Toward a Leadership Software Center (LSC)

Transforming Science R&D into World-class Leadership Software

Disclaimer

- The following description is notional
- We have shared it with stakeholders
- Goal: Be ready to execute a plan for post-ECP sustainability

Background

- The US DOE Exascale Computing Project (ECP) initiated the Extreme-scale Scientific Software Stack (E4S)
- E4S development will continue under ECP for two more years
- A brief reminder of E4S and SDKs

Extreme-scale Scientific Software Stack (E4S)

- E4S: HPC software ecosystem – a curated software portfolio
- A **Spack-based** distribution of software tested for interoperability and portability to multiple architectures
- Available from **source**, **containers**, **cloud, binary caches**
- Leverages and enhances SDK interoperability thrust
- Not a commercial product – an open resource for all
- Growing functionality: Nov 2021: E4S 21.11 – 91 full release products

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
Post-ECP Strategy LSSw, ASCR Task Force	

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 delivers products needed now and into the future

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

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

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

Delivering an Open, Hierarchical Software Ecosystem

Levels of Integration | Product | Source and Delivery

ECP ST Open Product Integration Architecture

- Build all SDKs
- Build complete stack
- Containerize binaries

E4S

Source: ECP E4S team; Non-ECP Products (all dependencies)
Delivery: spack install e4s; containers; CI Testing

- Group similar products
- Make interoperable
- Assure policy compliant
- Include external products

SDKs

Source: SDK teams; Non-ECP teams (policy compliant, spackified)
Delivery: Apps directly; spack install sdk; future: vendor/facility

- Standard workflow
- Existed before ECP

ST Products

Source: ECP L4 teams; Non-ECP Developers; Standards Groups
Delivery: Apps directly; spack; vendor stack; facility stack

ECP ST Individual Products

E4S and SDKs as Platforms are providing tremendous value

Activity	SDKs	E4S
Planning	Transparent and collaborative requirements, analysis and design, delivery	Campaign-based portfolio planning coordinated with Facilities, vendors, community ecosystem, non-DOE partners
Implementation	Leverage shared knowledge, infrastructure, best practices	ID and assist product teams with cross-cutting issues
Cultivating Community	Within a specific technical domain: Portability layers, LLVM coordination, sparse solvers, etc.	Across delivery and deployment, with software teams, facilities' staff
Resolving issues, sharing solutions	Performance bottlenecks and tricks, coordinated packaging and use of substrate, e.g., Desul for RAJA and Kokkos	Build system bugs and enhancements, protocols for triage, tracking & resolution, leverage across & beyond DOE
Improving quality	Shared practice improvement, domain-specific quality policies, reduced incidental differences and redundancies, per-commit CI testing	Portfolio-wide quality policies, documentation portal, portfolio testing on many platforms not available to developers
Path-finding	Exploration and development of leading-edge computational tools that provide capabilities and guidance for others	Exploration and development of leading-edge packaging and distribution tools and workflows that provide capabilities and guidance for others
Training	Collaborative content creation and curation, coordinated training events for domain users, deep, problem-focused solutions using multiple products	Portfolio installation and use, set up of build caches, turnkey and portable installations, container and cloud instances
Developer experience	Increased community interaction, increased overhead (some devs question value), improved R&D exploration	Low-cost product visibility via doc portal, wide distribution via E4S as from-source/pre-installed/container environment
User experience	Improve multi-product use, better APIs through improved design, easier understanding of what to use when	Rapid access to latest stable feature sets, installation on almost any HPC system, leadership to laptop
Scientific Software R&D	Shared knowledge of new algorithmic advances, licensing, build tools, and more	Programmatic cultivation of scientific software R&D not possible at smaller scales
Community development	Attractive and collaborative community that attracts junior members to join	Programmatic cultivation of community through outreach and funded opportunities that expand the membership possibilities

Moving Forward

- To better ensure continued growth and sustainability beyond ECP, we are exploring ideas now to better orient E4S efforts toward the post-ECP era
- Engaging key US agencies and international institutions is essential to the longevity of E4S
- We propose a plan for
  - A DOE ASCR **Leadership*** Software Center (LSC)
  - A leadership and stewardship role in sustaining and growing E4S through LSC

*We intend leadership in our setting to mean emerging and leading-edge software for emerging and leading-edge scientific computing environments, including HPC, AI/ML for science, large-scale edge computing for science, quantum, and other scientific computing software products that complement industry efforts and facilitate scientific progress.

Mission

- Deliver a robust, reliable, high-quality and sustainable software stack
- that enables the rapid development of DOE scientific applications
- for the pursuit of scientific discovery
- in leading-edge computing environments

Vision

- Build the best leadership computing scientific software stack in the world
- that enables innovative computational and data science solutions
- to global challenges and breakthrough problems
- on leadership computing platforms

Values

We value software that

- addresses leading edge and emerging application needs,
- on leading edge and emerging computing platforms,
- and unambiguously adds value to the software ecosystem
- because of its usefulness, quality, sustainability & complementarity

Goals

1. **Establish a sustainable organization** to effectively & efficiently deliver DOE leadership scientific software
2. **Deliver** a portable, high-performance scientific computing development and execution **software stack**
3. **Accelerate** the augmentation of scientific solutions with integrated **AI/ML and advanced workflows**
4. **Curate and deliver** a software ecosystem for emerging **edge workflows** to support large-scale data
5. **Prepare for** software stack requirements to address emerging **quantum computing** technologies
6. **Establish ecosystem partnerships** with other DOE, agency, industry, academic and international institutions
7. **Build a community and workforce** around scientific software development

Requirements

1. **Single Org, distributed R&D teams:** Single LSC organization drawing from all DOE labs, universities, industry
2. **Sustained sponsor and affiliate support:** Clear commitment to attract the best and brightest
3. **Business model for interactions:** How to engage ecosystem partners: ASCR Research, SciDAC, Facilities, industry
4. **Domain leaders:** Thought leaders in each supported technical domain
5. **Strategic platform and problem information access:** Information on computing and scientific futures
6. **Distributed collaboration environment:** Information platforms, tools, processes and support staff

LSC Motivation

- ECP makes a compelling case for coordinated development and delivery of DOE software products
  - **Planning:** Portfolio of inter-related capabilities in collaboration with application teams, facilities, vendors, open-source communities
  - **Execution:** Development and dissemination of best practices; use of shared platforms (e.g., Atlassian tools), testing infrastructure, effective and efficient processes
  - **Tracking:** Coordinated and transparent progress tracking, adaptation to evolving requirements
  - **Assessment:** Regular assessment and reporting of progress to stakeholders and community
- The ECP ST Portfolio approach promises improved effectiveness and efficiency of DOE software efforts vs independent software teams working alone
- The E4S/SDK open software architecture provides a framework for successful software development and delivery
- ECP has fostered a holistic approach to scientific software workforce development
- **A Leadership Software Center (LSC) provides a compelling approach as an enabler to coordinate the development and delivery of DOE software products after the end of ECP**

ECP Experience: Software sustainability requires a new kind of organization

Leadership Software Center (LSC): Will enable the sustainability of ECP contributions, and development and delivery of future capabilities, including new domains like AI/ML, Edge and Quantum

Tailored Agile: The LSC will use tailored project management practices, processes, tools, and a distributed multi-institutional organization to enable effective and efficient delivery of ASCR software investments

New Ecosystem Entity: The LSC will establish an essential and new ecosystem entity to complement Facilities, ASCR Research, vendors, industry and other entities.

Workforce Development: Establishing the LSC assures the creation of a scientific software workforce for sustainable leadership scientific software development and delivery

Key Elements of the LSC Structure

Full stack: E4S, future variations

LSC Project Leadership

- Project Director/Deputy (2 FTE) : Leads overall project, coordinates with DOE, Tech area leads, other stakeholders
- Project Controls & Admin (2 FTE): Manages budgets, funds, costs. Supports communication, processes & tools
- Overall responsibility for coordination and delivery of E4S (3 FTE)
- Conduit to vendor, facility stacks, open community platforms, standards (2 FTE)
- IDEAS/BSSw team (3 FTE)
- Training/Outreach (content curation, fellows) coordinated with Facilities (2 FTE)

Example SDK: Math Libs (xSDK)

Technical area SDKs

- Initial areas: Programming models & runtimes, development tools, math libs, data, viz, SW packaging & delivery
- Upcoming: ML/AI, Edge, Quantum
- Technical area SDK lead: Provides technical planning and oversight for the area as an aggregate
- One lead and deputy for each SDK collection of capabilities (1 FTE total per area)
- SDK leaders chosen for domain expertise and for broad institutional representation

Example product: OpenMPI

Product development teams

- PI plus team (1 – 3 FTE per team).
- Team members include:
  - Domain expert developers
  - Project coordinator (shared across multiple product teams) to manage processes & deliverables
  - Career software professionals, research software engineers (RSEs)

Leadership Software Center Cadence

Ongoing + Campaigns

LSC Core

Initiate → Refine → Sustain

Leadership Software Campaigns

LSC-1 (2024-26) → LSC-2 (2027-2029) → LSC-3 (2030-32)

LSC-1 (2024-26)

- Next phase core SW*
- Establish AI/ML SDK
- Scope Edge SDK
- Contingency

LSC-2 (2027-2029)

- Next phase core SW
- Next phase AI/ML
- Establish Edge SDK
- Scope Quantum SDK
- Contingency

LSC-3 (2030-32)

- Next phase core SW
- Next phase AI/ML
- Next phase Edge
- Establish Quantum SDK
- Contingency

*Next phase core SW: Scope necessary to address emerging needs in programming models, runtimes, tools, math libs, data, visualization, workflows and other established software technologies. Often this scope will be new features in existing LSC products, such as AI linear algebra features being added, or support for the latest AI devices, or both, to our existing Math Libs SDK.

LSC Execution Approach

Plan, Execute, Track, Assess Lifecycle

- All activities governed by phased development process
- Executed as “campaigns”: LSC-1, LSC-2, …
- Tailored agile approach
- Hierarchical approach:
  - Multi-year baseline as campaign
  - Refine annually
  - Add fidelity per milestone at “last responsible moment”

Change Management Process:

- Changes from campaign base plan managed by a process
- Any changes to cost, scope and schedule
- Explicit review process determined by degree of change
- Change control process assures lightweight transparency
- Objective: **Always do most important work at any time**

Capability Integration Strategy

DOE software products have four primary integration targets:

- **Vendors:** Specific HPC enhancements, integrated into system vendor stacks
- **Community SW:** C++, Fortran, LLVM
- **Facilities:** Tuned open-source SW for key platforms
- **Direct to apps:** Application teams download and build
- **Note:** Some products are available via 2 – 3 of the above targets

LSC goals:

- **Establish and ensure quality standards** for LSC product development and delivery
- Assure that funded projects **develop and deliver** to one or more integration targets
- **Track and assess** integration status of new capabilities

Building new infrastructure and business models toward a sustainable software ecosystem

Leadership Software Center (LSC): Some business needs overlap with Leadership Computing Facilities, but not all

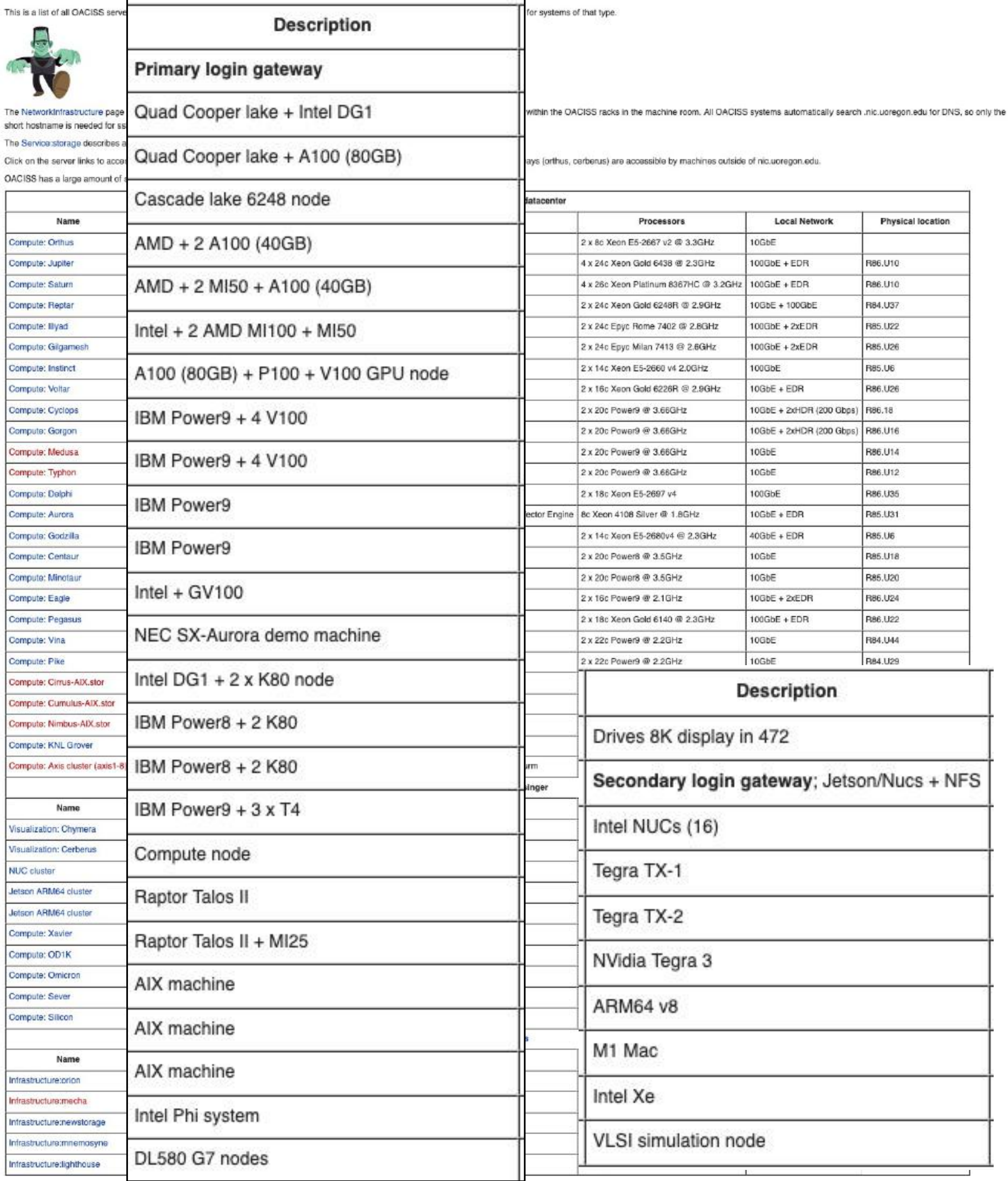
Preparing E4S products requires additional early-access resources: Frontier and Aurora early-access systems are essential, but E4S teams need more

E4S products require tiered support: Beyond tier-1 support at the facilities, users need issues addressed by staff funded to do the support in collaboration with product dev teams

Partnering with commercial software companies for E4S support enables cost and benefit sharing: Cost of E4S support is shared, opens possibility for industry use of E4S because they can pay for support

Frank – Designed for Libs & Tools Developers

- Prep system for ECP libs & tools
- Access to latest non-NDA HW/SW
- Shared file system – 1 copy of SW
- Port to many device types at once
- Porting support from E4S team
- CI testing workhorse (500K builds)
- Next: Bare metal, BIOS-changing support for low-level software work



This is a list of all OACISS serve… for systems of that type.

The NetworkInfrastructure page … within the OACISS racks in the machine room. All OACISS systems automatically search .nic.uoregon.edu for DNS, so only the short hostname is needed for ss…

The Service:storage describes a…

Click on the server links to acce… ays (orthus, cerberus) are accessible by machines outside of nic.uoregon.edu.

OACISS has a large amount of …

Description
Primary login gateway
Quad Cooper lake + Intel DG1
Quad Cooper lake + A100 (80GB)
Cascade lake 6248 node
AMD + 2 A100 (40GB)
AMD + 2 MI50 + A100 (40GB)
Intel + 2 AMD MI100 + MI50
A100 (80GB) + P100 + V100 GPU node
IBM Power9 + 4 V100
IBM Power9 + 4 V100
IBM Power9
IBM Power9
Intel + GV100
NEC SX-Aurora demo machine
Intel DG1 + 2 x K80 node
IBM Power8 + 2 K80
IBM Power8 + 2 K80
IBM Power9 + 3 x T4
Compute node
Raptor Talos II
Raptor Talos II + MI25
AIX machine
AIX machine
AIX machine
Intel Phi system
DL580 G7 nodes

Name	Processors	Local Network	Physical location
Compute: Orthus	2 x 8c Xeon E5-2667 v2 @ 3.3GHz	10GbE	
Compute: Jupiter	4 x 24c Xeon Gold 6438 @ 2.3GHz	100GbE + EDR	R86.U10
Compute: Saturn	4 x 26c Xeon Platinum 8367HC @ 3.2GHz	100GbE + EDR	R86.U10
Compute: Reptar	2 x 24c Xeon Gold 6248R @ 2.9GHz	10GbE + 100GbE	R84.U37
Compute: Illyad	2 x 24c Epyc Rome 7402 @ 2.8GHz	100GbE + 2xEDR	R85.U22
Compute: Gilgamesh	2 x 24c Epyc Milan 7413 @ 2.6GHz	100GbE + 2xEDR	R85.U26
Compute: Instinct	2 x 14c Xeon E5-2660 v4 2.0GHz	100GbE	R85.U6
Compute: Voltar	2 x 16c Xeon Gold 6226R @ 2.9GHz	10GbE + EDR	R86.U26
Compute: Cyclops	2 x 20c Power9 @ 3.66GHz	10GbE + 2xHDR (200 Gbps)	R86.18
Compute: Gorgon	2 x 20c Power9 @ 3.66GHz	10GbE + 2xHDR (200 Gbps)	R86.U16
Compute: Medusa	2 x 20c Power9 @ 3.66GHz	10GbE	R86.U14
Compute: Typhon	2 x 20c Power9 @ 3.66GHz	10GbE	R86.U12
Compute: Delphi	2 x 18c Xeon E5-2697 v4	100GbE	R86.U35
Compute: Aurora	8c Xeon 4108 Silver @ 1.8GHz	10GbE + EDR	R85.U31
Compute: Godzilla	2 x 14c Xeon E5-2680v4 @ 2.3GHz	40GbE + EDR	R85.U6
Compute: Centaur	2 x 20c Power8 @ 3.5GHz	10GbE	R85.U18
Compute: Minotaur	2 x 20c Power8 @ 3.5GHz	10GbE	R85.U20
Compute: Eagle	2 x 16c Power9 @ 2.1GHz	10GbE + 2xEDR	R86.U24
Compute: Pegasus	2 x 18c Xeon Gold 6140 @ 2.3GHz	100GbE + EDR	R86.U22
Compute: Vina	2 x 22c Power9 @ 2.2GHz	10GbE	R84.U44
Compute: Pike	2 x 22c Power9 @ 2.2GHz	10GbE	R84.U29
Compute: Cirrus-AIX.stor			
Compute: Cumulus-AIX.stor			
Compute: Nimbus-AIX.stor			
Compute: KNL Grover			
Compute: Axis cluster (axis1-8)			

Name
Visualization: Chymera
Visualization: Cerberus
NUC cluster
Jetson ARM64 cluster
Jetson ARM64 cluster
Compute: Xavier
Compute: OD1K
Compute: Omicron
Compute: Sever
Compute: Silicon

Description
Drives 8K display in 472
Secondary login gateway; Jetson/Nucs + NFS
Intel NUCs (16)
Tegra TX-1
Tegra TX-2
NVidia Tegra 3
ARM64 v8
M1 Mac
Intel Xe
VLSI simulation node

Name
Infrastructure:orion
Infrastructure:mecha
Infrastructure:newstorage
Infrastructure:mnemosyne
Infrastructure:lighthouse

Software Support Challenges

Support for DOE libraries & tools has been challenging in the past:

- Support funding is a tax on incoming R&D funds
- External user support ad hoc and cautious: Chicken vs pig in ham & eggs breakfast

Impact: Uptake is risky

- Support is contingent on
  - Continued R&D funding (DOE users)
  - Minimal incompatibility with DOE user needs (external users)
- App teams (even DOE) reluctant to adopt – concerned about long-term support

Commercial Software Support Partnership

Strategy: Commercial partnerships

- Fund external commercial support org
- Provide tier-2 facilities user support
- Triage E4S bugs
- Work with product teams to fix issues

Commercial partner leverage

- Commercial E4S expertise means industry, agencies can purchase support
- Creates broader E4S user base
- Spreads support costs
- Enables industry and agencies to adopt leading edge software more easily

Working toward software sustainability: ASCR Task force, community events

Leadership Scientific Software (LSSw) Portal

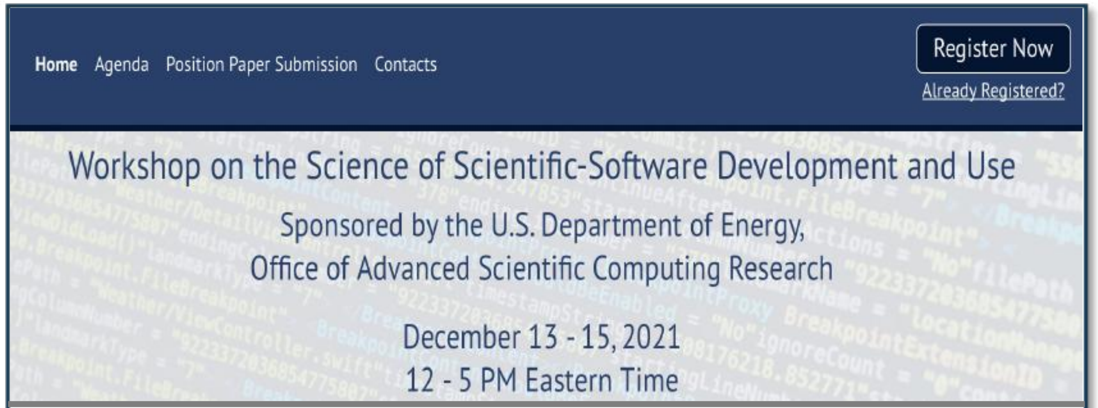
https://lssw.io

The LSSw portal is dedicated to building community and understanding around the development and sustainable delivery of leadership scientific software

- LSSw Town Hall Meetings (ongoing)
  - 3rd Thursday each month, 3 – 4:30 pm Eastern US time
  - 100+ attendees at each meeting, sessions recorded
- Town Hall Topics
  - Meeting 1: Overview of the ECP Software Technology Focus Area
  - Meeting 2: Progress, impediments, priorities & gaps in LSSw panel
  - Meeting 3: Expanding the LSSw User Communities - Panel
  - Meeting 4: Expanding the LSSw Developer Communities - Panel
  - Meeting 5: Other HPC Software Ecosystems - Panel
- Portal: Slack, Whitepapers & References: TBA

Workshop on Research Software Science

Software is an increasingly important component in the pursuit of scientific discovery. Both its development and use are essential activities for many scientific teams. At the same time, very little scientific study has been conducted to understand, characterize, and improve the development and use of software for science.



- 126 whitepaper submissions (at same time as RFI!)
- Whitepapers at: https://www.orau.gov/SSSDU2021
- 150 registrants (max), 100 registered observers
- 150 – 170 active participants across three days
- Goal: Leverage new skills applied to scientific SW

EXASCALE COMPUTING PROJECT

ASCR RFI Response

- ASCR RFI – Broad call, Dec 13, 2021 deadline
- 10 prompt sets:
  - 1 Software Dependencies and Requirements
  - 2 Security and Integrity of Software and Data
  - 3 Software Development Infrastructure Requirements
  - 4 Developing and Maintaining Community Software
  - 5 Building a Diverse Workforce and Inclusive Environment
  - 6 Technology Transfer and Building Software Communities
  - 7 The Scope of Software Stewardship
  - 8 Stewardship Management and Oversight
  - 9 Assessment for Success
  - 10 Additional Topics
- 37 community responses – From brief and long
- ECP Response:
  - 48 pages, addressing all prompts
  - Foundation for 2021 IPR Response

ECP EXASCALE COMPUTING PROJECT

A Response to the "Stewardship of Software for Scientific and High-Performance Computing" Request for Information

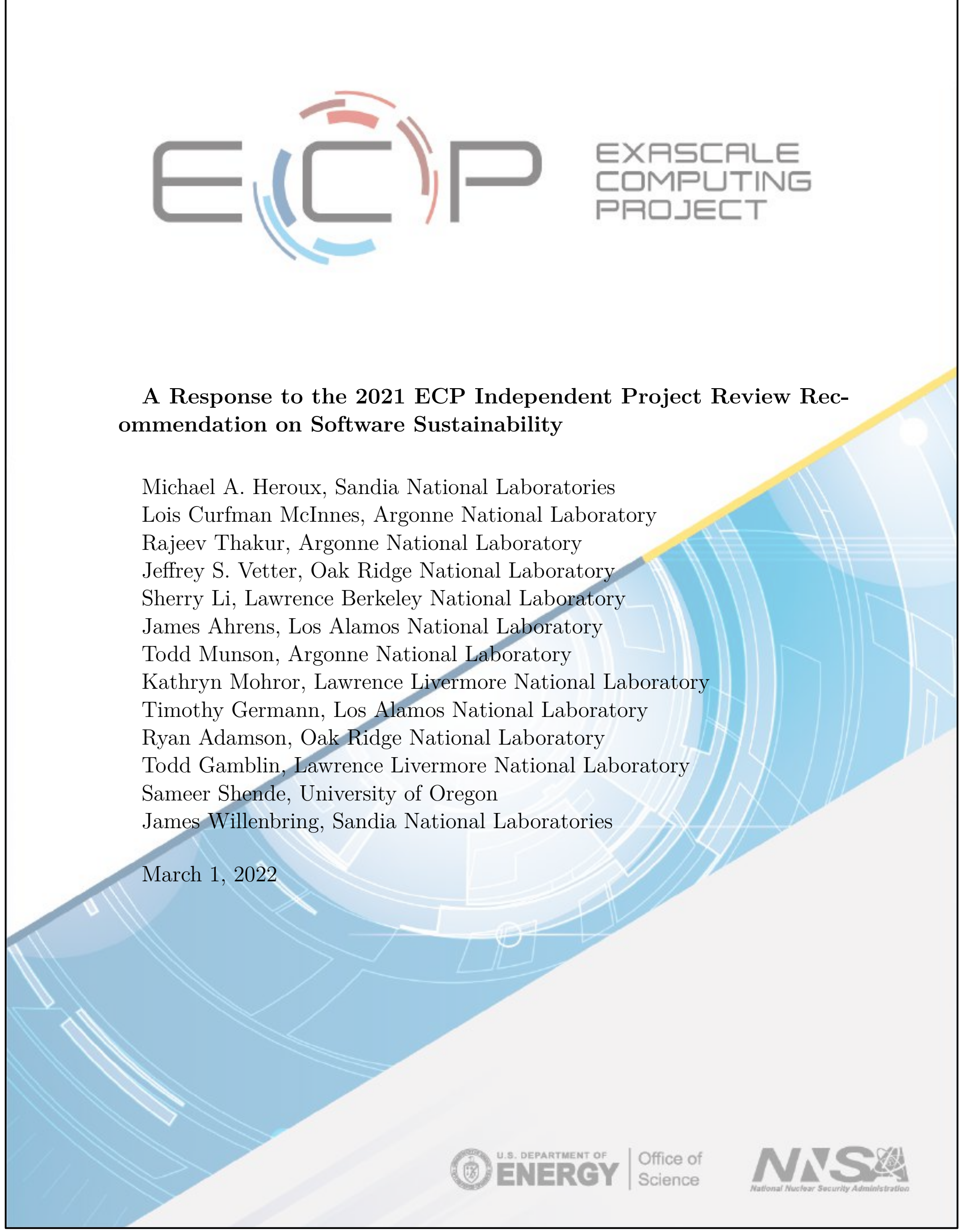
Michael A. Heroux, Sandia National Laboratories
Lois Curfman McInnes, Argonne National Laboratory
Rajeev Thakur, Argonne National Laboratory
Jeffrey S. Vetter, Oak Ridge National Laboratory
Sherry Li, Lawrence Berkeley National Laboratory
James Ahrens, Los Alamos National Laboratory
Todd Munson, Argonne National Laboratory
Kathryn Mohror, Lawrence Livermore National Laboratory
Timothy Germann, Los Alamos National Laboratory
Ryan Adamson, Oak Ridge National Laboratory
Todd Gamblin, Lawrence Livermore National Laboratory
Sameer Shende, University of Oregon
James Willenbring, Sandia National Laboratories

December 13, 2021

U.S. DEPARTMENT OF ENERGY | Office of Science
NNSA National Nuclear Security Administration

2021 IPR Recommendation Response

- Revision of RFI response plus:
- A proposed transition timeline and rationale document for the relevant timeline scenarios
- Public release of the dependency database, or some releasable subset
- Release date: March 1, 2022



EXASCALE COMPUTING PROJECT

A Response to the 2021 ECP Independent Project Review Recommendation on Software Sustainability

Michael A. Heroux, Sandia National Laboratories
Lois Curfman McInnes, Argonne National Laboratory
Rajeev Thakur, Argonne National Laboratory
Jeffrey S. Vetter, Oak Ridge National Laboratory
Sherry Li, Lawrence Berkeley National Laboratory
James Ahrens, Los Alamos National Laboratory
Todd Munson, Argonne National Laboratory
Kathryn Mohror, Lawrence Livermore National Laboratory
Timothy Germann, Los Alamos National Laboratory
Ryan Adamson, Oak Ridge National Laboratory
Todd Gamblin, Lawrence Livermore National Laboratory
Sameer Shende, University of Oregon
James Willenbring, Sandia National Laboratories

March 1, 2022

U.S. DEPARTMENT OF ENERGY | Office of Science
NNSA National Nuclear Security Administration

Questions?

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

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

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