme Scale Computing (JLESC) Workshop
Online: http://icl.utk.edu/jlesc12
February 26, 2021
Thank you

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

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

https://www.exascaleproject.org
Ecosystem: A group of independent but interrelated elements comprising a unified whole

Diversity is essential for an ecosystem to thrive.

- No element functions in isolation.
- Each element fulfills unique roles.
- The whole is greater than the sum of its parts.
Software is the foundation of sustained CSE collaboration and scientific progress.

CSE = Computational Science & Engineering

Ref: Research and Education in Computational Science and Engineering, U. Rüde, K. Willcox, L.C. McInnes, H. De Sterck, SIAM Review, 2018
We must explicitly consider community software ecosystem perspectives for next-generation computational science

• Complex, intertwined challenges
  – Technical and sociological

• Need community efforts
  – Improve software sustainability
  – Change research culture
  – Promote collaboration

• Emphasis for this presentation
  – ECP software ecosystem work
  – And community opportunities

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

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.

https://dx.doi.org/10.1038/s43588-021-00033-y
ECP’s holistic approach uses co-design and integration to achieve exascale computing

**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

---

Performant mission and science applications at scale
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

**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 multiphysics, 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

**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

**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
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:

<table>
<thead>
<tr>
<th>ExaWind: Turbine Wind Plant Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harden wind plant design and layout against energy loss susceptibility; higher penetration of wind energy</td>
</tr>
<tr>
<td>Lead: NREL DOE EERE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subsurface: Carbon Capture, Fossil Fuel Extraction, Waste Disposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliably guide safe long-term consequential decisions about storage, sequestration, and exploration</td>
</tr>
<tr>
<td>Lead: LBNL DOE BES EERE, FE, NE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExaSky: Cosmological Probe of the Standard Model of Particle Physics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unravel key unknowns in the dynamics of the Universe: dark energy, dark matter, and inflation</td>
</tr>
<tr>
<td>Lead: ANL DOE HEP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WDMApp: High-Fidelity Whole Device Modeling of Magnetically Confined Fusion Plasmas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare for ITER experiments and increase ROI of validation data and understanding; prepare for beyond-ITER devices</td>
</tr>
<tr>
<td>Lead: PPPL DOE FES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The MARBL Multi-physics Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-physics simulations of high energy-density physics and focused experiments driven by high-explosive, magnetic or laser based energy sources</td>
</tr>
<tr>
<td>• Magneto-radiation-hydrodynamics at the exascale</td>
</tr>
<tr>
<td>• Next-generation pulsed power / ICF modeling</td>
</tr>
<tr>
<td>• High-order numerical methods</td>
</tr>
<tr>
<td>Lead: LLNL</td>
</tr>
</tbody>
</table>
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

Selected ECP Software Technologies

Programming Models and Runtimes: MPI Kokkos RAJA CHAI Umpire ... and more

Tools and Technology: TAU HPCToolkit Flux Caliper PAPI ... and more

Math Libraries (xSDK): hypre PETSc/TAO SuperLU Trilinos SUNDIALS ArborX FFT BLAS, LAPACK MFEM ... and more

Visualization Analysis and Reduction: ALPINE Cinema STRUMPACK VTK-m SZ zip SPOT ... and more

Data Mgmt, I/O, Checkpoint Restart: SCR MPI-IO HDF5 NetCDF ADIOS UnityFS VeloC ... and more

Ecosystem: E4S at large: Spack ... and more

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.

See E4S.io for more ST products

AID AML BEE Darshan DTK Dyninst FleCSI ForTriliinios GASNet Ginkgo Kokkoskernels Legion libEnsemble MarFS NRM OpenACC Papyrus PaRSEC PDT PowerStack ScALAPACK SCR SICM SLATE SWIG Tasmanian Umap UPC++
xSDK: Primary delivery mechanism for ECP math libraries’ continual advancements toward predictive science

As motivated and validated by the needs of ECP applications:

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

Next-generation algorithms

- Toward predictive scientific simulations
- Advances in data structures for new node architectures
- Improving library quality, sustainability, interoperability
- Increasing performance, portability, productivity

Timeline:

- xSDK release 0.6.0 (Nov 2020)
- xSDK release 1
- xSDK release 2
- xSDK release n

Delivering an open, hierarchical software ecosystem

Levels of Integration | Product | Source and Delivery
--- | --- | ---
• Build all SDKs | E4S | Source: ECP E4S team; Non-ECP Products (all dependencies)
  • Build complete stack
  • Containerize binaries
  Delivery: spack install e4s; containers; CI Testing

• Group similar products | SDKs | Source: SDK teams; Non-ECP teams (policy compliant, spackified)
  • Make interoperable
  • Assure policy compliant
  • Include external products
  Delivery: Apps directly; spack install sdk; future: vendor/facility

• Standard workflow | ST Products | Source: ECP L4 teams; Non-ECP Developers; Standards Groups
  • Existed before ECP
  Delivery: Apps directly; spack; vendor stack; facility stack
Extreme-scale Scientific Software Stack (E4S)

- **E4S**: HPC Linux Ecosystem – a software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source, containers, 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

Lead: Sameer Shende (U Oregon)
**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
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: Spark-based Build and Installation Each E4S member package supports a scriptable Spark 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 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

- The DocPortal is live!
- Summary Info
  - Name
  - Functional Area
  - Description
  - License
- Searchable
- Sortable

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 Spack build cache: 10x acceleration of re-builds

- 27,000+ binaries
- No need to build from source code!

https://oaciss.uoregon.edu/e4s/inventory.html
WDMApp: Speeding up bare-metal installs using E4S build cache

The E4S project has created a build cache for Rhea. This provides many packages as precompiled binaries, so will reduce the installation time. To use it:

```bash
$ wget https://oaciss.uoregon.edu/e4s/e4s.pub
$ spack gpg trust e4s.pub
$ spack mirror add E4S https://cache.e4s.io/e4s
```

Building WDMapp

You should be able to just follow the generic instructions from Building WDMAPP.

Using E4S WDMapp docker container

Alternatively, the E4S project has created a docker image that mirrors the Rhea environment, which can be used for local development and debugging. To run this image, you need to have docker installed and then do the following:


Special Thanks to Sameer Shende, WDMapp Team

- E4S Spack build cache
- Adding E4S mirror
- WDMApp install speeds up!
# E4S summary

## What E4S is not

- A closed system taking contributions only from DOE software development teams.
- A monolithic, take-it-or-leave-it software behemoth.
- A commercial product.
- A simple packaging of existing software.

## What E4S is

- Extensible, open architecture software ecosystem accepting contributions from US and international teams.
- Framework for collaborative open-source product integration.
- A full collection if compatible software capabilities **and**
- A manifest of a la carte selectable software capabilities.
- Vehicle for delivering high-quality reusable software products in collaboration with others.
- The conduit for future leading edge HPC software targeting scalable next-generation computing platforms.
- A hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.
Recent advances in ECP software technologies as driven by needs of ECP apps

**Scalable Solvers**

**Speeding sparse algorithms on CPUs and GPUs**

- The STRUMPACK team has developed new capabilities for multifrontal rank-structured preconditioning.
- **Impact:** STRUMPACK provides robust and scalable factorization-based methods for ill-conditioned and indefinite systems that arise in multiscale, multiphysics simulations.

**Lossy Compression**

**Optimizing lossy compression methods to manage data volumes**

- The VeloC-SZ team has optimized SZ, an error-bounded prediction-based lossy compression model.
- **Impact:** SZ reduces dataset size while meeting users’ speed and accuracy needs by storing the most pertinent data during simulation and experiments.
- **More info:** Significantly Improving Lossy Compression for HPC Datasets with Second-Order Prediction and Parameter Optimization, HPDC20, K. Zhao et al.

**Performance Monitoring**

**Advancing performance counter monitoring capabilities for new ECP hardware**

- The Exa-PAPI team provides a consistent interface and methodology for the use of low-level performance counter hardware found across the entire system (CPUs, GPUs, on/off-chip memory, interconnects, I/O system, energy/power, etc).
- **Impact:** Exa-PAPI enables users to see, in near real time, relations between software performance and hardware events.
- **More info:** [https://icl.utk.edu/exa-papi](https://icl.utk.edu/exa-papi)

---

A few examples, among many more … all software available via [https://E4S.io](https://E4S.io)
More info about the impact of ECP software technologies

**ECP News**

- ECP-funded team investigates NVM techniques to improve data storage and performance speed
- ECP-funded researchers enable faster time-to-science with novel I/O processing method
- ECP project optimizes lossy compression methods to manage big science data volumes
- ALPINE project tests novel algorithm for in situ exascale data analysis
- Workflow technologies impact SC20 Gordon Bell COVID-19 award winner and two of the three finalists
- The Extreme-Scale Scientific Software Stack (E4S): A new resource for computational and data science research

**Technical Highlights**

- ECP Provides TAU, a CPU/GPU/MPI Profiler, for All HPC and Exascale Machines
- HeFFTe – a widely applicable, CPU/GPU, scalable multidimensional FFT that can even support exascale supercomputers
- SOLLVE: OpenMP for HPC and Exascale
- ECP releases of the tested and verified MAGMA Numerical Linear Algebra Library provide a wealth of cross-platform capabilities for exascale supercomputing
- STRUMPACK Speeds Sparse Algorithms on CPUs and GPUs
- STRUMPACK for GPU Developers – A Portable Sparse Solver Library That Runs Well on GPUs AND CPUs

**Let’s Talk Exascale Podcast:**

- Supporting Scientific Discovery and Data Analysis in the Exascale Era
- ECP Leadership Discusses Project Highlights, Challenges, and the Expected Impact of Exascale Computing
- Flexible Package Manager Automates the Deployment of Software on Supercomputers
- Accelerating the Adoption of Container Technologies for Exascale Computing
- Collaborative Community Impacts High-Performance Computing Programming Environments
- Simplifying the Deployment of High-Performance Computing Tools and Libraries
- Method Enables Collaborative Software Teams to Enhance Effectiveness and Efficiency
- Tackling the Complex Task of Software Deployment and Continuous Integration at Facilities
- Optimizing Math Libraries to Prepare Applications for Exascale Computing
ST Capability Assessment Report (CAR)

• Products discussed here are presented with more detail and further citations.

• We classify ECP ST product deployment as broad, moderate, or experimental.
  – Broad and moderate deployment is typically suitable for collaboration.
  – Web links are available for almost all products.
  – 67 of 71 of ECP ST products are available as part of the Extreme-scale Scientific Software Stack (E4S) http://e4s.io.

Community software ecosystems require high-quality software

- Complex, intertwined challenges
- Resources and opportunities to get involved
  - Software Sustainability Institute: http://www.software.ac.uk
  - US Research Software Sustainability Institute: http://urssi.us/
  - WSSSPE: http://wssspe.researchcomputing.org.uk
  - NUMFocus: https://www.numfocus.org
  - Software Carpentry: http://software-carpentry.org
  - Research Software Engineering (RSE) movement: https://bssw.io/items/an-introduction-to-national-rse-organizations
  - And more …


IDEAS-ECP team serve as catalysts – partnering 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)
What is BSSw?

Community-based hub for sharing information on practices, techniques, and tools to improve developer productivity and software sustainability for computational science.

We want and need contributions from the community … Join us!

- Types of content
  - Informative articles
  - Curated links
    - Highlight other web-based content
  - Events
  - WhatIs, HowTo docs
  - Blog articles

Recent articles

- Performance Portability and the ECP Project, A. Dubey
- Recent Successes with PSIP on HDF5, M. Miller, E. Pourmal & E. Gonsiorowski
- Testing Non-Deterministic Research Software, N. Eisty,
- What Does This Line Do? The Challenge of Writing a Well-Documented Code, M. Stoyanov

https://bssw.io
Upcoming community events … Join us!

- **Software-related events at SIAM CSE21** (March 1-5, 2021)

- **ECP panel series: Strategies for Working Remotely** (ongoing)
  - March 25: How Does Remote Work Impact Creativity and Innovation?

- **Webinar series: Best Practices for HPC Software Developers** (ongoing)

- **ECP training events** (ongoing)

- **2021 ECP Community BOF Days** (March 30 – April 1)
  - March 30, 11:00 am - The Tiered Testing Approach to Software Quality Assurance at Exascale and Beyond
  - March 30, 11:00 am - Exascale Computing Project Data Management, Data Analytics and Visualization Overview
  - March 30, 1:00 pm - Observing GPU performance using the TAU Performance System
  - March 30, 1:00 pm - Open MPI State of the Union
  - March 30, 3:00 pm - HDF5 Community BOF
  - March 31, 11:00 am - Updates and Roadmap for the PMIx Community
  - March 31, 11:00 am - Enabling Developer Productivity with Software Process Improvement
  - March 31, 1:00 pm - MPICH for Exascale: Supporting MPI-4 and ECP
  - March 31, 1:00 pm - Cultivating Software Sustainability, Productivity and Quality through BSSw.io
  - March 31, 3:00 pm - ECP Math Libraries Capabilities and Applications Engagement
  - April 1, 11:00 am - The LLVM Compiler Infrastructure --- Clang, Flang, OpenMP, and More
  - April 1, 1:00 pm - How to Measure and Analyze the Performance of GPU-accelerated Code
  - April 1, 3:00 pm - Tools for Data-driven Analysis and Improvement of HPC Scientific Software Development
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.

https://exascaleproject.org/better-scientific-productivity-through-better-scientific-software-the-ideas-report
We must explicitly consider **community software ecosystem perspectives** for next-generation computational science.

**Historically:** Organic growth of software ecosystems around packages

**What’s new now?** Bigger challenges, advances in technologies

**Let’s be intentional.**
- broader perspectives
- productivity, sustainability

Better science, Broader impact

**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

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.

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

Teams in the U.S. Exascale Computing Project (ECP) are working toward scientific advances on forthcoming exascale platforms, across a diverse suite of applications in chemistry, materials, energy, Earth and space science, data analytics, optimization, artificial intelligence, and national security. In turn, these applications build on software components, including programming models and runtimes, mathematical libraries, data and visualization packages, and development tools that comprise the Extreme-scale Scientific Software Stack (E4S). E4S represents a portfolio-driven effort to collect, test, and deliver the latest in reusable open-source HPC software products, as driven by the common needs of applications. E4S establishes product quality expectations and provides a portal as a starting point for access to product documentation. This presentation will discuss early experiences with how this software ecosystem approach delivers the latest advances from ECP software technology projects to applications, thereby helping to overcome software collaboration challenges across distributed aggregate teams. A key lesson learned is the need for close collaboration between teams developing applications and reusable software technologies, as well as the need for crosscutting strategies to increase developer productivity and software sustainability, thereby mitigating technical risks by building a firmer foundation for reproducible, sustainable science.