A cast of thousands: How the IDEAS Productivity project has advanced software productivity and sustainability

David E. Bernholdt (he/him)
Oak Ridge National Laboratory

On behalf of the entire IDEAS-ECP team

a Best Practices for HPC Software Developers webinar

learn more about IDEAS at https://ideas-productivity.org
and https://doi.org/10.48550/arXiv.2311.02010

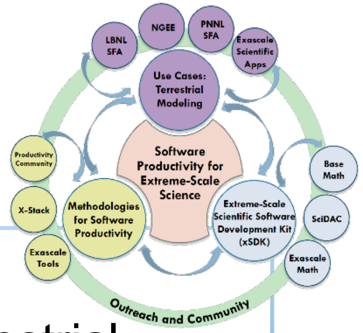
This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration

A Brief History of IDEAS

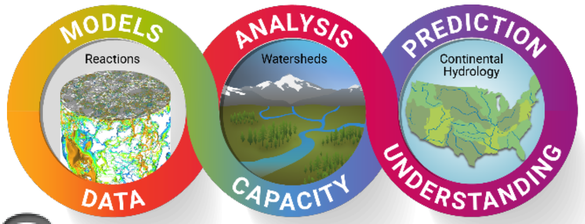
- IDEAS = Interoperable Design of Extreme-Scale Application Software
- First of its kind (in U.S.) with a focus on incubating, curating, and disseminating knowledge and methodologies about the sustainment of scientific software
  - Inspired by UK Software Sustainability Institute
- IDEAS is now a family of related projects
  - Different sponsors, different time frames, different people (but significant overlap), different approaches
  - **Common focus on improving developer productivity and software sustainability and trustworthiness**

Acronyms

DOE	= U.S. Department of Energy
ASCR	= Office of Advanced Scientific Computing Research
BER	= Office of Biological and Environmental Research
ECP	= Exascale Computing Project


LBNL SFA
NGEE
PNNL SFA
Exascale Scientific Apps
Use Cases: Terrestrial Modeling
Productivity Community
Software Productivity for Extreme-Scale Science
Base Math
X-Stack
Methodologies for Software Productivity
Extreme-Scale Scientific Software Development Kit (xSDK)
SciDAC
Exascale Tools
Exascale Math
Outreach and Community

- IDEAS-Classic (2014—2017)
  - Focus: multiscale multiphysics terrestrial ecosystem modeling
  - Sponsors: DOE/ASCR and BER
- IDEAS-ECP (2017—2023)
  - Focus: supporting the ecosystem of applications, libraries, and tools developed by ECP
  - Sponsor: DOE/ECP
- IDEAS-Watersheds (2019—present)
  - Focus: accelerating watershed science through a community driven software ecosystem
  - Sponsor: DOE/BER



IDEAS productivity

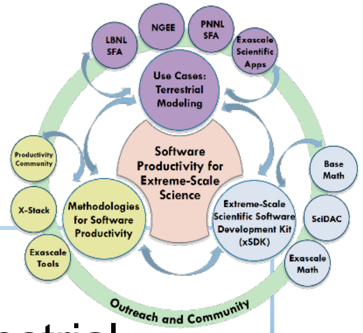
ECP EXASCALE COMPUTING PROJECT

A Brief History of IDEAS

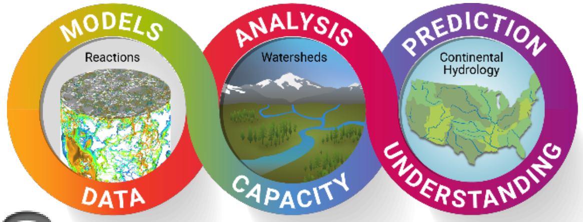
- IDEAS = Interoperable Design of Extreme-Scale Application Software
- First of its kind (in U.S.) with a focus on incubating, curating, and disseminating knowledge and methodologies about the sustainment of scientific software
  - Inspired by UK Software Sustainability Institute
- IDEAS is now a family of related projects
  - Different sponsors, different time frames, different people (but significant overlap), different approaches
  - **Common focus on improving developer productivity and software sustainability and trustworthiness**

Acronyms

DOE	= U.S. Department of Energy
ASCR	= Office of Advanced Scientific Computing Research
BER	= Office of Biological and Environmental Research
ECP	= Exascale Computing Project


LBNL SFA
NGEE
PNNL SFA
Exascale Scientific Apps
Use Cases: Terrestrial Modeling
Productivity Community
Software Productivity for Extreme-Scale Science
Base Math
X-Stack
Methodologies for Software Productivity
Extreme-Scale Scientific Software Development Kit (xSDK)
SciDAC
Exascale Tools
Exascale Math
Outreach and Community

- IDEAS-Classic (2014—2017)
  - Focus: multiscale multiphysics terrestrial ecosystem modeling
  - Sponsors: DOE/ASCR and BER
- IDEAS-ECP (2017—2023)
  - Focus: supporting the ecosystem of applications, libraries, and tools developed by ECP
  - Sponsor: DOE/ECP
- IDEAS-Watersheds (2019—present)
  - Focus: accelerating watershed science through a community driven software ecosystem
  - Sponsor: DOE/BER

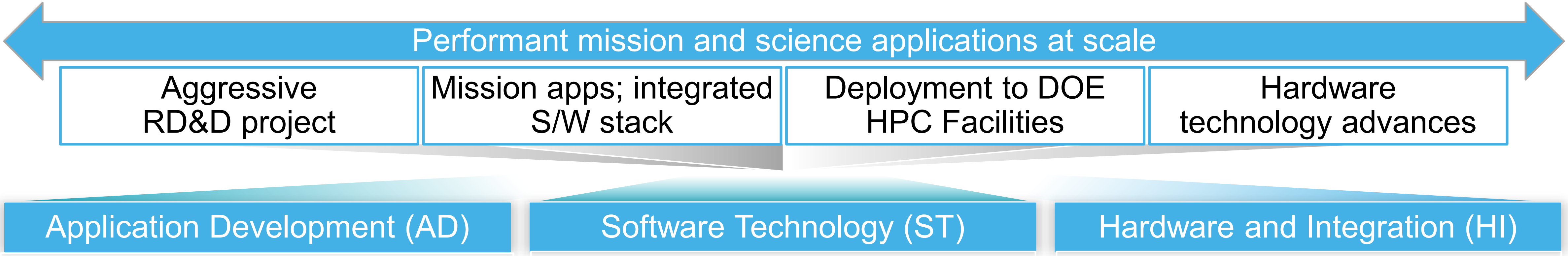


IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

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

80+ R&D teams, 1000 researchers!


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)
Software Technology (ST)
Hardware and Integration (HI)

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
ML, graph analytics, mesh refinement, PDE discretization, particles, online data analytics

Andrew Siegel, AD Director
Erik Draeger, AD Deputy Director

Software Technology (ST)

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

70 unique software products developed by 35 teams spanning programming models and runtimes, math libraries, data and visualization, development tools

Mike Heroux, ST Director
Lois Curfman McInnes, ST Deputy Director

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

Richard Gerber, HI Director
Susan Coghlan, HI Deputy Director

Science and beyond: Applications and discovery in ECP

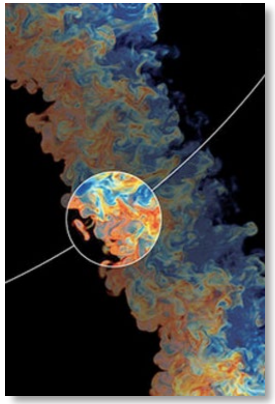
24 applications
+ 6 co-design projects
62 separate codes!

National security

Next-generation, **stockpile stewardship** codes

Reentry-vehicle-environment simulation

Multi-physics science simulations **of high-energy density physics** conditions



Energy security

Turbine **wind plant** efficiency

Design and commercialization of **SMR**s

Nuclear fission and fusion reactor **materials design**

Subsurface use for **carbon capture**, petroleum extraction, waste disposal

High-efficiency, low-emission **combustion engine** and gas turbine design

Scale up of **clean fossil fuel** combustion

Biofuel catalyst design

Economic security

Additive manufacturing of qualifiable metal parts

Reliable and efficient planning of the **power grid**

Seismic hazard risk assessment

Scientific discovery

Cosmological probe of the standard model of particle physics

Validate fundamental laws of nature

Plasma wakefield accelerator design

Light source-enabled **analysis of protein and molecular structure** and design

Find, predict, and control materials and properties

Predict and control **magnetically confined fusion plasmas**

Demystify **origin of chemical elements**

Earth systems

Accurate regional impact assessments in **Earth system models**

Stress-resistant crop analysis and catalytic conversion of **biomass-derived alcohols**

Metagenomics for analysis of biogeochemical cycles, climate change, environmental remediation

Health care

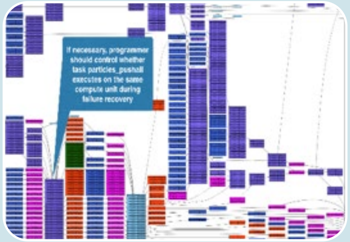
Accelerate and translate **cancer research** (partnership with NIH)

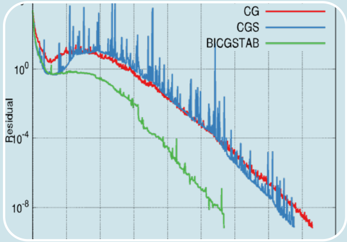
Thank you to Andrew Siegel, Erik Draeger and ECP applications teams

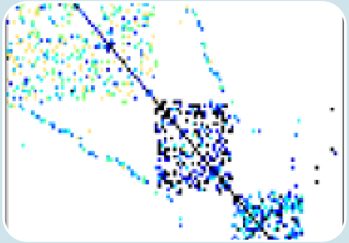
ECP ST has six technical areas

35 projects
70 software products

ECP ST Director: Mike Heroux
ECP ST Deputy Director: L.C. McInnes


If necessary, programmer should control whether task particles_pushall executes on the same compute unit during failure recovery





Programming Models & Runtimes

- Enhance and get ready for exascale the MPI and OpenMP programming models (hybrid programming models, deep memory copies)
- Develop 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

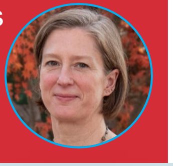
- I/O via the HDF5 API
- Insightful, memory-efficient in-situ visualization and analysis
- Data reduction via scientific data compression
- Checkpoint restart

Software Ecosystem

- Develop features in Spack necessary to support ST products in E4S, and the AD projects that adopt it
- Develop Spack stacks for reproducible turnkey software deployment
- Optimization and interoperability of containers for HPC
- 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

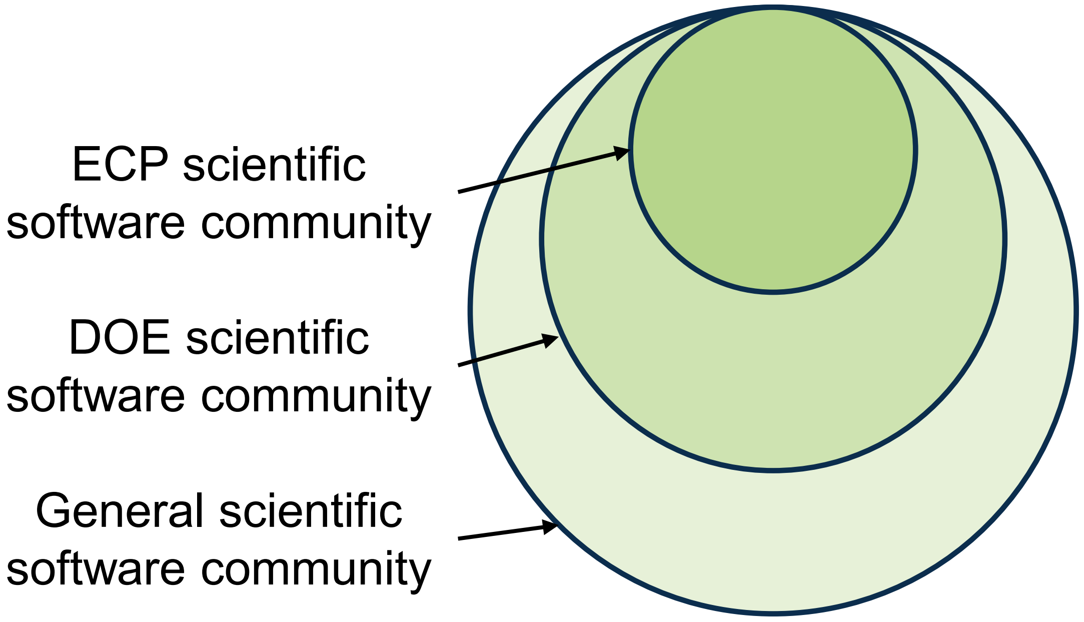


Area Leads:	Rajeev Thakur	Jeff Vetter	Sherry Li	Jim Ahrens	Todd Munson	Kathryn Mohror

IDEAS-ECP Had to Be Different

- The ECP environment and needs were very different from IDEAS-Classic
  - Stringent deliverable set for performance and capability
  - Requirement to utilize new exascale hardware platforms
    - Actual exascale hardware available only late in the project
    - Strong likelihood of encountering bugs or inadequacies in developing software environment
    - Extensive exploration and experimentation with code
  - Impossible to do 1:1 interactions with every ECP software team
    - 80+ software teams, 100+ software products, ~1000 people
- So, we had to develop a different approach for IDEAS-ECP, focusing on…
  - Fostering software communities,
  - Broadly applicable methodologies and resources, and
  - Disseminating knowledge broadly

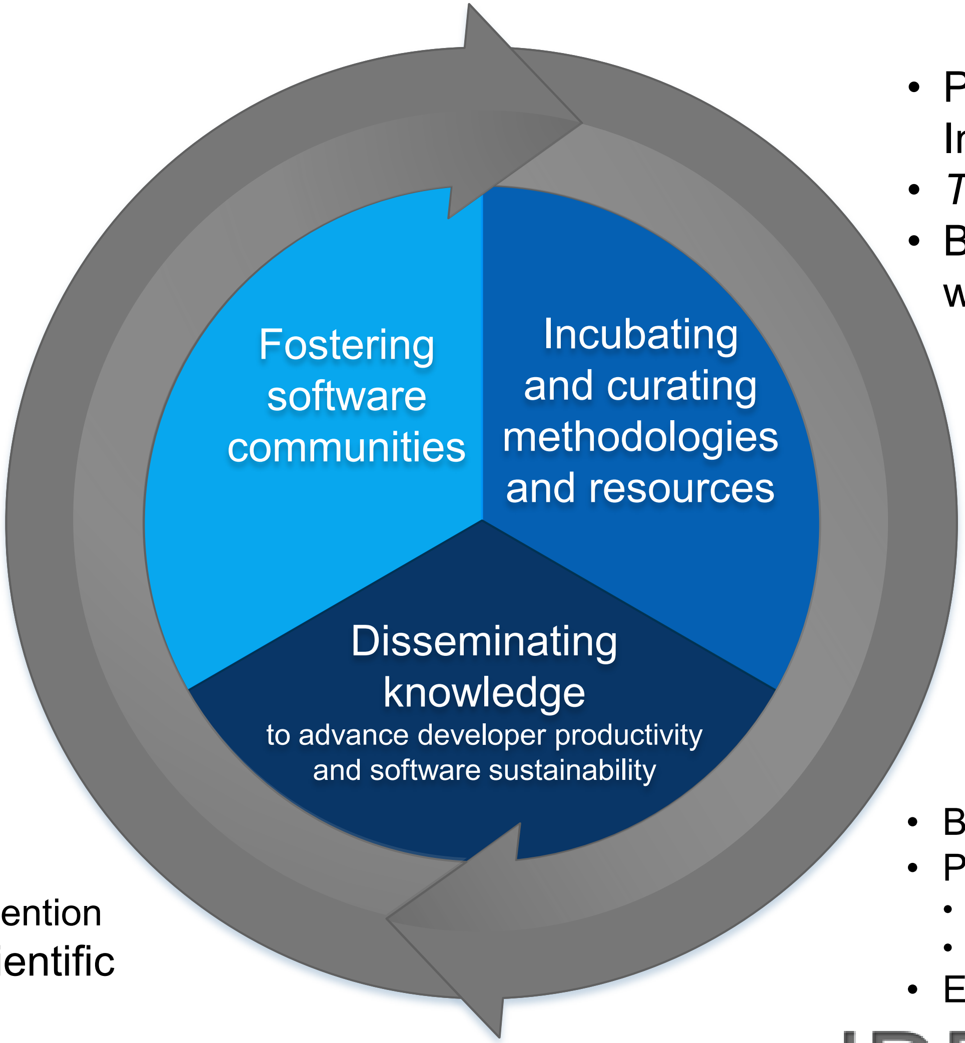
Scientific Software Ecosystems of Interest



ECP scientific software community
DOE scientific software community
General scientific software community

Diffusion of Innovations via Teams, Teams of Teams, and Communities

- Accelerating design space exploration
- Advancing quality, usability and interoperability, while respecting team autonomy
- Software community policies
- Software Development Kits (SDKs) and E4S
  - xSDK, CAT-SDK, DAV-SDK, SWAS, etc


Fostering software communities
Incubating and curating methodologies and resources
Disseminating knowledge
to advance developer productivity and software sustainability

- Productivity and Sustainability Improvement Planning (PSIP)
- *Team of teams* concepts
- Better Scientific Software (BSSw.io) website

- Webinar Series:
  - HPC Best Practices
  - HPC Workforce Development and Retention
- Tutorials on Practices for Better Scientific Software

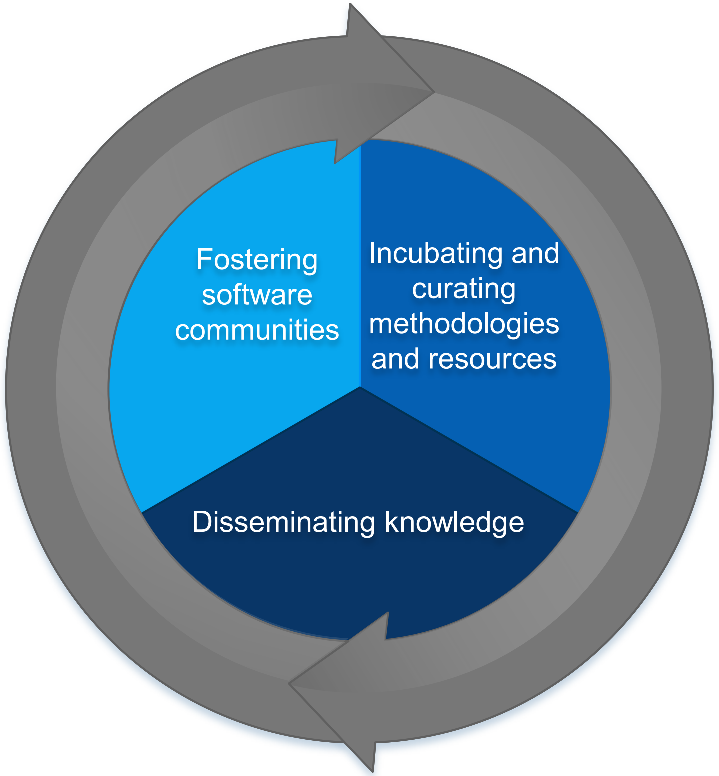
- BSSw Fellowship Program
- Panel Series:
  - Strategies for Working Remotely
  - Performance Portability
- Events: BOFs, workshops, and more

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Fostering Software Communities

- Accelerating design space exploration
- Advancing quality, usability, interoperability, and sustainability, while respecting team autonomy
- Software community policies
- Software Development Kits (SDKs) and E4S
  - xSDK, CAT-SDK, DAV-SDK, etc.


Fostering software communities
Incubating and curating methodologies and resources
Disseminating knowledge

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

xSDK lead: Ulrike Meier Yang (LLNL)
xSDK release lead: Satish Balay (ANL)

xSDK release 0.8.0 (Nov 2022)

- hypre
- PETSc/TAO
- SuperLU
- Trilinos

the initial xSDK 0.1.0 libraries

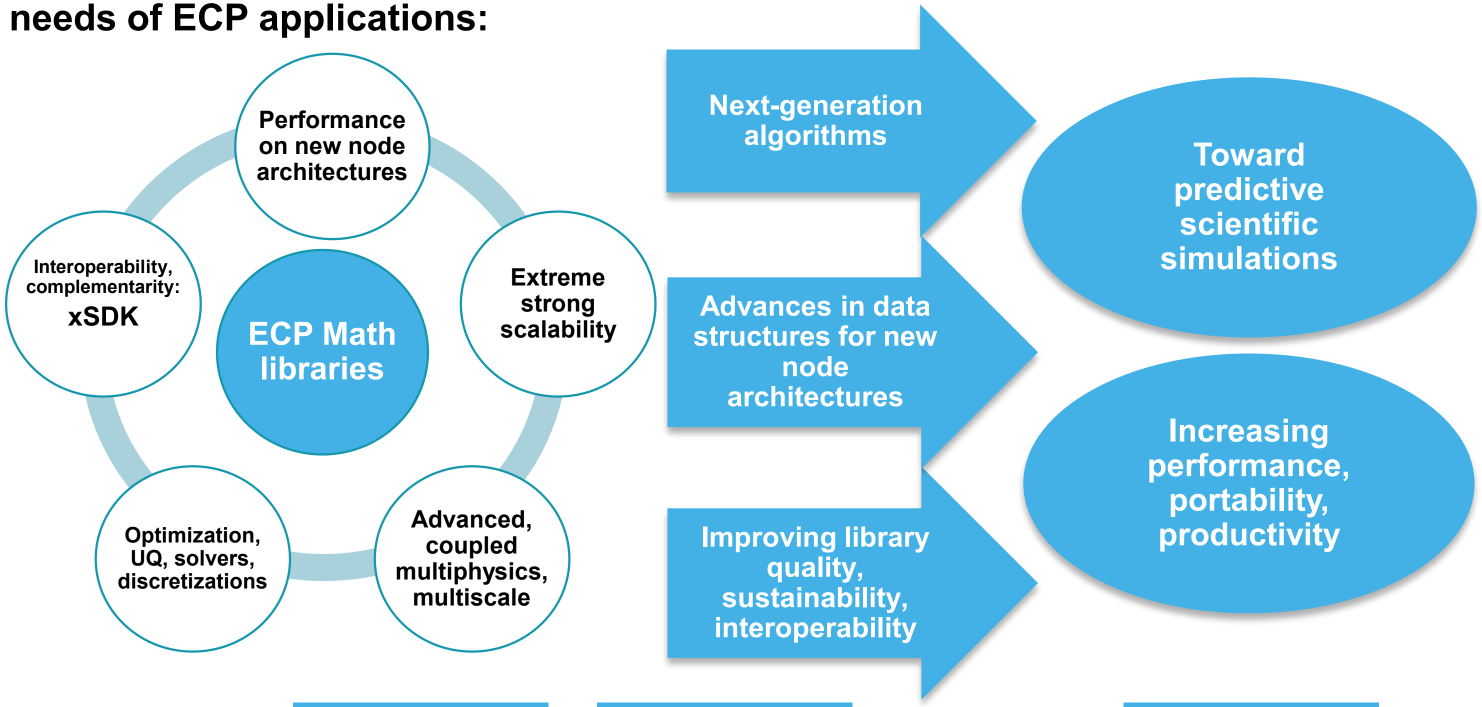
- Alquimia
- PFLOTRAN

domain libraries included in xSDK 0.1.0 or 0.2.0

- AMReX
- ArborX
- ButterflyPACK
- DTK
- Ginkgo
- heFFTe
- libEnsemble
- MAGMA
- MFEM
- Omega_h
- PLASMA
- PUMI
- SLATE
- Tasmanian
- SUNDIALS
- Strumpack
- 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

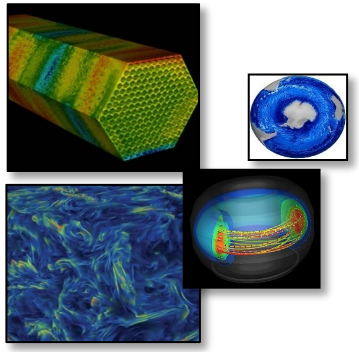
Next-generation algorithms

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

Refs: xSDK: Building an Ecosystem of Highly Efficient Math Libraries for Exascale, **SIAM News**, Jan 2021; Building Community through xSDK Software Policies, HPC-BP webinar, Dec 2019

xSDK Community Policies (v1.0.0, Feb 2023)
https://doi.org/10.6084/m9.figshare.13087196.v1

Mandatory Policies

1. Support portable installation through Spack
2. Provide a comprehensive test suite
3. Packages using MPI must not directly use MPI_COMM_WORLD
4. Package must support common platforms
5. Must provide a reliable way to contact developers
6. Respect decisions by other previously called packages about system resources (e.g., exception and signal handlers)
7. Permissive open-source licence
8. Runtime API call to return library version
9. Must not pollute namespaces
10. Must have a public repository
11. Must be able to turn off/off and redirect I/O
12. If a package embeds externally developed, it must be buildable against an outside copy of that software
13. Installations must conform to certain conventions for names and locations of header and library files
14. Must be buildable with 64-bit pointers
15. xSDK compatibility changes go into the regular development/release versions, not special branches
16. Build options must include a "debug" mode
17. Packages must provide sufficient documentation

Recommended Policies

1. Should have at least one "smoke" test
2. Should be possible to run test suite under Valgrind
3. Adopt and document a consistent system for handling error conditions
4. Free all resources acquired
5. Ability to export an ordered list of library dependencies
6. Document the versions of packages it works with or depends upon
7. Include README, SUPPORT, LICENSE, and CHANELOG files
8. Provide preprocessor macros to allow for version comparisons

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
- Growing functionality: E4S 23.08 – 115 packages

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 23.08 – August

Build caches
10X build time
improvement

Turnkey stack
A new user experience

https://e4s.io

E4S Strategy Group
US agencies, industry,
international

https://spack.io

Spack lead: Todd Gamblin (LLNL)

https://e4s.io

E4S lead: Sameer Shende (U Oregon)

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

E4S Community Policies: *A commitment to quality improvement*

- Purpose: Enhance sustainability and interoperability
- Will serve as membership criteria for E4S
  - Membership is not required for *inclusion* in E4S
  - Also includes forward-looking draft policies
- Modeled after xSDK community policies
- Multi-year effort led by SDK team
  - Included representation from across ST
  - Multiple rounds of feedback incorporated from ST leadership and membership

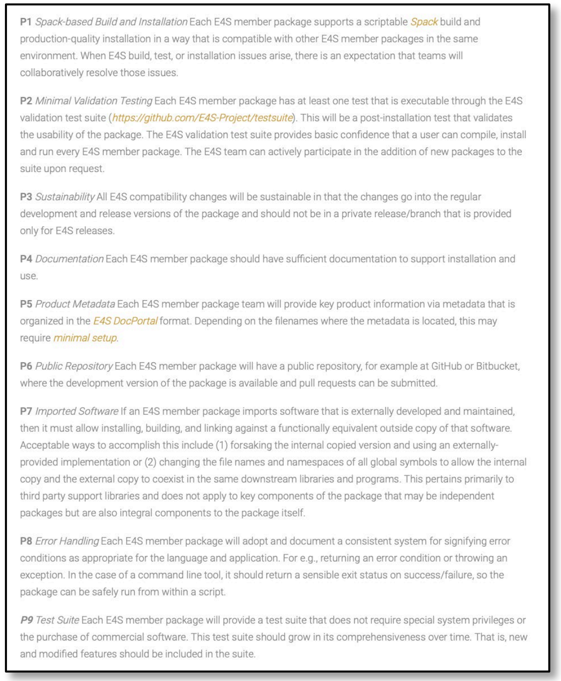
SDK lead: Jim Willenbring (SNL)

Policies: Version 1

Feedback welcome. What policies make sense for <u>your</u> software?

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

- **P1: *Spack-based Build and Installation***
- **P2: *Minimal Validation Testing***
- **P3: *Sustainability***
- **P4: *Documentation***
- **P5: *Product Metadata***
- **P6: *Public Repository***
- **P7: *Imported Software***
- **P8: *Error Handling***
- **P9: *Test Suite***



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.

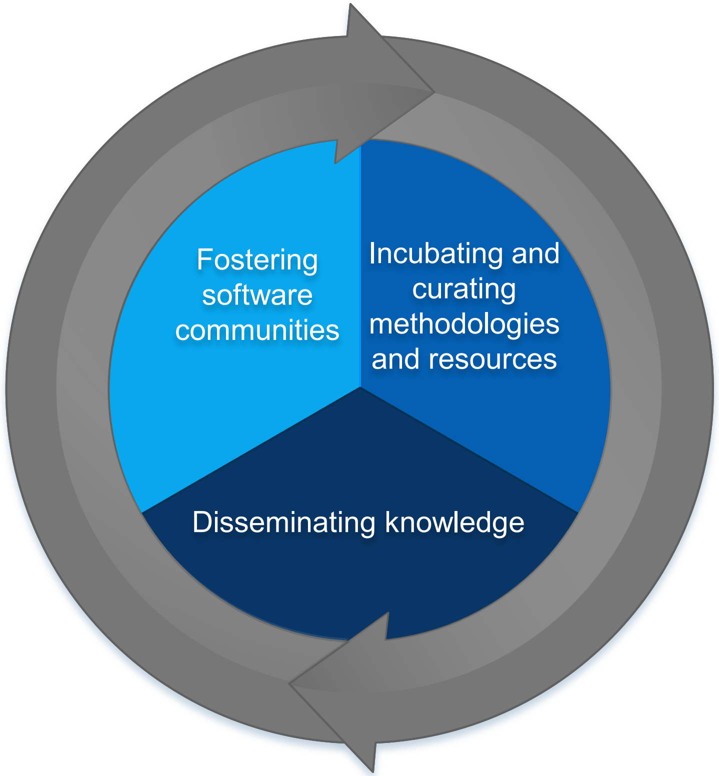
P7 *Imported Software* If an E4S member package imports software that is externally developed and maintained, then it must allow installing, building, and linking against a functionally equivalent outside copy of that software. Acceptable ways to accomplish this include (1) forsaking the internal copied 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.

Incubating and Curating Methodologies and Resources

- Productivity and Sustainability Improvement Planning (PSIP)
- Team of teams concepts
- Better Scientific Software (BSSw.io) resource portal


Fostering software communities
Incubating and curating methodologies and resources
Disseminating knowledge

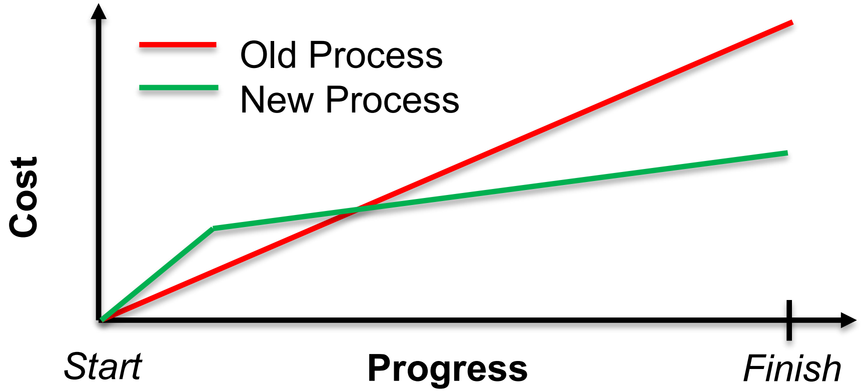
PSIP: Productivity and Sustainability Improvement Planning

Continual, Incremental Software Process Improvement

https://bssw.io/psip

1. Identify your team's "pain points" in your software development processes
   - Help: RateYourProject assessment tool: https://rateyourproject.org/
2. Set a goal for something to improve
   - Target processes and behaviors, not just tasks
   - Pick something that you can address in a few months that will give you a noticeable benefit
3. Agree on a plan to address it, identify markers of progress and what is "done"
   - Write them down
   - Help: Progress tracking card examples: https://bssw-psip.github.io/ptc-catalog/catalog
4. Work your plan, track your progress
5. When you are done, celebrate…

…then pick a new pain point to address



The new process costs something to implement, but it pays off over time

Target: your project should include "just enough" software engineering so that you can meet your short-term and longer-term scientific goals effectively

Lead: Elaine Raybourn (SNL)

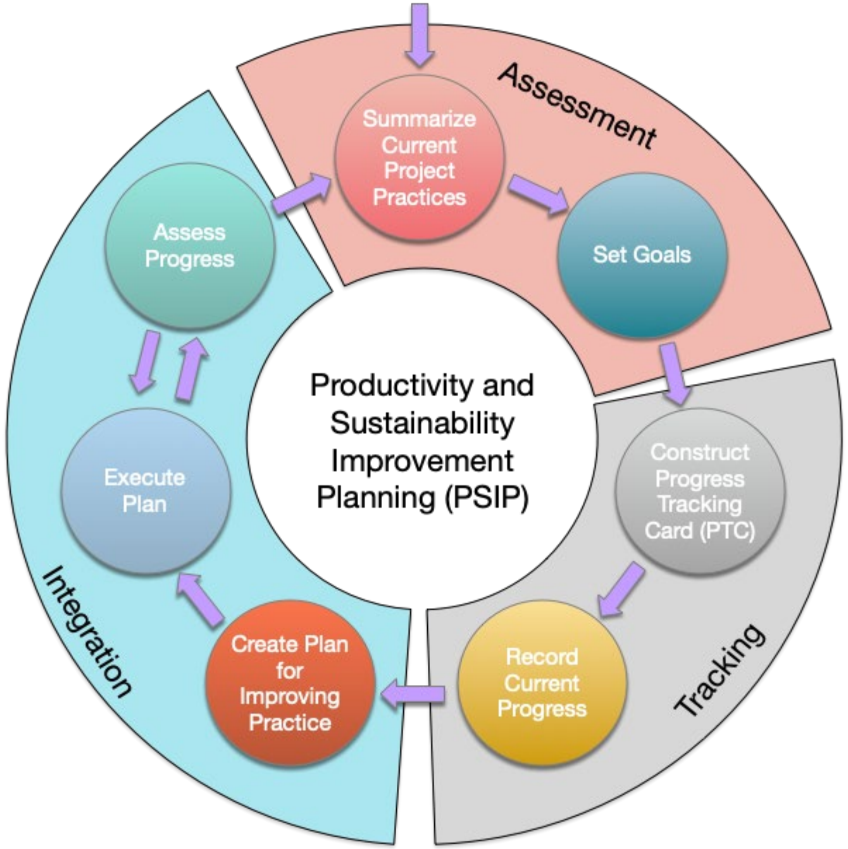
PSIP multi-pronged, socio-technical strategy

EDUCATION

Share knowledge via tutorials, paper publications, and seminars to engender a culture of productivity by engaging PSIP liaisons, early adopters, and champions

TEAM ENGAGEMENT

By directly working with ECP teams, PSIP facilitators help software teams to **IDENTIFY** opportunities to iteratively and incrementally **IMPROVE** software team practices and processes.



Productivity and Sustainability Improvement Planning (PSIP)
Assessment
Summarize Current Project Practices
Set Goals
Tracking
Construct Progress Tracking Card (PTC)
Record Current Progress
Integration
Create Plan for Improving Practice
Execute Plan
Assess Progress

RESEARCH

Conduct Team of Teams data-driven research (data mining + interviews) to characterize ECP organization, draw general conclusions about observable behaviors that contribute to team productivity, collaboration, and success

METHOD & TOOL DEVELOPMENT

Develop tools that automate PSIP to support ECP teams, scientific software quality, and developer productivity. Development of PSIP tools based on needs and salient issues identified during Team of Teams research and ECP team engagement

PSIP: Collaborative and **team-oriented.**
Measurable and **specific. Realistic increments.**

IDEAS productivity EXASCALE COMPUTING PROJECT

Examples of Who’s using PSIP

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

The HDF Group

Improvements to documentation, setting code style standards, transition to GitHub (blog article)

FLASH5

Improve testing and verification, transition development workflow to GitHub

EXAALT

Revamp build system, implement a CTest-based testing framework, implement a basic CI pipeline

ALPINE/ZFP

Created a VTK-m filter for APLINE in situ algorithm users

OAK RIDGE National Laboratory

Using a more detailed version for internal project assessment

TECHNISCHE UNIVERSITÄT DARMSTADT

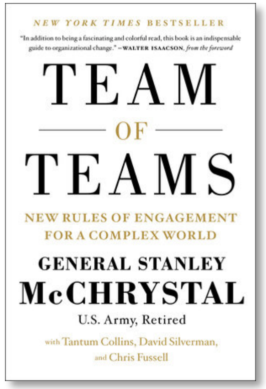
Completed PSIP tutorial, investigating how it can be used in academic context

Sandia National Laboratories

Using internally for reproducibility LDRD research, and for large projects updating version control systems, and updating documentation to support better onboarding

Collaboration via Teams of Teams

- The “team of teams” concept (ToT) was popularized by Stanley McChrystal’s 2015 book
  - IDEAS efforts are an offshoot of PSIP, led by Elaine Raybourn (SNL)
  - Using tools from the CAT-SDK software community for repository analysis
- ToT provides a powerful lens through which to better understand the ECP, as well as many other software ecosystems, and to improve their effectiveness
  - Strengthen community partnerships
  - Scaling productivity typically experienced in small teams (where it's easy), to larger groups via the team of teams paradigm
- ToT principles facilitated contributions of the HDF5 team to the E4S and Data & Viz SDK
  - Supported applications in modeling earthquakes, electronic structures, subsurface flow, reacting flow, stellar explosions, wind plants, and cosmology
- Distributed, Interconnected Teams through the Lens of Team of Teams Principles
  - Panel discussion with members of PETSc, Trilinos, xSDK, and E4S ECP projects
- Scaling productivity and innovation on the path to exascale with a “team of teams” approach
  - Case study of the ASC Ristra ECP project



NEW YORK TIMES BESTSELLER

“In addition to being a fascinating and colorful read, this book is an indispensable guide to organizational change.” —WALTER ISAACSON, *from the foreword*

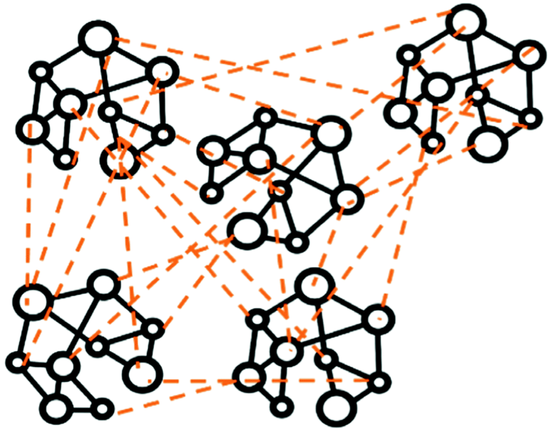
TEAM OF TEAMS

NEW RULES OF ENGAGEMENT FOR A COMPLEX WORLD

GENERAL STANLEY McCHRYSTAL

U.S. Army, Retired

with Tantum Collins, David Silverman, and Chris Fussell



Schematic illustration of a team of teams, from doi:10.1007/978-3-030-22338-0_33

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

better
scientific
software

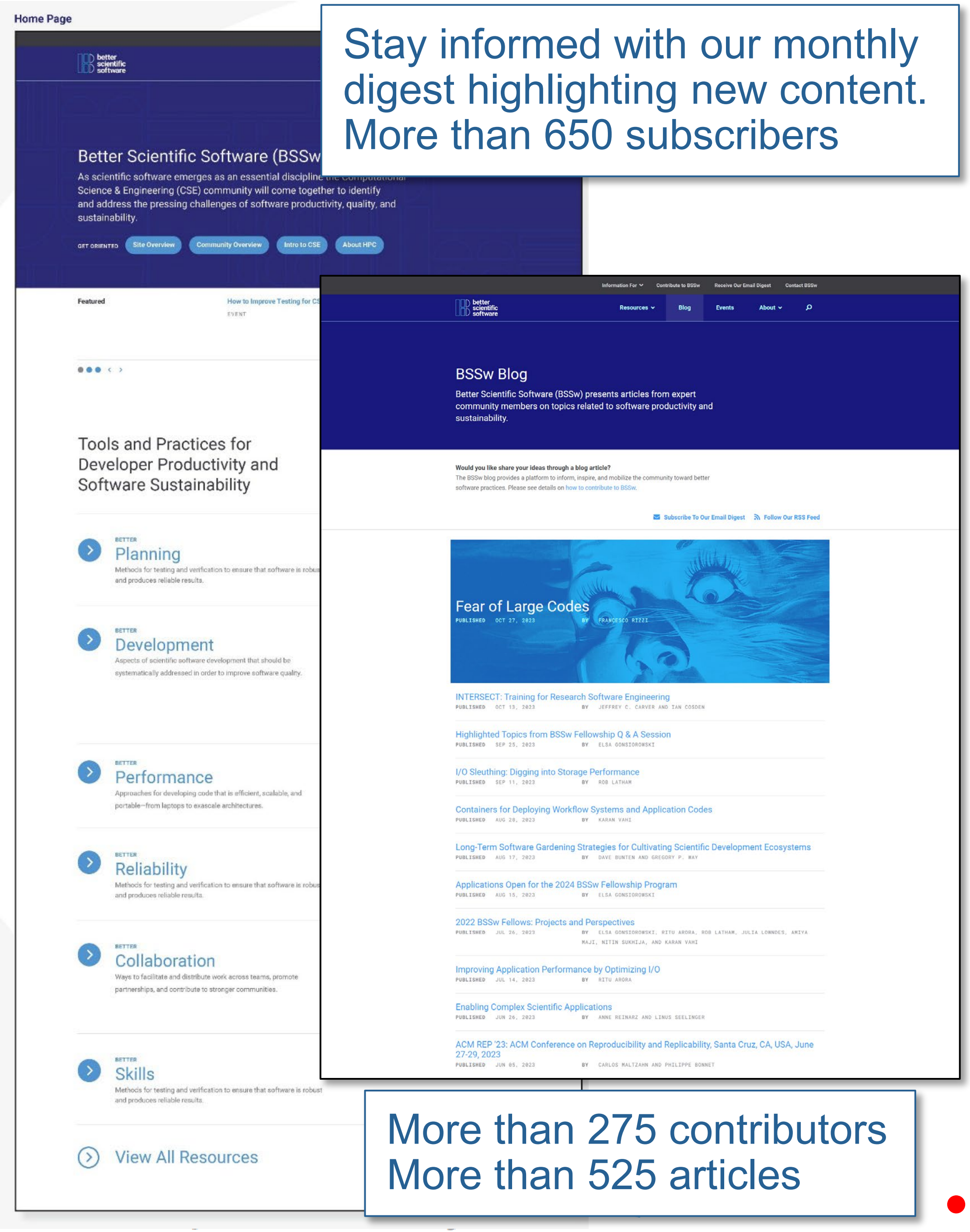
https://bssw.io

A central hub for sharing information on practices, techniques, experiences, and tools to improve developer productivity and software sustainability for computational science & engineering (CSE)

- **Find information** on scientific software topics
- **Contribute new resources** based on your experiences
- Editor-in-chief: Rinku Gupta (ANL)

Types of content on BSSw

- **Blog articles**: success stories, perspectives, opportunities, technical deep-dives, and more
- **Curated content:** short pointers to useful material already hosted elsewhere
- **Events:** increase awareness of events related to better scientific software



Home Page

Better Scientific Software (BSSw

As scientific software emerges as an essential discipline the Computational Science & Engineering (CSE) community will come together to identify and address the pressing challenges of software productivity, quality, and sustainability.

Featured

How to Improve Testing for CS
EVENT

Tools and Practices for Developer Productivity and Software Sustainability

BETTER Planning
Methods for testing and verification to ensure that software is robus and produces reliable results.

BETTER Development
Aspects of scientific software development that should be systematically addressed in order to improve software quality.

BETTER Performance
Approaches for developing code that is efficient, scalable, and portable—from laptops to exascale architectures.

BETTER Reliability
Methods for testing and verification to ensure that software is robus and produces reliable results.

BETTER Collaboration
Ways to facilitate and distribute work across teams, promote partnerships, and contribute to stronger communities.

BETTER Skills
Methods for testing and verification to ensure that software is robust and produces reliable results.

View All Resources

BSSw Blog

Better Scientific Software (BSSw) presents articles from expert community members on topics related to software productivity and sustainability.

Would you like share your ideas through a blog article?
The BSSw blog provides a platform to inform, inspire, and mobilize the community toward better software practices. Please see details on how to contribute to BSSw.

Fear of Large Codes
PUBLISHED OCT 27, 2023 BY FRANCESCO RIZZI

INTERSECT: Training for Research Software Engineering
PUBLISHED OCT 13, 2023 BY JEFFREY C. CARVER AND IAN COSDEN

Highlighted Topics from BSSw Fellowship Q & A Session
PUBLISHED SEP 25, 2023 BY ELSA GONSIOROWSKI

I/O Sleuthing: Digging into Storage Performance
PUBLISHED SEP 11, 2023 BY ROB LATHAM

Containers for Deploying Workflow Systems and Application Codes
PUBLISHED AUG 28, 2023 BY KARAN VAHI

Long-Term Software Gardening Strategies for Cultivating Scientific Development Ecosystems
PUBLISHED AUG 17, 2023 BY DAVE BUNTEN AND GREGORY P. WAY

Applications Open for the 2024 BSSw Fellowship Program
PUBLISHED AUG 15, 2023 BY ELSA GONSIOROWSKI

2022 BSSw Fellows: Projects and Perspectives
PUBLISHED JUL 26, 2023 BY ELSA GONSIOROWSKI, RITU ARORA, ROB LATHAM, JULIA LOWNDES, AMIYA MAJI, NITIN SUKHIJA, AND KARAN VAHI

Improving Application Performance by Optimizing I/O
PUBLISHED JUL 14, 2023 BY RITU ARORA

Enabling Complex Scientific Applications
PUBLISHED JUN 26, 2023 BY ANNE REINARZ AND LINUS SEELINGER

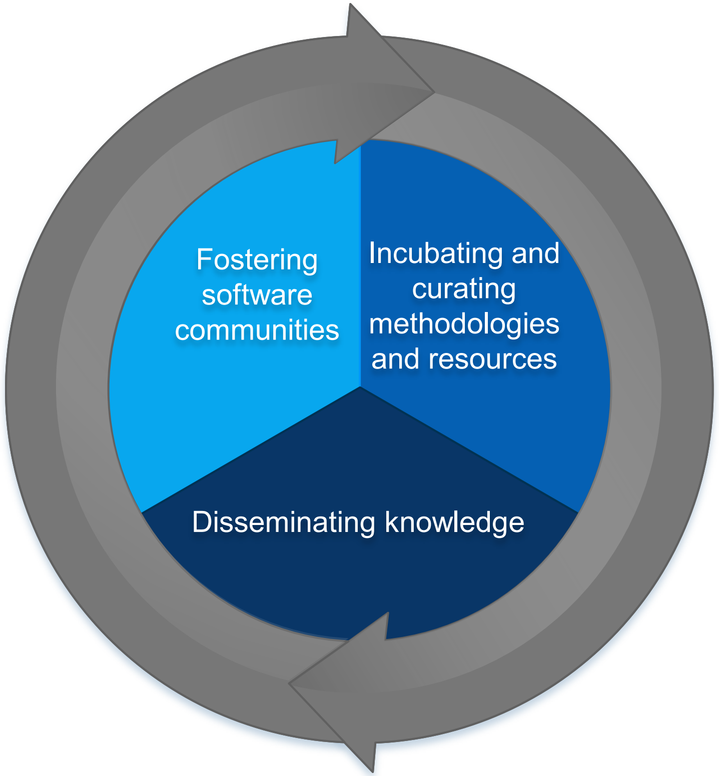
ACM REP '23: ACM Conference on Reproducibility and Replicability, Santa Cruz, CA, USA, June 27-29, 2023
PUBLISHED JUN 05, 2023 BY CARLOS MALTZAHN AND PHILIPPE BONNET

Stay informed with our monthly digest highlighting new content. More than 650 subscribers

More than 275 contributors
More than 525 articles

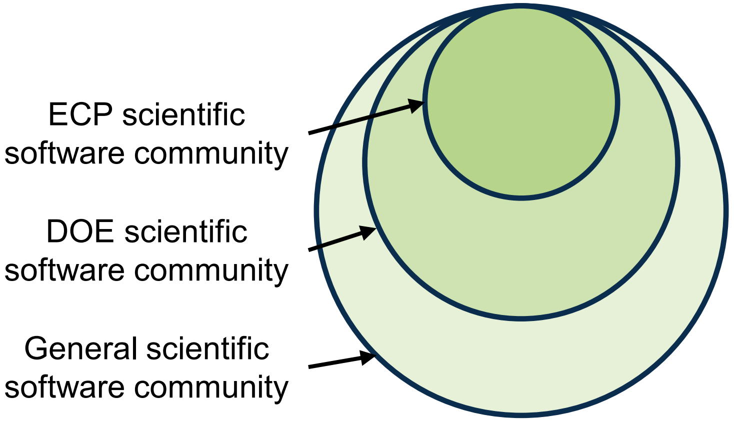
Disseminating Knowledge

- BSSw Fellowship Program
- Better Scientific Software tutorials
- Best Practices for HPC Software Developers webinar series
- Panel Series:
  - Strategies for Working Remotely
  - Performance Portability
- Events: BOFs, workshops, and more



Fostering software communities
Incubating and curating methodologies and resources
Disseminating knowledge

Scientific Software Ecosystems of Interest



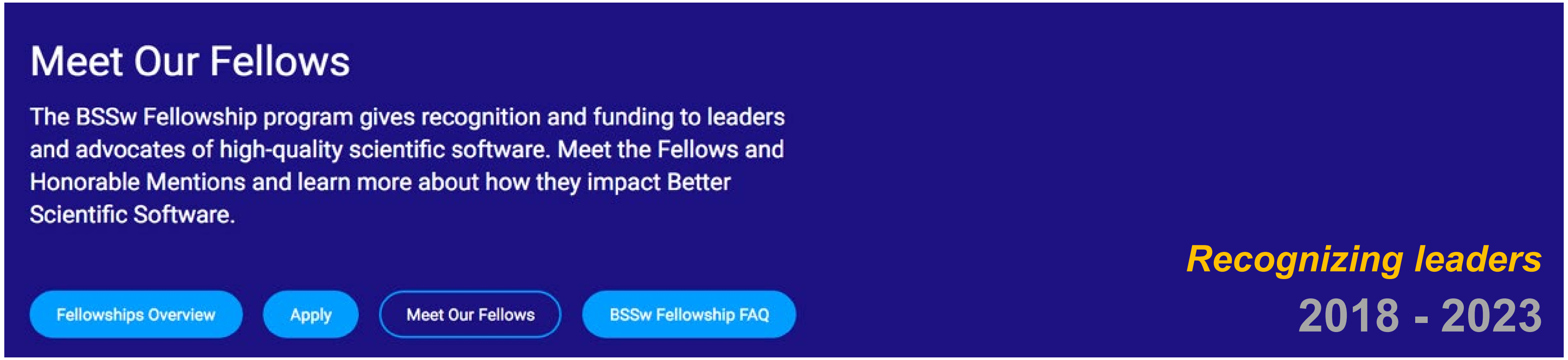
ECP scientific software community
DOE scientific software community
General scientific software community

IDEAS dissemination and outreach activities have always targeted the general scientific software community

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Better Scientific Software (BSSw) Fellowship Program

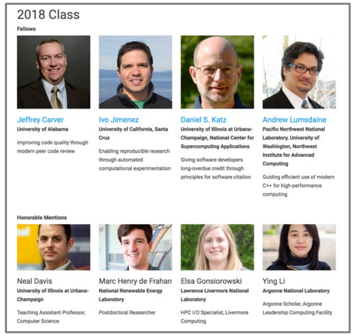


Meet Our Fellows

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.

Recognizing leaders

2018 - 2023



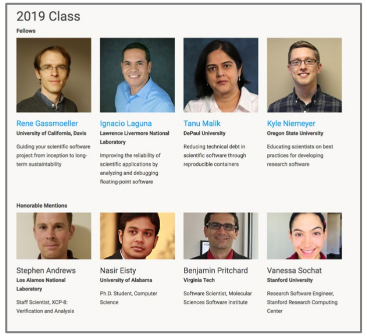
2018 Class

Fellows

- Jeffrey Carver — **University of Alabama** — Improving code quality through modern peer code review
- Ivo Jimenez — **University of California, Santa Cruz** — Enabling reproducible research through automated computational experimentation
- Daniel S. Katz — **University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications** — Giving software developers long-overdue credit through principles for software citation
- Andrew Lumsdaine — **Pacific Northwest National Laboratory, University of Washington, Northwest Institute for Advanced Computing** — Guiding efficient use of modern C++ for high-performance computing

Honorable Mentions

- Neal Davis — **University of Illinois at Urbana-Champaign** — Teaching Assistant Professor, Computer Science
- Marc Henry de Frahan — **National Renewable Energy Laboratory** — Postdoctoral Researcher
- Elsa Gonsiorowski — **Lawrence Livermore National Laboratory** — HPC I/O Specialist, Livermore Computing
- Ying Li — **Argonne National Laboratory** — Argonne Scholar, Argonne Leadership Computing Facility



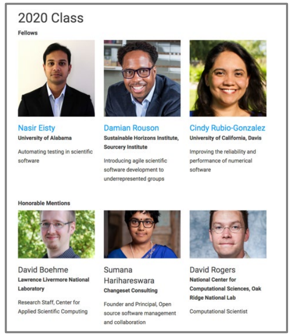
2019 Class

Fellows

- Rene Gassmoeller — **University of California, Davis** — Guiding your scientific software project from inception to long-term sustaintability
- Ignacio Laguna — **Lawrence Livermore National Laboratory** — Improving the reliability of scientific applications by analyzing and debugging floating-point software
- Tanu Malik — **DePaul University** — Reducing technical debt in scientific software through reproducible containers
- Kyle Niemeyer — **Oregon State University** — Educating scientists on best practices for developing research software

Honorable Mentions

- Stephen Andrews — **Los Alamos National Laboratory** — Staff Scientist, XCP-8: Verification and Analysis
- Nasir Eisty — **University of Alabama** — Ph.D. Student, Computer Science
- Benjamin Pritchard — **Virginia Tech** — Software Scientist, Molecular Sciences Software Institute
- Vanessa Sochat — **Stanford University** — Research Software Engineer, Stanford Research Computing Center



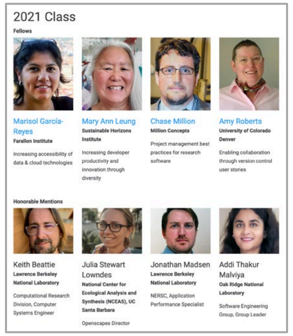
2020 Class

Fellows

- Nasir Eisty — **University of Alabama** — Automating testing in scientific software
- Damian Rouson — **Sustainable Horizons Institute, Sourcery Institute** — Introducing agile scientific software development to underrepresented groups
- Cindy Rubio-Gonzalez — **University of California, Davis** — Improving the reliability and performance of numerical software

Honorable Mentions

- David Boehme — **Lawrence Livermore National Laboratory** — Research Staff, Center for Applied Scientific Computing
- Sumana Harihareswara — **Changeset Consulting** — Founder and Principal, Open source software management and collaboration
- David Rogers — **National Center for Computational Sciences, Oak Ridge National Lab** — Computational Scientist



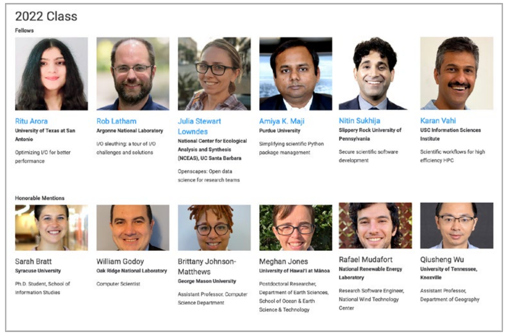
2021 Class

Fellows

- Marisol García-Reyes — **Farallon Institute** — Increasing accessibility of data & cloud technologies
- Mary Ann Leung — **Sustainable Horizons Institute** — Increasing developer productivity and innovation through diversity
- Chase Million — **Million Concepts** — Project management best practices for research software
- Amy Roberts — **University of Colorado Denver** — Enabling collaboration through version control user stories

Honorable Mentions

- Keith Beattie — **Lawrence Berkeley National Laboratory** — Computational Research Division, Computer Systems Engineer
- Julia Stewart Lowndes — **National Center for Ecological Analysis and Synthesis (NCEAS), UC Santa Barbara** — Openscapes Director
- Jonathan Madsen — **Lawrence Berkeley National Laboratory** — NERSC, Application Performance Specialist
- Addi Thakur Malviya — **Oak Ridge National Laboratory** — Software Engineering Group, Group Leader



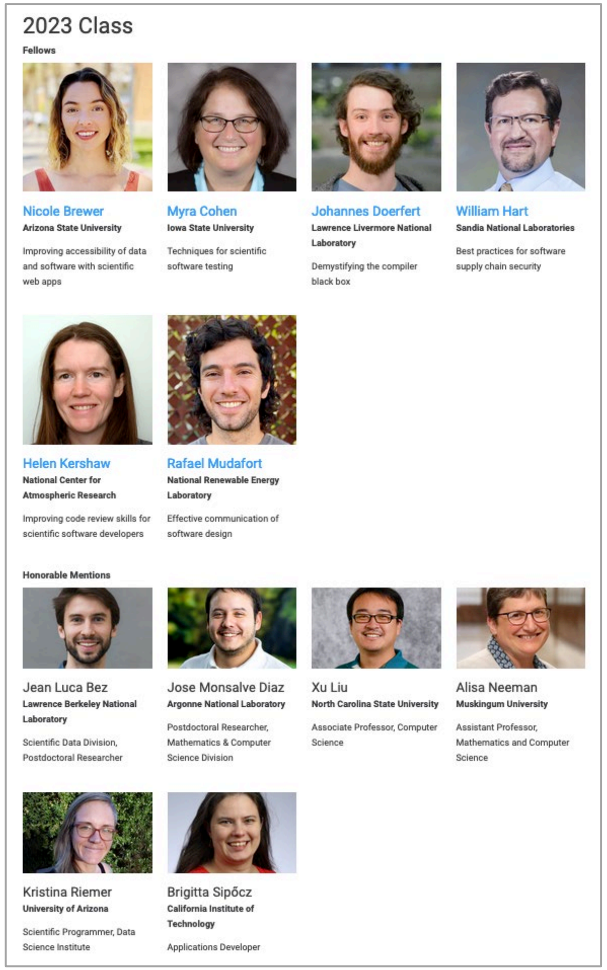
2022 Class

Fellows

- Ritu Arora — **University of Texas at San Antonio** — Optimizing I/O for better performance
- Rob Latham — **Argonne National Laboratory** — I/O sleuthing: a tour of I/O challenges and solutions
- Julia Stewart Lowndes — **National Center for Ecological Analysis and Synthesis (NCEAS), UC Santa Barbara** — Openscapes: Open data science for research teams
- Amiya K. Maji — **Purdue University** — Simplifying scientific Python package management
- Nitin Sukhija — **Slippery Rock University of Pennsylvania** — Secure scientific software development
- Karan Vahi — **USC Information Sciences Institute** — Scientific workflows for high efficiency HPC

Honorable Mentions

- Sarah Bratt — **Syracuse University** — Ph.D. Student, School of Information Studies
- William Godoy — **Oak Ridge National Laboratory** — Computer Scientist
- Brittany Johnson-Matthews — **George Mason University** — Assistant Professor, Computer Science Department
- Meghan Jones — **University of Hawai'i at Mānoa** — Postdoctoral Researcher, Department of Earth Sciences, School of Ocean & Earth Science & Technology
- Rafael Mudafort — **National Renewable Energy Laboratory** — Research Software Engineer, National Wind Technology Center
- Qiusheng Wu — **University of Tennessee, Knoxville** — Assistant Professor, Department of Geography



2023 Class

Fellows

- Nicole Brewer — **Arizona State University** — Improving accessibility of data and software with scientific web apps
- Myra Cohen — **Iowa State University** — Techniques for scientific software testing
- Johannes Doerfert — **Lawrence Livermore National Laboratory** — Demystifying the compiler black box
- William Hart — **Sandia National Laboratories** — Best practices for software supply chain security
- Helen Kershaw — **National Center for Atmospheric Research** — Improving code review skills for scientific software developers
- Rafael Mudafort — **National Renewable Energy Laboratory** — Effective communication of software design

Honorable Mentions

- Jean Luca Bez — **Lawrence Berkeley National Laboratory** — Scientific Data Division, Postdoctoral Researcher
- Jose Monsalve Diaz — **Argonne National Laboratory** — Postdoctoral Researcher, Mathematics & Computer Science Division
- Xu Liu — **North Carolina State University** — Associate Professor, Computer Science
- Alisa Neeman — **Muskingum University** — Assistant Professor, Mathematics and Computer Science
- Kristina Riemer — **University of Arizona** — Scientific Programmer, Data Science Institute
- Brigitta Sipőcz — **California Institute of Technology** — Applications Developer

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

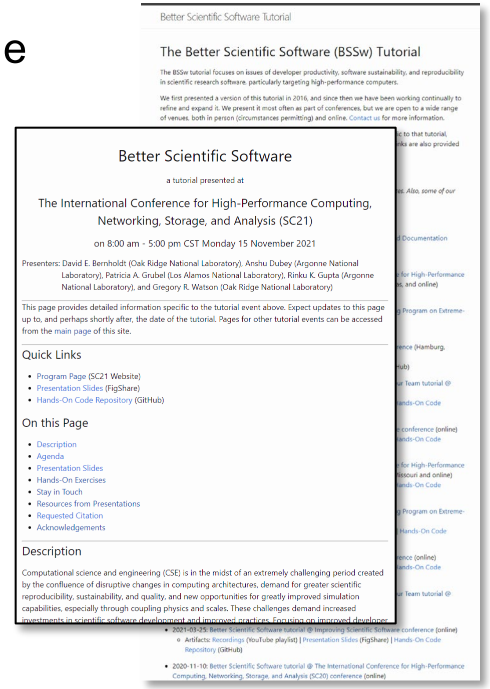
Also supported by the National Science Foundation since 2021

BSSw Fellowship Coordinator: Elsa Gonsiorowski (LLNL)

Deputy Coordinator, Community Building: Erik Palmer (LBNL)

Better Scientific Software Tutorials

- Covering issues of developer productivity, software sustainability and reliability, with a special focus on the challenges of complex, large-scale HPC
  - software design, agile methodologies, Git workflows, reproducibility, software testing, continuous integration testing, refactoring, and more
- https://bssw-tutorial.github.io
- Lead: David Bernholdt (ORNL)
- 32 tutorials since 2016
  - Presentations (all) and recordings (some) available
- Topics and content under continuous refinement
- Frequent venues
  - Supercomputing (2016-2023)
  - ATPESC (2016-2023)
  - ISC (2018-2019, 2021-2023)



Better Scientific Software Tutorial

The Better Scientific Software (BSSw) Tutorial

The BSSw tutorial focuses on issues of developer productivity, software sustainability, and reproducibility in scientific research software, particularly targeting high-performance computers.

We first presented a version of this tutorial in 2016, and since then we have been working continually to refine and expand it. We present it most often as part of conferences, but we are open to a wide range of venues, both in person (circumstances permitting) and online. Contact us for more information.

- 2021-03-25: Better Scientific Software tutorial @ Improving Scientific Software conference (online)
  - Artifacts: Recordings (YouTube playlist) | Presentation Slides (FigShare) | Hands-On Code Repository (GitHub)
- 2020-11-10: Better Scientific Software tutorial @ The International Conference for High-Performance Computing, Networking, Storage, and Analysis (SC20) conference (online)

Better Scientific Software

a tutorial presented at

The International Conference for High-Performance Computing, Networking, Storage, and Analysis (SC21)

on 8:00 am - 5:00 pm CST Monday 15 November 2021

Presenters: David E. Bernholdt (Oak Ridge National Laboratory), Anshu Dubey (Argonne National Laboratory), Patricia A. Grubel (Los Alamos National Laboratory), Rinku K. Gupta (Argonne National Laboratory), and Gregory R. Watson (Oak Ridge National Laboratory)

This page provides detailed information specific to the tutorial event above. Expect updates to this page up to, and perhaps shortly after, the date of the tutorial. Pages for other tutorial events can be accessed from the main page of this site.

Quick Links

- Program Page (SC21 Website)
- Presentation Slides (FigShare)
- Hands-On Code Repository (GitHub)

On this Page

- Description
- Agenda
- Presentation Slides
- Hands-On Exercises
- Stay in Touch
- Resources from Presentations
- Requested Citation
- Acknowledgements

Description

Computational science and engineering (CSE) is in the midst of an extremely challenging period created by the confluence of disruptive changes in computing architectures, demand for greater scientific reproducibility, sustainability, and quality, and new opportunities for greatly improved simulation capabilities, especially through coupling physics and scales. These challenges demand increased investments in scientific software development and improved practices. Focusing on improved developer

Popper falsifiable.us

Practical Reproducible Evaluation of Computer Systems

Ivo Jimenez, Michael Sevilla, Noah Watkins,

On-demand Learning for Better Scientific Software: How to Use Resources & Technology to Optimize your Productivity

A Best Practices for HPC Software Developers Webinar

aine M. Raybourn
incipal Member of the Technical Staff
andia National Laboratories
nraybo@sandia.gov, @elaineraybourn

ay 9, 2018

Python in HPC

IDEAS Webinar
June 7, 2017

as (NERSC), William Scullin (ALCF),
nd Matt Belhorn (OLCF)

Open Source Best Practices: From Continuous Integration to Static Linters

Barely Sufficient Project Management
A few techniques for improving your scientific software development efforts

HPC Best Practices Webinar Series

Michael A. Heroux
Senior Scientist, Sandia National Laboratories
Scientist in Residence, St. John's University, MN

jupyter
Jupyter in HPC

Feb 28th, 2018

Matthias Bussonnier

bussonniermatthias@gmail.com
GitHub: @carreau
Twitter: @mbussonn

Managing Defects in HPC Software Development

Presented to
OLCF Webinar Series

Tom Evans
ORNL, PI ExaSMR ECP Applications Project

November 1, 2017

Bringing Best Practices to a Long-Lived Production Code

Software sustainability - lessons learned from different disciplines

https://doi.org/10.6084/m9.figshare.6935840
21st August 2018, Best Practices for HPC Software Developers Webinar Series
Neil Chue Hong (@npch), Software Sustainability Institute
ORCID: 0000-0002-8876-7606 | N.ChueHong@software.ac.uk

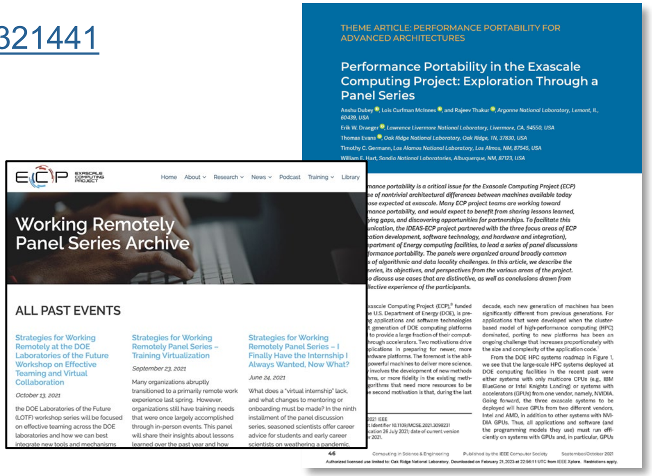
Supported by EPSRC BBSRC
Project funding from Jisc NERC

Webinar Series: Best Practices for HPC Software Developers (HPC-BP)

- Covering topics in software development and HPC
- https://ideas-productivity.org/resources/series/hpc-best-practices-webinars/
- Lead: Osni Marques (LBNL)
- Presented by the community to the community
- Monthly series, since May 2016 (offered live and archived)
  - To date: 80 webinars, >12,000 registrations, >5,300 attendees
  - 84 attendees per webinar, on average

Panel Series: Performance Portability & ECP

- Lead: Anshu Dubey (ANL). Refs:
  - Performance Portability in the Exascale Computing Project: Exploration Through a Panel Series, A. Dubey et al, IEEE CiSE, Sept 2021
  - SIAM CSE21 minisymposium: https://doi.org/10.6084/m9.figshare.c.5321441
  - ECCOMAS 2022 minisymposium



THEME ARTICLE: PERFORMANCE PORTABILITY FOR ADVANCED ARCHITECTURES

Performance Portability in the Exascale Computing Project: Exploration Through a Panel Series

Anshu Dubey, Lois Curfman McInnes, and Rajeev Thakur, Argonne National Laboratory, Lemont, IL, 60439, USA
Erik W. Draeger, Lawrence Livermore National Laboratory, Livermore, CA, 94550, USA
Thomas Evans, Oak Ridge National Laboratory, Oak Ridge, TN, 37830, USA
Timothy C. Germann, Los Alamos National Laboratory, Los Almos, NM, 87545, USA
William E. Hart, Sandia National Laboratories, Albuquerque, NM, 87123, USA

mance portability is a critical issue for the Exascale Computing Project (ECP) se of nontrivial architectural differences between machines available today ose expected at exascale. Many ECP project teams are working toward mance portability, and would expect to benefit from sharing lessons learned, ying gaps, and discovering opportunities for partnerships. To facilitate this unication, the IDEAS-ECP project partnered with the three focus areas of ECP ation development, software technology, and hardware and integration), partment of Energy computing facilities, to lead a series of panel discussions formance portability. The panels were organized around broadly common s of algorithmic and data locality challenges. In this article, we describe the series, its objectives, and perspectives from the various areas of the project. o discuss use cases that are distinctive, as well as conclusions drawn from llective experience of the participants.

xascale Computing Project (ECP),[a] funded e U.S. Department of Energy (DOE), is pre- g applications and software technologies t generation of DOE computing platforms to provide a large fraction of their comput- hrough accelerators. Two motivations drive plications in preparing for newer, more rdware platforms. The foremost is the abil- powerful machines to deliver more science. involves the development of new methods hms, or more fidelity in the existing meth- gorithms that need more resources to be e second motivation is that, during the last

decade, each new generation of machines has been significantly different from previous generations. For applications that were developed when the cluster-based model of high-performance computing (HPC) dominated, porting to new platforms has been an ongoing challenge that increases proportionately with the size and complexity of the application code.[1]

From the DOE HPC systems roadmap in Figure 1, we see that the large-scale HPC systems deployed at DOE computing facilities in the recent past were either systems with only multicore CPUs (e.g., IBM BlueGene or Intel Knights Landing) or systems with accelerators (GPUs) from one vendor, namely, NVIDIA. Going forward, the three exascale systems to be deployed will have GPUs from two different vendors, Intel and AMD, in addition to other systems with NVIDIA GPUs. Thus, all applications and software (and the programming models they use) must run efficiently on systems with GPUs and, in particular, GPUs

2021 IEEE
t Identifier 10.1109/MCSE.2021.3098231
cation 26 July 2021; date of current version
r 2021.

46 Computing in Science & Engineering Published by the IEEE Computer Society September/October 2021

Authorized licensed use limited to: Oak Ridge National Laboratory. Downloaded on February 21,2023 at 22:56:11 UTC from IEEE Xplore. Restrictions apply.

Working Remotely Panel Series Archive

ALL PAST EVENTS

Strategies for Working Remotely at the DOE Laboratories of the Future Workshop on Effective Teaming and Virtual Collaboration
October 13, 2021
the DOE Laboratories of the Future (LOTF) workshop series will be focused on effective teaming across the DOE laboratories and how we can best integrate new tools and mechanisms

Strategies for Working Remotely Panel Series – Training Virtualization
September 23, 2021
Many organizations abruptly transitioned to a primarily remote work experience last spring. However, organizations still have training needs that were once largely accomplished through in-person events. This panel will share their insights about lessons learned over the past year and how

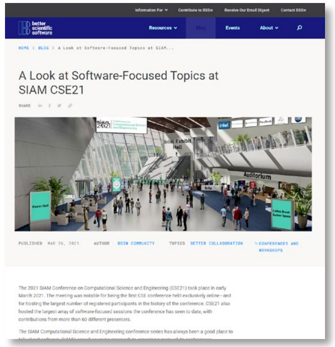
Strategies for Working Remotely Panel Series – I Finally Have the Internship I Always Wanted, Now What?
June 24, 2021
What does a "virtual internship" lack, and what changes to mentoring or onboarding must be made? In the ninth installment of the panel discussion series, seasoned scientists offer career advice for students and early career scientists on weathering a pandemic,

Panel Series: Strategies for Working Remotely

- Exploring strategies for working remotely, with emphasis on how HPC teams can be effective and efficient in long-term hybrid settings
- https://www.exascaleproject.org/strategies-for-working-remotely
- Lead: Elaine Raybourn (SNL)
- Quarterly series, since April 2020 (offered live and archived)
- Ref: Why We Need Strategies for Working Remotely: The ECP Panel Series, E. Raybourn, SC20 State of the Practice, Nov 2020

Technical Meetings and Birds of a Feather Sessions

- Creating opportunities to talk about software development, productivity, and sustainability
- https://ideas-productivity.org/resources/series/technical-sessions-and-meetings/
- Minisymposia
  - SIAM CSE, SIAM PP (2015-2023), PASC (2018, 2019)
  - Ref: A Look at Software-Focused Topics at SIAM CSE21, March 2021
- Thematic poster sessions
  - SIAM CSE (2017, 2019, 2021)
- BOF sessions
  - Software Engineering and Reuse in Modeling, Simulation and Data Analytics for Science and Engineering
    - http://bit.ly/swe-cse-bof
    - Supercomputing (2015-2023), ISC (2019, 2022-2023)
- Collegeville Workshop Series on Scientific Software,
  - Ref: Software Team Experiences and Challenges, K. Beattie et al, Oct 2021



Information For
Contribute to BSSw
Receive Our Email Digest
Contact BSSw
better scientific software
Resources
Blog
Events
About
HOME > BLOG > A Look at Software-Focused Topics at SIAM...
SHARE
siam 2021 Conference on Computational Science and Engineering
intel
Intel Exhibit Hall
Auditorium
Info
Poster Hall
Coffee Break Gather Space

A Look at Software-Focused Topics at SIAM CSE21

PUBLISHED MAR 26, 2021 AUTHOR BSSW COMMUNITY TOPICS BETTER COLLABORATION ↳ CONFERENCES AND WORKSHOPS

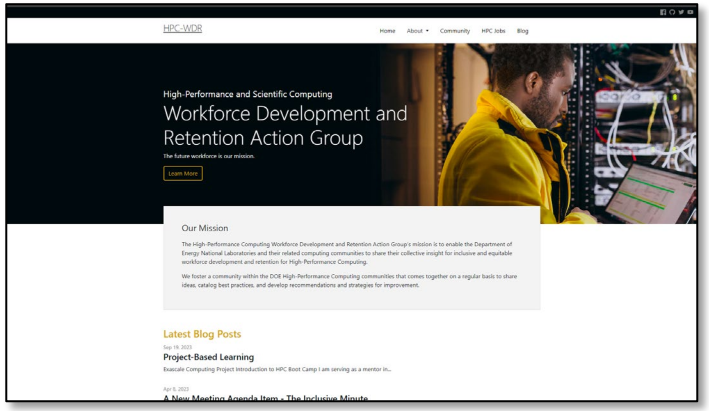
The 2021 SIAM Conference on Computational Science and Engineering (CSE21) took place in early March 2021. The meeting was notable for being the first CSE conference held exclusively online—and for hosting the largest number of registered participants in the history of the conference. CSE21 also hosted the largest array of software-focused sessions the conference has seen to date, with contributions from more than 60 different presenters.

The SIAM Computational Science and Engineering conference series has always been a good place to

IDEAS productivity

Promoting Culture Change

- In all of our work, IDEAS promotes a change in the culture around scientific software
  - Recognizing and valuing the software professionals who are increasingly contributing to the development and maintenance of research software
  - Recognizing and valuing the software itself as a first-class product of compute-intensive research
- Engagement with sponsors
- Support for the developing research software engineer (RSE) community
  - Many IDEAS team members are also members of, and leaders in, US-RSE
- HPC Workforce Development and Retention Action Group
  - Part of the ECP Broadening Participation Initiative to expand the pipeline and workforce for DOE high-performance computing
  - HPC-WDR website and webinar series
    - Topics have included ally skills, diversifying computing, mentoring, and normalizing inclusion by embracing difference



High-Performance and Scientific Computing

Workforce Development and Retention Action Group

The future workforce is our mission.

Our Mission

The High-Performance Computing Workforce Development and Retention Action Group's mission is to enable the Department of Energy National Laboratories and their related computing communities to share their collective insight for inclusive and equitable workforce development and retention for High-Performance Computing.

We foster a community within the DOE High-Performance Computing communities that comes together on a regular basis to share ideas, catalog best practices, and develop recommendations and strategies for improvement.

Latest Blog Posts

Sep 19, 2023

Project-Based Learning

Exascale Computing Project Introduction to HPC Boot Camp I am serving as a mentor in...

Apr 8, 2023

A New Meeting Agenda Item - The Inclusive Minute

Lead: Suzanne Parete-Koon (ORNL)

IDEAS-ECP Impacts

- Feedback underscores IDEAS's role in **enhancing software quality, promoting best practices, and expanding awareness** of the importance of software development
- Curating best practices for software development and team productivity has **empowered teams** to build new practices into their workflows and increase cross-project collaboration
  - Many community members express a desire for additional resources
- Software communities have proven to be a **source of inspiration** for building shared foundations for software ecosystems while respecting team autonomy
- IDEAS outreach mechanisms have enabled innovators in scientific software practices to **share knowledge** with the community
- *Model for other multi-institutional software ecosystems*

Moving Forward

We believe that IDEAS has been an important catalyst for the scientific software community in ECP and beyond, but two further elements are essential for continued qualitative growth...

- Increasing focus on **research software science**
  - Applying our experience with hypothesis-driven science to improve our understanding of how scientific software is developed and used
  - Social-, cognitive-, hard-science, and engineering viewpoints needed
- Changing the prevailing attitude that scientific software productivity, sustainability, and trustworthiness are not just “nice to have” but **“must-have”**
  - Innovators and early adopters are there
  - But many still prioritize more scientific results at the expense of beneficial investments in the software behind them

doi:10.1109/MCSE.2023.3260475

THEME ARTICLE: THE FUTURE OF RESEARCH SOFTWARE ENGINEERS IN THE UNITED STATES

Research Software Science: Expanding the Impact of Research Software Engineering

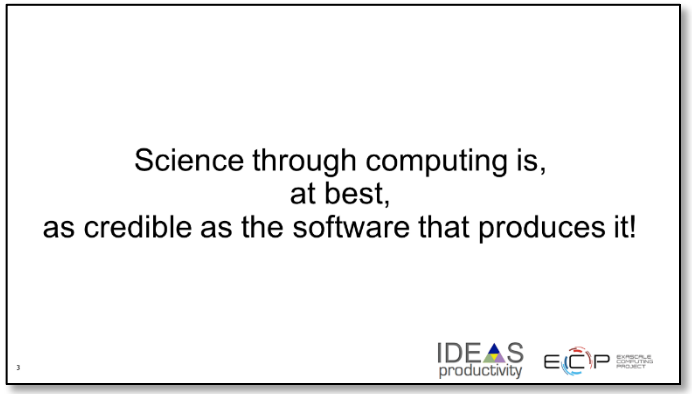
Michael A. Heroux, St. John's University, Collegeville, MN, 56321, USA

BASIC RESEARCH NEEDS IN

The Science of Scientific Software Development and Use

Investment in Software is Investment in Science

doi:10.2172/1846009



Science through computing is,
at best,
as credible as the software that produces it!

3

A key message of the BSSw tutorials

What’s Next?

- As part of the ECP project, IDEAS-ECP ends at the end of 2023
  - IDEAS-Watersheds continues, with focus on watershed modeling
- **BSSw Fellowship** class of 2024 to be announced soon
  - Thanks to support from DOE/ASCR, DOE/NNSA, and NSF
- **Next Generation Scientific Software Technologies (NGSST)** initiative is emerging to help address scientific software stewardship needs across the ASCR software ecosystem
  - Focus is different than IDEAS-ECP
  - But you can see IDEAS influence in some of the planned work
- Looking for increased engagement from the community to continue some IDEAS legacy activities

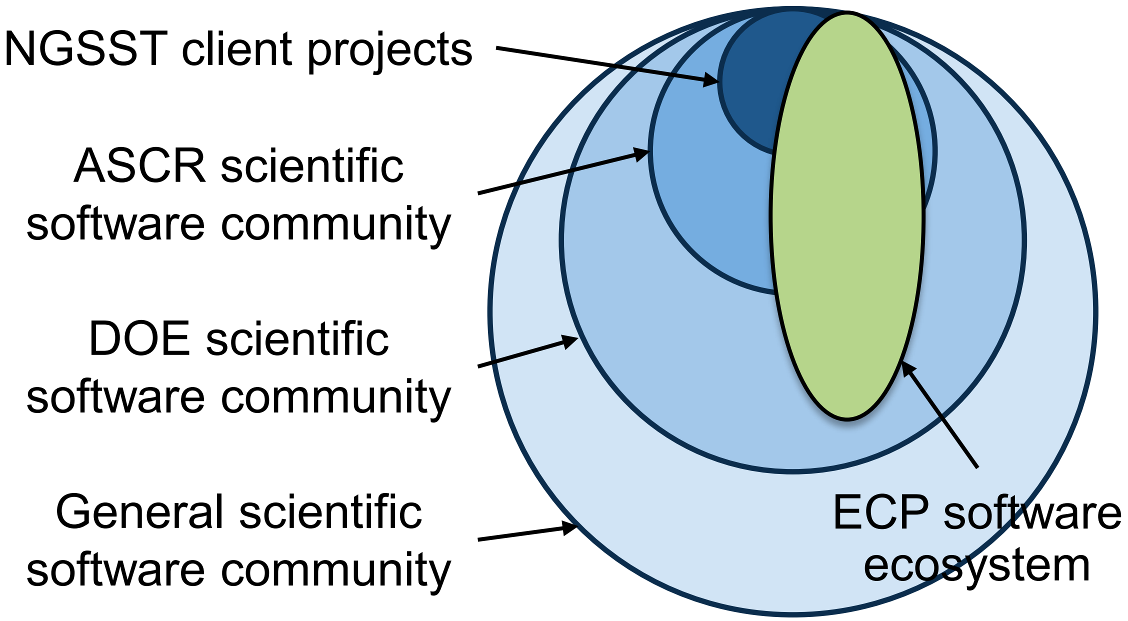
Acronyms

NNSA	= National Nuclear Security Administration
NSF	= National Science Foundation

Next Generation Scientific Software Technologies (NGSST)

- Seven teams submitted interlocking proposals for a **Scientific Software Stewardship Consortium (S3C)** with distinct but complementary foci
- Focus on the ASCR scientific software ecosystem, which includes software products from
  - Exascale Computing Project (ECP) Software Technologies
  - ECP Co-Design Centers
  - Other ASCR-supported products
- At anticipated funding levels, the NGSST will be able to support only a subset of the eligible projects
- Phase 2 proposals under review, most awards expected to start in January 2024

Scientific Software Ecosystems of Interest



NGSST client projects
ASCR scientific software community
DOE scientific software community
General scientific software community
ECP software ecosystem

Continuing the IDEAS Legacy in NGSST

- Fostering software communities
  - Extreme-Scale Scientific Software Stack (E4S)
  - Some of the other community-focused NGSST organizations may pursue SDKs
- Incubating and curating methodologies and resources
  - Software quality assurance (emphasis on ecosystem integration)
  - And possibly in other areas
  - BSSw.io resource portal
- Disseminating knowledge
  - Best Practices for HPC Software Developers webinar series
  - Better Scientific Software tutorials
    - Complemented by trainings from other NGSST organizations
- Promoting culture change
  - HPC Workforce Development and Retention
  - Building a community of practice for research software engineers in the DOE national labs

Items in green are specific IDEAS activities that we plan to continue in NGSST – with additional help from the larger community!

You Can Help Change the Culture around Scientific Software!

- IDEAS (or NGSST) can't do it alone!
- If you're a developer or user of scientific software (or a manager), **you have a role to play in making scientific software better**
  - Be **thoughtful about the stewardship** of your own software
  - Work with your team to **learn about and implement better software development practices**
    - Focus on incremental, but continual software process improvement, tailored to your needs
- **Share** your software development knowledge, experience, and resources with others
- **Engage with communities** relevant to your work and interests (see also doi:10.1109/MCSE.2018.2883051)
- **Talk with your sponsors** about the importance of software stewardship, what you're doing, the benefits, and how they can help
- Remember: **Science through computing is, at best, as credible as the software that produces it!**

Acknowledgements: The IDEAS-ECP Team and the ECP

Ann Almgren (LBNL)

Ross Bartlett (SNL)

David E. Bernholdt (ORNL)
Institutional PI

Kita Cranfill (ORNL)

Anshu Dubey (ANL)
Institutional PI

Don Frederick (LLNL)

William Godoy (ORNL)

Elsa Gonsiorowski (LLNL)
Institutional PI

Patricia Grubel (LANL)

Rinku Gupta (ANL)

Rebecca Hartman-Baker (LBNL)
Computing Facility Liaison

Michael Heroux (SNL)
Lead Co-PI

Axel Huebl (LBNL)

Megan Phinney (LANL)
Computing Facility Liaison

Rose Lynch (ANL)

Addi Malviya Thakur (ORNL)

Osni Marques (LBNL)
Institutional PI

Lois Curfman McInnes (ANL)
Lead Co-PI

Reed Milewicz (SNL)

Mark Miller (LLNL)

J. David Moulton (LANL)
Institutional PI

Miranda Mundt (SNL)

Hai Ah Nam (LBNL)

Boyana Norris (U. Oregon)
Institutional PI

Erik Palmer (LBNL)

Suzanne Parete-Koon (ORNL)
ECP Workforce Development and Retention Action Group Lead

Elaine Raybourn (SNL)
Institutional PI

Katherine Riley (ANL)
Computing Facility Liaison

David Rogers (ORNL)
Computing Facility Liaison

Ben Sims (LANL)

Deborah Stevens (ANL)

Greg Watson (ORNL)

Jim Willenbring (SNL)

Alumni: Satish Balay (ANL), Lisa Childers (ANL), Todd Gamblin (LLNL), Judy Hill (ORNL), Steve Hudson (ANL), Christoph Junghans (LANL), Alicia Klinvex (SNL), Shannon Lindgren (ANL), Lena Lopatina (LANL), Jared O'Neal (ANL), Michele Rosso (LBNL), Jean Shuler (LLNL), Barry Smith (ANL), Louis Vernon (LANL), Paul Wolfenbarger (SNL)