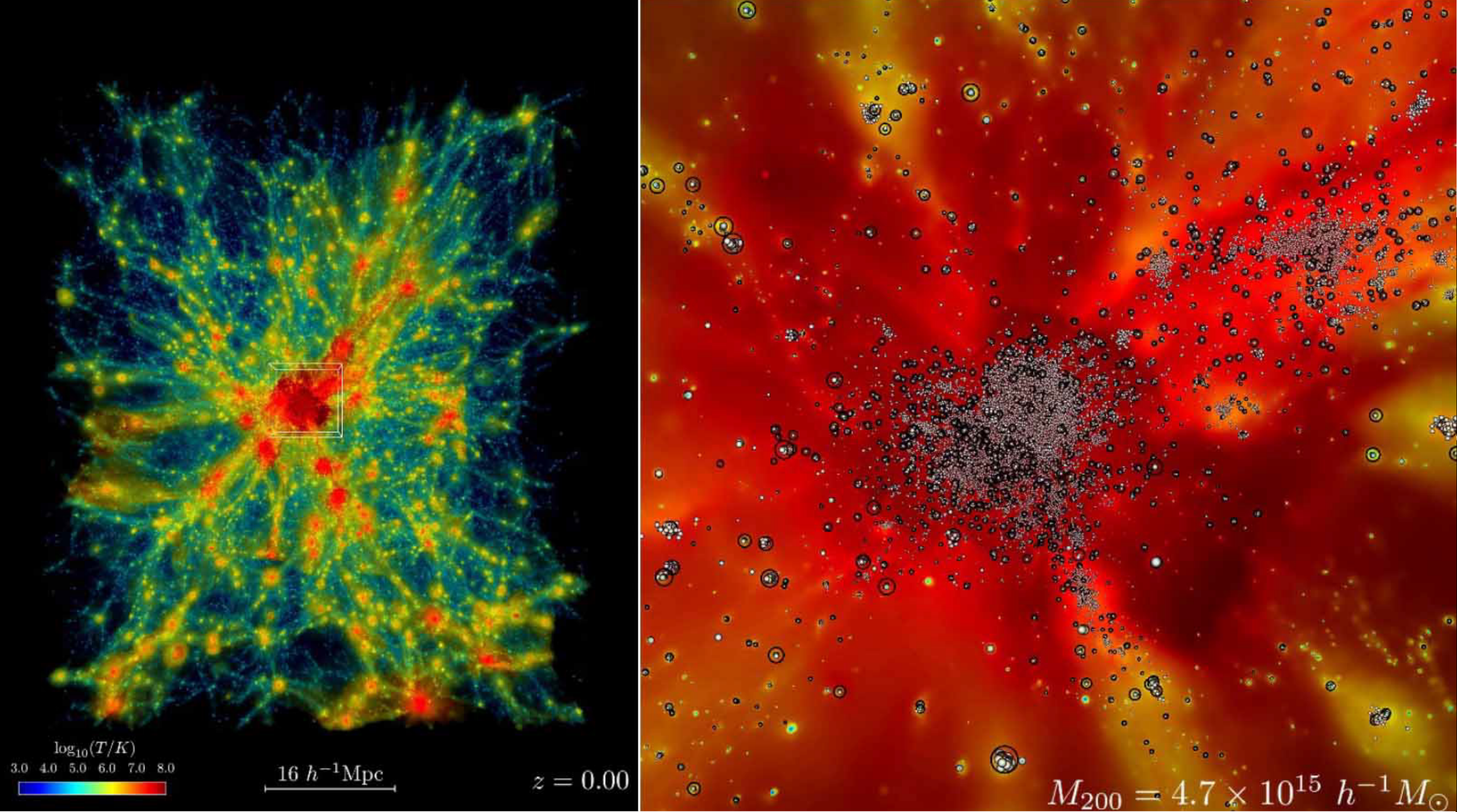


```
        int leafj = __ldg(&leafs2[blockIdx.x]);
#ifdef KSK_RECALC_CORRECTIONS
        int recalcX = __ldg(&recalc[leafi]); int recalcY = __ldg(&recalc[leafj]);
        if(useRecalc && !recalcX && !recalcY)return;//skip any pair where either side does not need to be recalculated
#endif
        int leafXcnt = __ldg(&leafCount[leafi]); int leafYcnt = __ldg(&leafCount[leafj]);
        int leafXcntDM = __ldg(&leafCountDM[leafi]); int leafYcntDM = __ldg(&leafCountDM[leafj]);
        int leafXoff = __ldg(&leafOffset[leafi]); int leafYoff = __ldg(&leafOffset[leafj]);
        int actvi = __ldg(&activity[leafi]); int actvj = __ldg(&activity[leafj]);
        int upi = (actvi == IS_ACTIVE);
        int upj = (actvj == IS_ACTIVE);
#ifdef UPDATE_DEAD_LEAF
        int upDeadi = __ldg(&upDead[leafi]);  int upDeadj = __ldg(&upDead[leafj]);
        upi |= (actvi == IS_DEAD && upDeadi);
        upj |= (actvj == IS_DEAD && upDeadj);
#endif
#ifndef HYBRID_SKIP_PASSIVE
```

A Report by the ASCR Facilities Software Task Force

POST EXASCALE SOFTWARE
in the ASCR Facilities Ecosystem

July 2025

ANL/ALCF-25/2

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free at OSTI.GOV (http://www.osti.gov/), a service of the US Dept. of Energy's Office of Scientific and Technical Information.

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

A Report by the ASCR Facilities Software Task Force

POST EXASCALE SOFTWARE in the ASCR Facilities Ecosystem

Task Force Chair

Saswata Hier-Majumder, ORCID ID 0000-0002-2629-1729
Program Manager, Advanced Scientific Computing Research, Office of Science
US Department of Energy

Task Force Members

Wahid Bhimji, ORCID ID 0000-0002-6213-8617
Group Lead, Data and Analytics Services, National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

Brandon Cook, ORCID ID 0000-0002-4203-4079
Group Lead, Programming Environments and Models, National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

Graham Heyes, ORCID ID 0000-0001-9902-8190
Technical Director, High Performance Data Facility
Thomas Jefferson National Accelerator Facility

Kalyan Kumaran, ORCID ID 0000-0002-6447-3195
Director of Technology, Argonne Leadership Computing Facility
Argonne National Laboratory

John MacAuley, ORCID ID 0000-0003-1277-1032
Principal Architect, Energy Sciences Network
Lawrence Berkeley National Laboratory

Bronson Messer, ORCID ID 0000-0002-5358-5415
Director of Science, Oak Ridge Leadership Computing Facility
Oak Ridge National Laboratory

Philip Roth, ORCID ID 0000-0001-9583-1103
Group Leader, Algorithms and Performance Analysis, National Center for Computational Sciences
Oak Ridge National Laboratory

Jiachuan Tian, ORCID ID 0009-0001-2227-4673
Computer Systems Engineer, Energy Sciences Network
Lawrence Berkeley National Laboratory

Brice Videau, ORCID ID 0000-0001-6824-1263
Performance Engineering Team Lead, Argonne Leadership Computing Facility
Argonne National Laboratory

Thanks to ALCF Communications for help with report editing and design.

On the cover: A small sample from the Frontier simulations, generated with the cosmological N-body simulation code HACC, reveals the evolution of the expanding universe in a region containing a massive cluster of galaxies from billions of years ago to present day (left). Red areas show hotter gasses, with temperatures reaching 100 million Kelvin or more. Zooming in (right), star tracer particles track the formation of galaxies and their movement over time. An excerpt of the HACC code is presented (bottom).
Credit: Argonne National Laboratory, U.S Dept of Energy

Contents

Executive Summary

In September 2023, the ASCR leadership charged the ASCR facilities to create a shared vision and strategy for software in the ASCR facilities ecosystem. With tectonic shifts in computational science workflows, rise of artificial intelligence (AI) based software and hardware and new national level initiatives, the ASCR ecosystem is poised to create long-lasting impacts in the domain of scientific software. The ASCR Facilities Software Task Force—led by ASCR with members from the Argonne Leadership Computing Facility (ALCF), Energy Sciences Network (ESnet), High Performance Data Facility (HPDF) project, National Energy Research Scientific Computing Center (NERSC), and the Oak Ridge Leadership Computing Facility (OLCF)—undertook an effort to identify the existing technological gaps that may prevent the ASCR facilities from achieving their long and short term software-dependent scientific goals, to identify challenges to bridge these gaps, and to recommend a set of actions. To develop the ASCR facilities' strategic vision for software, the task force focused its effort on six areas: system software; programming environments, tools, and libraries (PE); data; artificial intelligence software; integrated research infrastructure (IRI); and stakeholder engagement. These areas were identified by the task force based on scientific needs of ASCR facility users, changes in the global software and hardware landscape, and current and future investments by the Office of Science.

The task force arrived at a vision for *an ASCR facility ecosystem strategy for software stewardship: continue cross-facility collaboration on software development, increase participation of ASCR facilities' personnel in software and data standardization bodies, and strategically invest in shared infrastructure and system software*. This vision will allow the ASCR facilities to maintain their leadership roles, provide seamless integration of new user applications across multiple facilities, develop new performance tools, and deliver mission critical science. The task force further identified crosscutting gaps and recommendations for future ASCR efforts.

The following sections present findings and recommendations for these crosscutting gaps and each of the six key areas mentioned above.

Crosscutting Gaps

Across the different focus areas, five key areas of technological gaps were identified. Most of these technological gaps arise from changes in the scientific workflow, an increase in the use of cloud-based environments, changes in the hardware architecture, and the rise of AI-based scientific tools. These crosscutting technological gaps arise in the following areas:

- **Containers:** Currently, there is no common container image or runtime that spans across ASCR facilities. Users also lack support to create such container images using vendor provided tools.
- **Standard for user and data security:** Currently, an ASCR-wide trust system for security authentication is lacking. Standards are also lacking for defining confidentiality level and securing data across the ASCR ecosystem.

- **Energy use:** Currently, the ASCR ecosystem lacks standards for energy use at the software level. In addition, users are unable to implement energy usage related features in applications during runtime.
- **Cloud software:** A growing number of ASCR HPC users also rely on hyperscalers or other cloud computing resources. Currently, these two alternative computing platforms differ in a number of ways, including the programming environment and tools. This difference results in a reduced quality of user experience and prevents seamless integration of new scientific applications across platforms.
- **Tools for AI:** Performance tools and debuggers for AI on HPC need capabilities beyond those currently available in the ASCR ecosystem. These tools are rapidly gaining popularity across a broad swath of ASCR facilities' users.

Crosscutting Recommendations

The task force reviewed the recommended set of actions from each of the six key areas and made the following four crosscutting recommendations:

- **Collaboration:** Members of the ASCR facilities ecosystem should continue to engage in coordinating efforts and sharing information about benchmarks, technology specifications for upcoming systems, user facing capabilities, and workflows. The task force recognized past and ongoing collaborations within the ASCR ecosystem and encouraged continuing such efforts.
- **Participation:** The ASCR facilities should fund personnels' involvement in activities related to software standard bodies outside ASCR.
- **Shared infrastructure:** The ASCR facilities should consider investing in shared infrastructure—such as continuous integration and containers—for new system software, libraries, and AI tools.
- **Software stack:** ASCR should consider developing an ASCR facility software stack that can be tailored to the specific needs of each ASCR facility, while enabling IRI related workflows and data transfer.

Background

Over the last few years, the ASCR ecosystem has been involved in a number of significant activities including deployment of the exascale supercomputers Frontier and Aurora; start of three new supercomputer upgrade projects; the launching of the integrated research infrastructure (IRI) initiative [1] and ASCR's high performance data facility (HPDF); responding to an executive order on safe secure and trustworthy AI for research and science [2]; participating in the National AI Research Resource (NAIRR) [3]; and funding the Consortium for the Advancement of scientific software (CASS) following the end of the exascale computing project. Collectively, the confluence of these events presents the ASCR facilities ecosystem with a number of opportunities and challenges, especially in the domain of scientific software. These challenges and opportunities arise from the change in the nature of scientific workflows, the growth of AI-powered computational tools, increase in data intensive applications, the vendor landscape for AI hardware, and the need for a unified user experience spanning across the ASCR ecosystem. To maintain ASCR facilities' leadership role, it is important to strategically evaluate the software demands on the ASCR ecosystem and identify critical areas of near and long-term investment.

The traditional workflow at the DOE computing facilities is changing [4—6]. DOE's investments in the experimental facility upgrades led to a new generation of facilities with unprecedented levels of brightness coupled with orders of magnitude increase in data generation. ASCR's IRI initiative enables the users of these facilities to utilize the power of supercomputing and high performance networks to solve mission critical science problems. While these new initiatives will open ASCR facilities to a wider range of science users, they also require the ASCR ecosystem to reinvent the workflow models. A large subsection of the new users currently employ cloud based, on-demand resources. In the near future, ASCR facilities need to consider employing workflows, where deployment of containers and container images as well as orchestration of container resources become a significant part of the software landscape [7—10]. To enhance interoperability and user code portability among ASCR facilities and external cloud environment, it will be also beneficial to engage in activities implementing standardization efforts [11].

The rapid growth of AI and machine learning enabled applications impacts both the hardware and software of the ASCR ecosystem. There is a clear increase in demand for computing power for machine learning applications. Prior to the arrival of deep learning in 2010, this demand doubled every 20 months. It has since changed to a doubling rate of every 6 months [12]. In contrast, the market share for the hardware developers has shrunk [13], posing challenges for scientific users. These changes in computing demand and vendor ecosystem have significant impacts for the upcoming system requirements [14—16]. While a number of new AI chips promise performance increase in many AI-enabled applications, proprietary system software and programming environments in these systems pose challenges for user code portability. Additionally, this new era of science also promises new advances in libraries and algorithms that rely on mixed precision [17—18].

With these changes in the computational science landscape, the ASCR facilities ecosystem needs to reevaluate its strategic vision for scientific software stewardship in six key areas: system software; programming environments, tools, and libraries (PE); data; artificial intelligence software; integrated research infrastructure (IRI); and stakeholder engagement. The new strategic vision builds from an analysis of existing gaps, challenges and opportunities associated with bridging these gaps, and recommended actions in each of these key areas. The following sections of this report outline the task force's findings from each of these key areas.

System Software

Summary

System software such as operating system, work scheduling software, storage management, and data transfer software exposes the capabilities of a system's hardware to its users. As such, system software is a fundamental determinant of those users' perceptions about the system's capabilities and ease of use. Traditionally, a variety of factors determine which system software is used on a given facility system, including the preferences of the system's vendor(s), the experience and expertise of the facility's staff, and the other systems and services with which the facility system must integrate. Consistency across facilities has become an important factor for some types of system software (e.g., operating system kernel type) but not yet all. We recommend long-term investment in the development and support of a system software stack for ASCR facility systems, such that the stack has some flexibility to be tailored to each facility's own needs. At a minimum, we recommend investment to support activities (e.g., workshops) that define the shared foundational capabilities common across ASCR facilities and portable ways to use them. The shared software stack or at least shared interfaces will provide returns in better application portability, sharing of expertise and experience between facilities, and support for inter-facility activities like DOE's Integrated Research Infrastructure (IRI).

Background

ASCR facilities share many system software needs for capabilities such as scalable system boot, workload scheduling, scalable job launch, authentication (including inter-facility authentication), and high performance data transfer (intra-system, intra-facility, and inter-facility). There have been very few external mandates about the software that satisfies these facility needs, with the most notable exceptions regarding data security (e.g., for health data) or authentication. However, several de facto standards have emerged and have been used by ASCR facilities and the wider high performance computing (HPC) community. For instance, Linux has become the ubiquitous standard operating system for HPC systems, though the specific Linux distribution varies. Storage with a POSIX (or at least POSIX-like) programming interface is very common on HPC systems. Though not as ubiquitous, Globus is very common for data transfer between facility systems and external sites. The emergence of de facto standards goes hand-in-hand with the increasing maturity of open source software, and some collections of HPC centers and users have leveraged the open source model to produce de facto collections of software targeting HPC resources, such as the Tri-lab Operating System Stack (TOSS) and the Extreme Scale Scientific Software Stack (E4S). Because the individual ASCR facilities have been largely left to negotiate with the system vendors regarding the development and support for system software for upcoming systems, there is a strong incentive toward open source software, community-supported (and thus free) collections.

Gap Analysis

There are several gaps between the needs of ASCR facilities personnel and the software provided by ASCR-funded, vendor-provided, and open source software projects. There are also situations where software options exist to satisfy a need, but there is no standard (de facto or otherwise) in use across ASCR facilities or even computing centers in general. These gaps range from support for user-facing capabilities related to performance and energy efficiency monitoring and control, to portable container support, to capabilities supporting operations and deployment of systems.

User control: Users and user-level runtimes often do not have sufficient control over hardware to enable them to configure it for energy-efficient computation. There is a lack of consistency in the exposure of device-specific counters and controls across ASCR facilities systems.

Container images: Users have no way to construct a single container image that will run across ASCR facilities systems, a capability that could be important for some IRI use cases. Users lack support for constructing container images using the same development tools and runtime libraries available on ASCR facilities production systems.

Software and hardware support: There are very few systems available with hardware and software similar to that in ASCR facilities production systems, that also allow system software research. There is no clear and consistent choice for provisioning software across ASCR facilities systems, or the wider HPC community in general. There is also a lack of automated frameworks and tools supporting hardware (especially node-level) deployment and maintenance. There are many manual steps required when installing a new node for diagnostics and entering into the system's scheduler, which increases the load on facilities personnel and increases the training difficulty.

Security: There is a lack of support for multi-level security environments (e.g., open science and more restrictive on the same resource) and for a consistent posture toward trust-based authentication across ASCR facilities systems.

Challenges and Opportunities

There are significant challenges to filling these gaps, but the gaps also present some important opportunities to the ASCR community. Several of these challenges and opportunities arise due to the heavy reliance of ASCR facilities, and HPC computing centers in general, on open source software.

Culture: There are several strong disincentives for changing how ASCR facilities have operated, including the loss of control over decisions and configuration regarding system software, waste of experience and institutional knowledge built up over a long period of time, and lack of personnel and time to implement a new approach while still satisfying the production system commitments (e.g., to deliver enough cycles for INCITE and ALCC allocations). This issue is further complicated by the fact that the HPC and networking resources deployed by HPC facilities are complex systems, and the software for configuring and monitoring them is generally also complex. It is very challenging to find a good balance between the software's ability to manage, monitor, and scale that complex system with the costs in both time and effort required to operate the system and diagnose functionality, performance, and scalability problems.

Open source software: ASCR facilities often lack relevance to the target "market" for open source software. Some types of system software are developed mainly, or even only, for high performance computing, such as batch queue software and scalable parallel file systems. For other types of system software, the HPC market is a niche in a larger market and tailoring the "generic" software to satisfy ASCR facilities needs requires substantial effort by ASCR facilities personnel. Because of its lack of relevance, ASCR facilities personnel often experience a lack of timely support from open source software projects, including those that ASCR facilities vendors leverage for their products. This leaves facility personnel with the choice of waiting for the community to fix it or fixing it themselves with the hopes of contributing it back to the open source project. Depending on the project, such "upstreaming" of modifications can take substantial amounts of time, especially if the vendor then has to validate the modified software within their software stack. The ability of ASCR facilities personnel to contribute modifications to open source projects is sometimes hindered by the software's license. This is especially true when an upstream project's maintainers change licenses to protect their intellectual investment and market share, causing vendors to switch to alternative software to avoid the use of the new license.

Container images: It is challenging to strike the right balance between homogeneity that supports portability and consistency across ASCR facilities, and heterogeneity that supports fault tolerance. For example, ASCR facilities production systems currently support different container runtimes and the images they support are not necessarily portable, requiring multi-facility users to produce site-specific images. But standardizing on a single container runtime opens all facilities to risk if the runtime contains a defect that does not have a timely fix.

Engagement: Each ASCR facility is part of an institution that houses computer science researchers, and ASCR Research supports many projects that produce software that is of great interest to the users and operators of ASCR Facility systems. Some facilities are more willing than others to rely on research products in their system software strategy. The research products vary in terms of maturity and stability, and in the amount and quality of support that the research team producing the software can provide. In addition, ASCR facilities face the challenge of both rapid change among workloads and of long-lived workloads. The system software that best supports a facility's current workload may not be a good choice if the workload changes due to emergence and adoption of new software paradigms (e.g., the integration of AI/ML into traditional modeling and simulation workflows). But some science communities regularly plan campaigns that exceed the lifetime of any individual production computing system.

Enterprise technology: Despite some of the challenges in adopting open source software noted above, there is a great opportunity to leverage non-HPC technologies in ASCR facilities environments. System management technologies pioneered and hardened for non-HPC-focused data centers (e.g., Redfish) are often applicable to HPC facilities also. Adopting these enterprise technologies would provide several benefits including: lower net costs, controlled risk, broader hiring pools, reduced operational churn.

Recommended Actions

Based on our analysis of the gaps, challenges, and opportunities regarding system software for ASCR facilities, we recommend the following actions:

Software stack: We recommend long term investment in a flexible, supported software stack that can be tailored to the specific needs of each ASCR facility. We estimate that the development of such a software stack will require at least five (5) and probably closer to ten (10) years of investment, which would have to occur in parallel to normal ASCR facilities operations. Supporting such a stack would also require sustained investment for an indefinite duration. Such a software stack should:

- Feature enough flexibility to be tailored to the specific needs of each ASCR Facility, while retaining shared interfaces across facilities to support portability and implementation of support for projects like IRI.
- Feature zero trust capabilities, similar to those found in the cloud services industry, plus support for confidential computing on the same resources as open computing.
- Include scalable tools that provide a clear view of the state of the entire system, especially when running unknown workloads.
- Automation to reduce the staff effort required to deploy and operate ASCR Facility systems.

Standards: We recommend a near-term effort by facilities personnel to define the user-facing capabilities provided by all (or at least most) ASCR facilities and to agree on the interface/standard or implementation that all facilities will deploy to expose each capability. This activity may be supported by a workshop or series of workshops and should last no longer than a year. We recognize the need to balance having sufficient representation from each facility with a variety of expertise, with the experience that it is difficult to reach consensus with too many people involved. We also recognize the need for these meetings to include people with decision-making authority.

Open source software: We recommend the adoption of open standards and open source software whenever possible. Such adoption should occur in a timely fashion, with IPv6 adoption serving as an example cautionary tale. As a corollary, we recommend ASCR facilities and ASCR Research provide support for participating in standards development and standards-compliant implementations.

Software research: We recommend that ASCR facilities fund the deployment of test and development/proof of concept systems to host system software research and the evaluation of the products of that research. These systems should be similar to the hardware and software of ASCR facilities production systems and should be excluded from requirements regarding workload scale (e.g., requirements on number of available cycles consumed by leadership-class jobs). This support may involve procurement of additional resources when procuring new production systems (e.g., buying an extra cabinet), or extending the lifetime of an aged system (e.g., Summit).

Programming Environments, Tools, and Libraries

Summary

The landscape of HPC software used in HPC facilities is evolving, including a shift towards open source tools and protocols, new programming languages and models, and the increasing importance of emerging workloads like AI/ML and performance portable IRI workflows. In the context of this evolution the “traditional” HPC languages and tools (e.g. Fortran, MPI) remain important. There is also a shift from primarily system integrator software components to increasingly more software development from the HW/chip vendors. Sustained investment in programming models, software packaging, testing, distribution and deployment, is needed to meet the challenges created by this demanding and dynamic landscape and deliver high value to users of HPC facilities. We recommend cross facility collaboration, coordinated investment in key projects, engagement with open source and standardization efforts, requirements gathering and sharing, and continued support of traditional models while embracing new technologies.

Background

One change in the traditional computing ecosystem arises from the increased importance of containers. HPC facilities typically provide a core programming environment consisting of compilers; MPI libraries; and math libraries like BLAS and LAPACK, that are delivered via modules, RPMs, shared filesystems, or system images. Increasingly, containers are used to increase portability and give application developers more control than offered by the traditional HPC programming environments. As new users enter the ASCR HPC ecosystem through frameworks such as IRI, the contrast between these two platforms become even more important in determining the users’ experience in the ASCR ecosystem.

A separate, but related, change in software programming environments arises from the software providers. Major potential hardware vendors are now providing much of the software stack, often leveraging open source projects like GNU, LLVM, OpenMPI, and MPICH as upstream bases for their tools. At the same time in the changing vendor landscape, there are fewer traditional integrators and those that remain, are increasingly relying on the hardware vendor provided toolchains. One notable exception to this trend is the DOE funded Exascale Computing Project (ECP). ECP has made significant investments across a wide range of software relevant for programming environments. For example: the Spack package manager has emerged as a popular HPC tool; open source MPI libraries such as MPICH, OpenMPI were supported; compilers such as LLVM/ flang received contributions; and programming models such as OpenMP were targets of DOE investment.

Finally, the nature of scientific software stack is changing rapidly. While traditional languages and models like MPI, C/C++, Fortran, and OpenMP still remain crucial, new options like Python and Julia are appearing. Additionally, parallel features are becoming part of base language standards of C++

and Fortran. Portable workflows and AI/ML based software are also becoming increasingly important, especially for IRI.

These changes impact the way user-facing software is deployed, maintained, and supported at facilities in both the short and long terms. Support staff must understand the tools and methods users need for their science goals. The open source/standards shift signals opportunities for lasting impact from investments in those projects. Internal facility software like Linux OS features, systems software and schedulers also need updates to support new capabilities like containers. Gap analysis is required in areas like programming model implementations for advanced architectures, power management, workflows tools and more.

Gap Analysis

The ASCR ecosystem has supported a number of tools and libraries that are relevant to the scope of this report. For example, several compiler and programming efforts such as flang, OpenMP, and Kokkos, have been funded. ASCR also supported MPICH and OpenMPI and funded widely used "next generation" accelerator aware math libraries such as SLATE or PETSc. Despite these activities, there are a number of gaps in the current capability to meet the needs of the future demands/challenges.

System integration: Traditionally, system integrator vendors have largely performed the integration work. As a consequence, standards for power management and energy efficiency are nascent.

Continuous integration: New tools such as Spack still require significant customization work to meet facility needs. Sustained and continuous investment in programming models for advanced architectures for performance and portability is needed to ensure a competitive landscape. For example, investment in multiple OpenMP offload implementations that can target architectures in ASCR facilities can be beneficial. System software changes or customizations for unique considerations are needed to support these capabilities at scale.

New workflows: Tools that support large scale workflows—such as coupled AI and simulation at scale–and profiling and debugging at scale have varying levels of maturity depending on the architectures involved.

Coordination with other stakeholders: The draft technical requirements for the upcoming ASCR HPC systems [14—16] are in close alignment, but this could be further strengthened for future systems. Integration into standards and open source communities requires long term planning, time, and effort to be effective influential contributors. In addition, the facilities should consider gathering system requirements from users. While the facilities have input from the users through allocation managers, tickets, readiness programs, and upgrade projects, these efforts do not always include the full scope for long term planning.

Containers: Users lack tools for constructing container images with apps built with vendor-provided development tools. Also, currently the ASCR ecosystem lacks the capability for users to build a single container image that can span across the ASCR facilities.

Challenges and Opportunities

While the path to the future will ideally be achieved by bridging the gaps, a number of challenges still exist. These challenges arise from

- Trend toward single-point-of-failure situations, especially in C/C++ compiler space;
- Large and changing workload, as compared to National Nuclear Security Administration (NNSA) computing facilities;
- Localized preferences for specific debugging and performance tools;
- Research, time, and investment needed to design and stabilize hardware features codesigned with traditional language parallelism features.

The timeframe to address these challenges can be medium to long-term, especially when work involves standards and major open source projects. For example, adding a new feature to an ISO standard such as Fortran or C+ may take 3-7 years (or even more for major changes).

Recommended Actions

Coordination with other stakeholders: The ASCR ecosystem stands to benefit from cross facility collaboration between upgrade project teams on topics such as microbenchmarks, benchmarks, technical requirements, and non-recurring engineering (NRE). The ASCR facilities should consider coordinating investment and long term engagement with key standards and open source projects including MPI, containers, parallel runtimes, compilers, and data management. We also recommend considering strategic investment in programming models across the labs. An assessment and definition of common requirements, taking into account new and emerging workloads, can also be beneficial.

Support for PE and compilers: The ASCR facilities should consider increasing support for traditional HPC Programming Environments, libraries and tools by targeted support for parallelism in traditional HPC languages and in abstraction layer software; common foundations for scalable debugging and performance tools; linear algebra, particularly targeting accelerators; visualization standard, frameworks, and libraries (ANARI Khronos). Also consider continuing investment in several open compilers to have multiple options (e.g. LLVM, GNU, and OpenXLA). The ASCR facilities should consider mitigating the risk arising from a single functional toolchain with future investments.

Software stewardship: Software products need extended and sustained investment and support for “hardening” in order to transition from research ideas and proofs of concept to robust production quality tools. The ASCR ecosystem should consider creating and supporting platforms that are able to support pilots, new development efforts relating to PE or tools, with reporting requirements tailored to match these use cases.

New workloads: A rapidly evolving AI landscape introduces new tools and use cases. Many of these tools and use cases are new to the ASCR computing ecosystem. Conversely, the ASCR computing ecosystem is also new to many users of these AI workloads. To continue delivering cutting edge scientific results while providing a seamless introduction for the new workloads, these tools and use cases need to be scaled and hardened to run on the ASCR HPC systems.

Data

Summary

Whether derived by experiment or simulation, data is the one of the key assets of DOE science. Scientific user facilities are now producing data that is both complex in structure and high volume while the availability of exascale computing resources is enabling complex high data volume simulations. At the same time, data science—particularly AI/ML—is rapidly evolving into a powerful tool to extract science from data. The convergence of innovations in experiment, simulation, and data science presents both challenges and opportunities that will be discussed below.

Background

The DOE Office of Science (SC) user facilities enable research across a wide range of science programs that will generate increasing volumes of data over the coming years [4—6]. Meanwhile technologies, such as AI/ML, are increasing in capability and adoption [12]. Optimal use of research data, both for AI/ML and more traditional algorithmic data processing, demands that data be Findable, Accessible, Interoperable, and Reproducible (FAIR principles). The importance of FAIR principles was highlighted in a 2022 Office of Science and Technology Policy (OSTP) memo on ensuring free, immediate, and equitable access to federally funded research [19].

Surveys of science communities at user facilities have highlighted capability gaps relative to the needs of data driven science. These drivers have led to the instigation of an effort to better integrate data sources, such as SC user facilities, with data simulation, processing, and storage capabilities of the ASCR facilities as an integrated research infrastructure (IRI). ASCR has recently announced the project to construct a new ASCR facility, the High Performance Data Facility (HPDF) to provide data focused capabilities, in particular supporting real time science workflows.

Gap Analysis

The ASCR led IRI Architecture Blueprint Activity [1] identified three broad categories of science use cases that demand software infrastructure interoperability and optimization: time critical patterns where data must be processed, and results returned with well-defined and predictable timing; data integration patterns, where data from multiple sources are combined (for example simulation and experiment); and long term campaigns, where resources must be available for extended periods of time. Consideration of these patterns and comments from facility users highlight a number of gaps, discussed next, for data-driven processing.

Allocation models: A number of gaps exist for currently operative allocation models, which prioritize compute needs over data transfer rates. The quality of service for data processing allocations need to show some improvement. Currently, models for prioritizing time sensitive data allocation mechanisms are lacking. When a data source controls the time for data release, some flexibility is needed to extend or shift the allocation period.

Authentication, accounting, and data security: Current authentication and accounting procedures can restrict users from a large collaborative project and overburden a small fraction of researchers authorized to use and analyze the data. In addition, there is not a uniform standard across the ASCR ecosystem for defining confidentiality level and data security. Projects face data lifecycle challenges such as transparently accessing datasets across multiple facilities, encountering limitations in utilizing distributed data sharing mechanisms, and lacking tools that cater to the complete data lifecycle.

Data storage, transport, and management: Data storage at different ASCR facilities is often transient and the tiered data storage management software is not uniform across the ASCR ecosystem. Current data transfer approaches (e.g. with Globus) need adaptation and new tools for streaming data use cases. The quality of data archive across the ASCR ecosystem is nonuniform. There is an overall lack of standards for data format, metadata, and management.

Network issues: Historically there have been issues with high-speed network connections into equipment at many facilities. For example, network interface issues on the HPC systems, or campus network design inserting low speed devices such as firewalls in the data path. This can have a huge impact on the performance of data transfers, introducing a bottleneck to the overall IRI architecture.

Challenges and Opportunities

The IRI patterns, and the gaps that they highlight, point to broad requirements for the software ecosystem. The most significant challenge involves establishment of a dynamic and scalable data management infrastructure that is integrated with the DOE computing ecosystem ("networked data infrastructure at scale"). Another challenge arises from network, workflow, and application performance monitoring and visualization tools. The establishment of the IRI program and the HPDF facility provides an opportunity. In addition, commercial applications of AI driven data management (such as VAST) provide future collaborative opportunities.

Recommended Actions

To bridge the existing gaps by leveraging available opportunities and addressing existing challenges, ASCR and ASCR facilities should consider investing in three key areas.

R&D program: An R&D program complementary to IRI and HPDF that sources problems from these two facilities and contributes ideas and potential solutions to them.

Storage: Develop extreme scale distributed tiered data storage, archiving, and cataloging capability that supports data management and federated learning and facilitates FAIR data stewardship.

Transport: Extend data transport and workflow tools to support non-file-based data flows and "on demand computing".

Artificial Intelligence Software

Summary

The adoption of AI based applications is rapidly growing in DOE science. It is also having a considerable impact on the computing industry as a whole. The transformative potential, as well as the wider investment in AI-enabled software and hardware, offer considerable opportunity. ASCR-led coordination among facilities and investment, however, are needed to leverage these tools for science outcomes, and to tackle critical gaps in available AI software for science and HPC.

Background

As outlined in some recent DOE and NSF reports [3, 20—21], deep learning based workloads continue to expand while novel applications emerge. In parallel, a broader AI market is shaping the computing industry, leading to rapid development of software and tools. In a July 2024 memorandum, the Secretary of Energy Advisory Board suggests that DOE creates and makes available open and publicly accessible data and AI algorithms [22]. This AI software landscape will have to be well understood, tracked, and supported by the ASCR facility systems. AI software development activities within the ASCR ecosystem should not only leverage these industry efforts but also need to develop tools and applications focused on scientific mission needs for DOE and the broader scientific community.

Currently, the broad AI software ecosystem includes several active areas of development. The illustrative list below, while not comprehensive, outlines some of these key areas of development:

- **Framework:** Frameworks allow scientific applications to utilize the power of AI tools. These include AI and deep learning frameworks (e.g. currently Pytorch, Tensorflow, and Jax); traditional Machine Learning (e.g. sklearn, numpy, pandas)—including GPU/multi-GPU support; software for scaling to HPC resources (e.g. DeepSpeed, Horovod, Megatron, PyTorch DDP/ FSDP); orchestration, experiment tracking and model hosting (e.g. Ray, Weights&Biases, Hugging Face); and “interfaces” such as portals, inference servers, and JupyterHub.
- **Tools:** A number of AI software tools are also integral parts of the programming environment and system software. This array of AI tools includes software deployment tools (Conda, Containers, k8s); compilers (Numba, XLA, MLIR etc.); performance tools and debuggers for AI software; and tools, libraries and other software for interfacing AI with existing scientific software.
- **Management**: This group of AI software is useful for monitoring and managing complex AI-assisted workflows and associated data. These include workflows for fine tuning; data management (including e.g. vector databases); AI governance software; and consideration of traditional scientific HPC on AI accelerators.
- **Security:** A group of software is currently being developed to address AI security issues. This includes privacy-preserving, AI-safe model development; uncertainty quantification; and potentially AI-driven validation framework and tools.

This spectrum of AI software and tools enables an array of rapidly evolving applications. As these applications mature and evolve, some of the tools may also evolve with emphases different from traditional HPC software. Another area of rapid development involves inference and training workloads, and the interoperation of these workloads with other scientific software. For the future development of ASCR facilities' software ecosystem, the developing trends in this area should be closely monitored.

Gap Analysis

Much of the existing AI software capability is currently developed in the industry. Thus, the technical gaps considered below can be considered both from an industry and ASCR ecosystem perspective.

Disparity across vendors: Deep learning frameworks themselves are currently supported by DOE systems while the compatibility of these frameworks is supported by hardware developers or vendors. The level of functionality of the framework, however, varies considerably among vendors and can involve significant development and testing on new, or less well supported, hardware.

Hyperscalers vs DOE HPC systems: There is a current gap between the tools that are available in hyperscaler cloud environments and the DOE HPC systems. There are a range of reasons for this disparity, including differences in underlying technologies, such as k8s vs batch environment; policy and security stances; and use of proprietary tooling in specific cloud environments.

Tools: Performance tools and debuggers for AI on HPC need capabilities beyond those currently available in the ASCR community. The popularity of particular tools within this space has changed very rapidly in the last few years and still doesn't appear to have converged in the wider AI community or in science.

New workflow models: Workflows that combine scientific software (e.g. simulation, data pipelines) with AI will need development in DOE space. These are much less mature and with smaller community support.

Challenges and Opportunities

Challenges in bridging the technical gaps for AI software arise from the growth of commercial AI, which relies on tools somewhat distinct from AI for science. Other challenges arise from integrating AI into the traditional HPC software. Some of these challenges are outlined next.

Commercial AI vs AI for science: There is a considerable opportunity for the ASCR ecosystem to benefit from the investment by industry in AI software, especially engaging in community forums for tools such as PyTorch, which is a part of the open-source Linux foundation. There is also an opportunity arising from the significant industry investment in AI hardware and the training of large general purpose models which may in some cases be applicable (or adaptable) for science, and in some cases are open source.
Challenges arise, however, as commercial AI is not an exact match to AI for Science. Currently,

there is a heavy industrial focus on large-language models (LLM)s. Some of this focus is misaligned with scientific AI applications. There are risks arising from industry control in maintaining support for production science codes that rely on commercially developed tools. Difficulties in upstreaming can be encountered, both by vendors when contributing code to products controlled by others, as well as by users and facilities. Orchestration and workflow tools developed by the industry may not be compatible with current HPC orchestration or facility policies. Scaling and performance (particularly beyond "off-the-shelf" LLM examples) have gaps in methodology, expertise and tools. Further issues arise from the lack of convergence of AI tools. Fast moving software—as well as AI methods and approaches—lacks sufficient attention to issues such as backward compatibility required for production science. There are no standard frameworks or approaches for optimization, uncertainty quantification, and sharing.

Integration with existing scientific software: Integration of AI into existing scientific simulation and data pipelines is an open area with varied and changing approaches. Support for general purpose computing or programming for novel AI accelerators is limited and varies which also provides a challenge for integrating AI with existing scientific codes. Reduced, mixed-precision focus of AI offers both an opportunity and challenge for traditional scientific codes, e.g. solvers for domain sciences [17—18]. Currently, large-scale AI workloads at ASCR computing facilities are focused on training of models. As these workloads shift to inferencing, that shift will bring new challenges and introduce uncertainties arising from the way the inference is built into scientific pipelines.

Other issues: There are a range of potential issues and opportunities for energy-efficiency in AI that are relatively underexplored. Challenges also arise in recruiting and retaining expertise with valuable AI skills in DOE.

Recommended Actions

The working group recommends two broad high-priority action areas that will help bridge the existing technology gaps.

AI software stack: The ASCR facilities should consider investing in AI software stacks for HPC and Science, leveraging industry activity and bringing those tools to science communities, while addressing gaps. This coordinated, cross-facility initiative should involve: porting AI software efficiently to the DOE HPC software stacks; ensuring that the AI tools work in the DOE HPC system as well as hyperscaler cloud environments; developing portability across relevant accelerators; ensuring functionality in HPC/batch environment (including adaptation of that environment itself where appropriate); developing science use-cases and applications; developing or adapting tools for AI scaling and workflows; building test and deployment infrastructure for AI software; and hiring sufficient support staff for both tracking software evolution and changing approaches to scaling up AI workloads.

Integration of AI with existing scientific software: The ASCR facilities should also consider particular investment in software for the integration of AI with existing scientific software, such as simulation and data processing. This should involve efforts in service platforms for hosting models and allowing experimentation and integration with simulations and data; workflows that couple scientific software with AI; and development of evaluation metrics for science, working along with application teams. In addition, we recommend exploring the development of general-purpose computing or programming for novel AI accelerators as well as exploitation of reduced/mixed precision. We also recommend continuous education and staff training to keep up with AI state-of-the-art.

Integrated Research Infrastructure

Summary

The Integrated Research Infrastructure (IRI) vision to provide a collaborative automatable infrastructure for scientific workflows requires DOE labs and facilities to establish a set of standards and best practices for each key functional area defined within IRI. This document surveys the gaps, challenges, and recommended directions for the IRI as described in documents [1] and [23] through [28].

Background

Over the last decade, researchers within the Department of Energy's (DOE) Office of Science (SC) have been experimenting with ways to more efficiently perform their science through automated workflows. For these workflows to ultimately be successful the scientists must more effectively integrate data produced at their experimental facility with high-performance computing and high-performance data facilities located across the United States. As the next generation of exascale experiments come online the scientific community is demanding more functionality and seamless integrations into the high-performance computing facilities.

This rise of integrated-science approaches has led the DOE to develop the following vision for an Integrated Research Infrastructure (IRI):

> *To empower researchers to meld DOE's world-class research tools, infrastructure, and user facilities seamlessly and securely in novel ways to radically accelerate discovery and innovation. To respond to the evolving computational requirements of research and the competitive international innovation landscape, experimental facilities could be connected with high performance computing resources for near real-time analysis, and resources should be provided for merging enormous and diverse data for AI/ML techniques and analysis.*
>
> *This new integration paradigm will demand continuing evolution to ensure the U.S. remains a global leader in research and innovation.*

In 2022, ASCR carried out the Integrated Research Infrastructure Architecture Blueprint Activity. The goal was to generate a reference framework that would guide a unified, SC-wide approach to IRI. A series of documents were generated outlining technological, policy, and sociological challenges that would need to be considered to implement IRI. This document contains a summary of the software related gaps identified in those documents, challenges and opportunities relating to the gaps, and then recommended actions to ASCR that would help address the identified gaps.

Gap Analysis

The analysis presented here identifies gaps between the ideal state of the ASCR infrastructure as outlined in the IRI whitepaper [23] and the IRI blueprint document [1], with the current integration between experimental facilities and the current ASCR High-Performance Computing (HPC), High-Performance Data (HPD), and High-Performance Networking (HPN) facilities. At the time [23] and [24] were written, the High-Performance Data Facility (HPDF) had not yet been announced but is the first example of a new facility specifically designed to meet the data lifecycle requirements of IRI. References are provided to source material used for either gap, challenge, or recommendation items.

Containers and automation tools: There is currently a lack of portable code capabilities across HPC facilities, so a user needs to spend effort porting their application to each heterogeneous facility before it can be executed [1]. These HPC users desire a standardized software stack or a container environment on their HPC systems that can be utilized across facilities. A need for standard abstracted workflows and automation tools was identified, indicating a general desire for tools and libraries supporting workflow development [1]. Since different facilities have diverse capabilities, it's important to have tools that can identify and adjust for these differences, making the infrastructure more user-friendly.

Scheduling: Expanding the scope of the scheduling problem from solely HPC computing resources to (quasi) real-time coscheduling of multiple resource types across numerous facilities for a diverse array of workloads presents challenges that are not currently addressed by existing schedulers.

Security: The lack of federated authentication capabilities across ASCR facilities means that HPC facility users have to create separate user accounts and identities at each facility. Authorization models in use today are typically role-based solutions unique to each facility. The introduction of new IRI software across all ASCR facilities with externally accessible API and (possibly) shared security models will expose a large software-related attack surface for potential wrongdoers looking to expose inter-site vulnerabilities.

Data and software stewardship policy: Currently, projects encounter challenges in transparently accessing extensive datasets across multiple facilities, utilizing distributed data-sharing mechanisms, and lacking tools that cater to the complete data lifecycle [23]. In addition, the ASCR ecosystem lacks a common or well-understood set of data policies, FAIR data, and an ecosystem of-wide storage and searching capabilities [1]. While individual facilities are developing their own software systems (e.g. Intersect, Nexus, Superfacility, and SENSE), it is essential to consider the long-term stewardship—including upgrades, updates, and user support—for the future integrated IRI software ecosystem.

Challenges and Opportunities

The proposed IRI project presents a unique mix of challenges and opportunities. From a technical standpoint, it must integrate and manage vast and diverse data sets across multiple high-performance computing, data, and networking facilities. This involves ensuring seamless interoperability and robust performance. On an organizational level, it requires coordinated efforts among various stakeholders, including researchers, facility providers, and administrative bodies. This demands efficient project management and clear communication channels. Politically, it must navigate funding, policy-making, and intra-agency collaboration, aligning diverse interests and priorities. However, these challenges are balanced by significant opportunities: advancing scientific research through enhanced data sharing and computational capabilities, fostering innovation through interdisciplinary collaboration, and setting a model for future large-scale scientific infrastructure projects. The successful implementation of the ASCR IRI project promises to drive substantial progress in scientific discovery and technological advancement. This section summarizes a set of challenges and opportunities captured by the working group.

Codesign: We face a unique challenge with long-term campaign patterns [1], as they typically last 5 to 30 years and will undergo many changes to the IRI runtime as it evolves. In addition, IRI software will require some level of standardization across ASCR facilities, including a wider range of workloads [23]. An integrated API approach may be complicated or limited by differences in HPC system architectures, storage systems, services, resource allocation policies, and cybersecurity policies [23].

Secure versus usable: Balancing the need for good user experience, compliance to federal Zero Trust cybersecurity mandates, and individual facility security missions will introduce constraints on IRI that can lead to sources of impedance [1]. Many global authentication deployments, both in industry and academia, already utilize widely available federated identity technologies meant to address the IRI authentication problem. In addition, the ASCR facilities have expertise in threat assessment, established practices for detecting vulnerabilities, and existing infrastructure to handle security event responses. However, these security vulnerability assessments can sometimes hinder the introduction of new technology and infrastructure into production, so it's important to strike a balance.

Performance and usage metrics: Understanding what performance metrics need to be monitored/measured in this integrated environment may present a complex problem. These metrics relate to the performance of the control infrastructure (Facility API, control plane components, etc.) and the availability of the HPC, HPD, and HPN facilities (jobs continually executed for 48+ hours), including fine-grained metrics for individual resources and services with those facilities. Additionally, it will be important to specify a clear definition of "real-time" and what it means in the context of each facility type. Finally, an additional challenge to user satisfaction arises from troubleshooting across a multi domain system.

Recommended Actions

In these initial stages of the IRI project, a comprehensive review of the Integrated Research Infrastructure Architecture Blueprint [1] and this document has revealed numerous functional gaps that have yet to be thoroughly examined. The following recommended actions take into consideration the existence of the IRI program and aim to offer additional guidance on high-priority development items as determined by this working group.

Software stack: Develop a Facility software stack to enable IRI workflows by defining and providing common services (baseline API) for cross-facility workflows, authentication, authorization, accounting, and telemetry functions. Help develop new IRI specific scientific workflows, codesigned with users, that can frictionlessly leverage resources across DOE facilities. Continue to develop schedulers with appropriate capabilities and policies to enable said workflows. Develop tools and services required for outcome-driven resource planning across IRI to allow maximizing system efficiency. Refine availability, efficiency, and resiliency expectations of computing, network, and data resources. Deploy cross-facility measurement, analytics, and telemetry collection to facilitate more effective scheduling decisions, identify performance issues (API and resources), and facilitate troubleshooting.

Software stewardship: Activities should be coordinated with the established IRI program to perform ongoing maintenance on the developed Facility software stack. Continually assess the vulnerability, resiliency, functionality, and performance of the infrastructure and perform required improvements. Actively maintain the software stack while keeping backward compatibility. Optimize IRI performance and resiliency for workflows. Provide support, troubleshooting, and issue resolution for users of the IRI ecosystem and associated testbeds. The IRI program should establish an IRI testbed to develop and validate the software stack and user workflows. The program should also develop a data management infrastructure supporting the scientific data lifecycle within the IRI ecosystem (meta-data, FAIR, high volume, high performance, long term, etc.) and enable high-performance data transfer and real-time communication capabilities (data streaming).

Stakeholder engagement

Summary

ASCR facilities personnel are part of a complex software ecosystem. Many interactions occur within this ecosystem, such as collaboration on software products and standards, procurement of software from industry, and use of software provided on facility resources. Although these mission-critical interactions are often successful, improvement is possible. There are numerous gaps with respect to collaboration and engagement across facilities and with the wider open-source community. To address the gaps, we recommend support for better collaboration between facilities and the open source and open standards communities, long-term support for software developed with ASCR funding that is of importance to ASCR facilities users and operations, better recognition for non-research activities important to ASCR facilities personnel, and better consistency across facilities regarding what is allowed regarding open source contributions and support for common software development activities like continuous integration on ASCR facilities resources.

Background

There are three types of relationships between ASCR facilities and other entities with respect to software: client, provider, and synergistic.

ASCR facilities act as clients when they use software produced by other entities to satisfy facility software needs. There are many types of entities that might provide such software: other DOE labs or divisions within the facility's own lab (e.g., a research-focused computing division); academic partners which provide most of ASCR facilities workforce of experts; industrial partners that provide ASCR facilities with system hardware and accompanying software; standard bodies, foundations, and international collaboration efforts that define standards and direction future machines and software will take, as well as allow engaging industrial partners in a different setting; and finally the broader open-source community. Ideally, the facility provides input to the providing entity about the facility's needs to help ensure the software is more likely to satisfy those needs, but in some cases, it is necessary for facility personnel to take the software "as is" and adapt it to facility needs.

ASCR facilities act as providers when they produce software and/or provide services that other entities use to advance their scientific and/or engineering goals. Examples of such users include users of facility systems and services at DOE laboratories, academic institutions, and industry entities. It may also include members of the broader open-source community if facility staff produce software of general interest beyond high performance computing.

ASCR facilities act as synergistic partners when they collaborate closely with other entities that share their goals and needs. An example of synergistic partnership exists when a facility works with other computing facilities within the DOE and elsewhere that seek to provide familiar and/or interoperable computing experiences to users of multiple facilities. Professional societies that recognize and promote ASCR Facility members' expertise and accomplishments also act in a synergistic relationship.

Gap Analysis

With respect to ASCR facilities software and stakeholder engagement, there are several gaps where existing relationships are insufficient.

Research software engineers: There is a lack of recognition for Research Software Engineer (RSE) personnel, both inside and outside of DOE, as highlighted by the RFI response from US-RSE [29] to the "Stewardship of Software for Scientific and High-Performance Computing" notice from DOE. This lack of recognition reduces career attractiveness, hinders employee retention, and limits future opportunities for DOE personnel in an RSE role.

Future system software stack: There is a lack of regular coordination between ASCR facilities on software requirements for future system acquisitions. This leads to duplication of effort during software Non-Recurring Engineering (NRE), discrepancies in tool requirements, and inconsistent adoption of open standards. The result is increased cost for DOE, either directly when buying machines or indirectly when funding the efforts needed to port HPC applications and software between DOE machines. The lack of coordination also threatens the ability to develop and deploy an effective IRI infrastructure.

ASCR representation: Currently, there is insufficient representation of ASCR facilities' needs among standard bodies, foundations, and international collaborations. In the modern open source-driven environment, these entities drive the standards and software frameworks that facilities rely on for both user-facing and non-user facing software (e.g., operations support, systems software). Similar to how open access publication tends to increase citations and thus usefulness to the scientific community, the use of open-source software that was designed with facility needs in mind will tend to increase its impact within the DOE facilities that use it. The lack of facility interaction with these entities results in standards and software that are often less responsive to facility needs, which in turn results in poor adoption of open standards and software frameworks by facilities and their users. Instead, facilities rely on vendor-provided solutions, possibly funded through NRE or codeveloped with the help of DOE. Although this software may be more responsive to a facility's current needs, there is a danger that it is less likely to be supported long term due to its semi-custom nature.

AI standard body: A related gap is the lack of standard and standard bodies around AI. This lack has been highlighted in the National Artificial Intelligence Research Resource (NAIRR) final report [3]. With the extreme computational potential of the Leadership Computing Facility (LCF) systems, DOE is expected to have a huge impact in the AI and machine learning space, providing the only publicly available systems capable of training potentially malicious models as described in the "Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence" from the White House [2].

Engagement: Related to lack of participation in (de facto) community standards, there is a failure to engage the broader open-source community around facilities open-source projects, in spite of the extensive expertise facilities possess in scientific software development and dissemination. Community contributions and engagement in open-source project is critical to ensure the longevity of projects and their relevance to a broad audience.

Software support: Software of strategic importance to ASCR is left unsupported or insufficiently supported. These risks are known and have been studied in the "Enterprise Risks for Scientific Software in the Post Exascale Era" technical report [28]. Most of these software products are system software, part of the programming environment, or tools. Examples of these strategic components are alternative compilers, at scale debuggers, benchmark suites, or AI frameworks.

Cloud computing vs HPC: Finally, there is a widening divergence between cloud computing and HPC. Because the cloud computing market is much larger and more profitable for cloud providers than HPC, software development activity and thus funding that might go toward HPC is being steered toward cloud computing. Although some of this cloud-based software may be used effectively in HPC environments, for the most part HPC facilities are not benefiting from investment in cloud software.

Challenges and Opportunities

While there are a number of challenges in bridging the above-mentioned gaps, there are also a number opportunities.

Research Software Engineers: Unlike the situation with researchers and publications, there are no widely accepted criteria for success for people in RSE roles. Although organizations like US-RSE are working toward better recognition for RSEs, opportunities for recognition (e.g., professional society awards, or even DOE facility institutional awards) are few. Because of the importance of RSEs to ASCR facilities and ASCR in general, there is a great opportunity for ASCR to work with organizations like US-RSE to develop better recognition of RSEs within our organizations and thus increase the attractiveness of facility RSE positions for better workforce development and retention.

Future system software stack: Coordination among facilities requires compromises, usually a challenging cultural change to an organization. Also, the ASCR facilities' system must retain some diversity to avoid developing a single point of failure. They must maintain a certain autonomy to be able to be responsive to their own users' needs, and strict software requirements chosen solely for consistency across ASCR facilities could drive users and potential system vendors away. Nonetheless, continually defining and publicizing a common set of software requirements allows potential system offerors to better anticipate facilities requests while also defining a common foundation for application software development, which would control the cost of software porting efforts.

ASCR representation: ASCR facilities personnel do not currently receive support from ASCR facilities to represent their facilities in software standard bodies. Participation requires a nonnegligible amount of time and effort, and it is difficult to quantify and to evaluate a staff member's participation and recognize the individual's contributions. Also challenging is the lack of consistency across ASCR facilities for participation in such organizations. Because meaningful participation would help the standard bodies define and adopt standards that take ASCR facilities needs into account, there is substantial opportunity to have positive impact by providing support at a level needed for meaningful participation.

AI standard bodies: The growth of AI capability in recent years has made it extremely attractive to users of ASCR facilities, but the rapidity of that growth has left all stakeholders struggling with questions of ethical and responsible use. Because ASCR facilities provide unique resources that are very attractive as AI platforms, and because of the diversity of the workloads that target ASCR Facility systems, there is a great opportunity and strong need for ASCR facilities to help define the guard rails of AI use on ASCR facility systems and elsewhere.

Engagement: ASCR facilities vary in their posture toward open-source licensing and contribution rules. This variability presents a challenge for potential collaborations between DOE personnel on open-source projects. Support for continuous integration for software targeting ASCR facility systems is highly desirable but challenging to implement within the ASCR facility context due to security and attribution challenges with the validation of software contributions by external collaborators.

Software support: Rather than adopting software developed at ASCR facilities or even outside a facility but with ASCR funding, industry tends to redevelop software internally. Although such redevelopment provides greater control over intellectual property and potentially an ability to gain performance over competitors (e.g., the ability to add the company's "secret sauce"), any duplication of effort should be avoided. An alternative is for ASCR to fund maintenance and hardening activities, which is challenging because it competes with the need for funding research activities.

Cloud vs HPC: Cloud-based software does not require the level of robustness and performance that HPC—especially leadership computing—requires, and its general response to failures is very different. Cloud-based software is often focused on supporting high throughput, whereas HPC more often rewards low latency (e.g., running a program as quickly as possible). Nevertheless, there is substantial overlap in the software stacks being developed for the use and operation of cloud platforms and ASCR facilities, especially as cloud providers seek to attract HPC workloads.

Recommended Actions

To bridge the gaps while addressing the challenges and leveraging on the opportunities, we recommend the following actions:

Research software engineers: Better recognition for RSE personnel by defining evaluation guidelines for RSE personnel, as well as fostering awards within ASCR facilities, the institutions that host them, and the relevant professional societies that better highlight RSE contribution to science and society.

ASCR software stack: We recommend increased levels of collaboration among ASCR facilities on software requirements for future system acquisitions by defining a set of common requirements while still addressing needs unique to each facility.

Participation: ASCR should consider increasing support for participation of facilities in standard bodies, foundations, and international collaborations at a level sufficient for development time, regular and exceptional meetings, and membership fees. Personnel contributions to these structures should be recognized. To increase the impact of these contributions, the collaboration on software requirements for system acquisitions described above should include open standards in RFPs where appropriate. We also recommend further study within ASCR facilities to define a shared legal and support framework around licensing, intellectual property, etc. of open-source development by facilities personnel.

AI standard bodies: We also recommend increasing ASCR facilities' participation in standard and standard bodies around AI through active engagement in efforts, such as NAIRR, to help define and adopt standards for AI. Resources shared through common platforms such as the NAIRR pilot need to be distinct from leadership class jobs, with relaxed reporting requirements and flexible policies compared to the LCF production computing resources.

Software stewardship: ASCR should continue to provide support for maintenance and future development of software of strategic importance to ASCR via an extended program for software maintenance. Specifically, such support should consider increasing the scope of current ASCR funded initiatives such as the Consortium for the Advancement of Scientific Software (CASS) , including credible alternatives for critical software (compilers, at scale debugging) and benchmark suites.

Engagement: ASCR facilities should consider increased collaboration with the open-source community by developing and deploying comprehensive services in support of scientific software development, stewardship, and sustainability. These collaborations can include providing, on open and collaborative platforms, continuous integration (CI) capabilities closely resembling production or experimental resources, as well as training in the usage of these resources, plus development of sufficient security and attribution guarantees to allow use by external users. We also recommend the development and deployment of a match-making service for users to identify ASCR-funded open-source software that serves their needs and is known to run well on ASCR facilities systems.

Cloud and HPC programming environment: ASCR facilities should consider exploring the common elements between the HPC programming environment and clouds. Identification of these elements and possible future codesign efforts can be mutually beneficial. On one hand, new ASCR users, who might be familiar with cloud-based services, will be able to integrate their scientific applications more readily into the ASCR systems. On the other hand, such codesign efforts can also benefit cloud providers to offer cloud-based HPC infrastructure to more seasoned users.

References

1. Miller, William, Deborah Bard, Amber Boehnlein, Kjiersten Fagnan, Chin Guok, Eric Lançon, Sreeranjani (Jini) Ramprakash, Mallikarjun Shankar, Nicholas Schwarz, and Benjamin L Brown. “Integrated Research Infrastructure Architecture Blueprint Activity (Final Report 2023).” July 3, 2023. https://doi.org/10.2172/1984466.
2. The White House. “Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence.” October 30, 2023. https://www.federalregister.gov/documents/2023/11/01/2023-24283/safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence
3. “Strengthening and Democratizing the U.S. Artificial Intelligence Innovation Ecosystem: An Implementation Plan for a National Artificial Intelligence Research Resource.” NAIRR Task Force Final Report, January 2023. https://nsf-gov-resources.nsf.gov/2023-10/NAIRR-TF-Final-Report-2023.pdf?VersionId=2RqgASgtGLzEI6QKsMIL.MWITnjgrmh_
4. Yager, Kevin G, Pawel W Majewski, Marcus M Noack, and Masafumi Fukuto. “Autonomous X-Ray Scattering.” *Nanotechnology* 34, no. 32 (May 22, 2023): 322001. https://doi.org/10.1088/1361-6528/acd25a
5. Proudhon, Henry, Maxime Pelerin, Andrew King, and Wolfgang Ludwig. “In Situ 4D Mechanical Testing of Structural Materials: The Data Challenge.” *Current Opinion in Solid State and Materials Science* 24, no. 4 (August 2020): 100834. https://doi.org/10.1016/j.cossms.2020.100834.
6. Babu, Anakha V., Tao Zhou, Saugat Kandel, Tekin Bicer, Zhengchun Liu, William Judge, Daniel J. Ching, et al. “Deep Learning at the Edge Enables Real-Time Streaming Ptychographic Imaging.” *Nature Communications* 14, no. 1 (November 3, 2023). https://doi.org/10.1038/s41467-023-41496-z.
7. Nguyen, Thanh-Tung, Yu-Jin Yeom, Taehong Kim, Dae-Heon Park, and Sehan Kim. 2020. "Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration." *Sensors* 20, no. 16: 4621. https://doi.org/10.3390/s20164621
8. Zhou, Naweiluo, Yiannis Georgiou, Marcin Pospieszny, Li Zhong, Huan Zhou, Christoph Niethammer, Branislav Pejak, Oskar Marko, and Dennis Hoppe. “Container Orchestration on HPC Systems through Kubernetes.” *Journal of Cloud Computing* 10, no. 1 (February 22, 2021). https://doi.org/10.1186/s13677-021-00231-z.
9. Zhong, Zhiheng, Minxian Xu, Maria Alejandra Rodriguez, Chengzhong Xu, and Rajkumar Buyya. “Machine Learning-Based Orchestration of Containers: A Taxonomy and Future Directions.” *ACM Computing Surveys* 54, no. 10s (January 31, 2022): 1–35. https://doi.org/10.1145/3510415.
10. Trott, Christian R., Damien Lebrun-Grandie, Daniel Arndt, Jan Ciesko, Vinh Dang, Nathan Ellingwood, Rahulkumar Gayatri, et al. “Kokkos 3: Programming Model Extensions for the Exascale Era.” *IEEE Transactions on Parallel and Distributed Systems* 33, no. 4 (April 1, 2022): 805–17. https://doi.org/10.1109/tpds.2021.3097283

11. Ramalingam, Chithambaramani, and Prakash Mohan. 2021. "Addressing Semantics Standards for Cloud Portability and Interoperability in Multi Cloud Environment." *Symmetry* 13, no. 2: 317. https://doi.org/10.3390/sym13020317
12. Sevilla, Jaime, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. "Compute Trends Across Three Eras of Machine Learning." *2022 International Joint Conference on Neural Networks (IJCNN)*, July 18, 2022, 1–8. https://doi.org/10.1109/ijcnn55064.2022.9891914.
13. Jon Peddie Research. Dow, Robert. "Shipments of Graphics Add-in Boards Decline in Q1 of 24 as the Market Experiences a Return to Seasonality." June 6, 2024. https://www.jonpeddie.com/news/shipments-of-graphics-add-in-boards-decline-in-q1-of-24-as-the-market-experiences-a-return-to-seasonality/
14. Argonne Leadership Computing Facility. "Draft Technical Requirements for the ALCF-4 System." Accessed December 18, 2024. https://www.alcf.anl.gov/draft-technical-requirements-alcf-4-system
15. Oak Ridge Leadership Computing Facility. "Technical Requirement Documents for OLCF-6." Accessed December 18, 2024. https://www.olcf.ornl.gov/wp-content/uploads/OLCF-6-Draft-Technical-Requirements-v4.0.pdf
16. National Energy Research Scientific Computing Center. "Draft NERSC-10 Technical Requirements." Last modified March 13, 2024. https://www.nersc.gov/systems/nersc-10/draft-tech-req/.
17. Abdelfattah, Ahmad, Hartwig Anzt, Erik G Boman, Erin Carson, Terry Cojean, Jack Dongarra, Alyson Fox, et al. "A Survey of Numerical Linear Algebra Methods Utilizing Mixed-Precision Arithmetic." *The International Journal of High Performance Computing Applications* 35, no. 4 (March 19, 2021): 344–69. https://doi.org/10.1177/10943420211003313.
18. Dongarra, J., Gunnels et al. "Hardware Trends Impacting Floating-Point Computations in Scientific Applications." https://doi.org/10.48550/arXiv.2411.12090.
19. Nelson, A. "Ensuring Free, Immediate, and Equitable Access to Federally Funded Research". August 25, 2022. Office of Science and Technology Policy, Executive Office of the President of the United States. https://bidenwhitehouse.archives.gov/wp-content/uploads/2022/08/08-2022-OSTP-Public-Access-Memo.pdf.
20. Carter, Jonathan, John Feddema, Doug Kothe, Rob Neely, Jason Pruet, Rick Stevens, Prasanna Balaprakash, et al. "Advanced Research Directions on AI for Science, Energy, and Security: Report on Summer 2022 Workshops." May 1, 2023. https://doi.org/10.2172/1986455.
21. Stevens, Rick, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, and David Brown. "AI for Science: Report on the Department of Energy (DOE) Town Halls on Artificial Intelligence (AI) for Science." February 1, 2020. https://doi.org/10.2172/1604756
22. "Recommendations on Artificial Intelligence for the U.S. Department of Energy." Secretary of Energy Advisory Board, July 2024. https://www.energy.gov/sites/default/files/2024-08/Artificial%20Intelligence%20at%20the%20U.S.%20Department%20of%20Energy%20Recommendations%20July%202024.pdf
23. Adams, Corey, Katie Antypas, Debbie Bard, Shane Canon, Eli Dart, Chin Guok, Ezra Kissel, et al. "Toward a Seamless Integration of Computing, Experimental, and Observational

Science Facilities: A Blueprint to Accelerate Discovery." March 8, 2021. https://doi.org/10.2172/1863562.

24. Brown, Benjamin. "A Vision for the ASCR Facilities Enterprise." Advanced Scientific Computing Advisory Committee Meeting. Address presented at the Advanced Scientific Computing Advisory Committee Meeting, September 29, 2021. https://science.osti.gov/-/media/ascr/ascac/pdf/meetings/202109/Brown_ASCR_Facilities_Vision_202109.pdf
25. Brown, Benjamin. "DOE's Integrated Research Infrastructure (IRI): Vision, Strategy, and Implementation." Advanced Scientific Computing Advisory Committee Meeting. Address presented at the Advanced Scientific Computing Advisory Committee Meeting, June 12, 2023.
26. Dart, Eli, Jason Zurawski, Carol Hawk, Benjamin Brown, and Inder Monga. "ESnet Requirements Review Program Through the IRI Lens: A Meta-Analysis of Workflow Patterns Across DOE Office of Science Programs (Final Report)." November 9, 2023. https://doi.org/10.2172/2008205.
27. Monga, Inder, Chin Guok, and Mallikarjun Shankar. "Federated IRI Science Testbed (FIRST): A Concept Note." December 1, 2023. https://doi.org/10.2172/2205149.
28. Bernholdt, David, Wahid Bhimji, Benjamin Brown, Saswata Hier-Majumder, John MacAuley, and Thomas Uram. "Enterprise Risks for Scientific Software in the Post Exascale Era." June 30, 2023. https://doi.org/10.2172/2222458.
29. Mundt, Miranda, Vanessa Sochat , Daniel S Katz, Sandra Gesing , and Verónica G Melesse Vergara. "DOE Software Stewardship Challenges in Diversity, Professional Development, and Retention of Research Software Engineers." *Department of Energy*, January 7, 2022. https://www.regulations.gov/comment/DOE-HQ-2021-0027-0033

Glossary of Acronyms

AI: Artificial Intelligence

ALCC: ASCR Leadership Computing Challenge

ALCF: Argonne Leadership Computing Facility

API: Application Programming Interface

ASCR: Advanced Scientific Computing Research

BLAS: Basic Linear Algebra Subprograms

CASS: Consortium for the Advancement of Scientific Software

CI: Continuous Integration

DOE: Department of Energy

E4S: Extreme Scale Scientific Software Stack

ECP: Exascale Computing Project

ESnet: Energy Sciences Network

FAIR: Findable, Accessible, Interoperable, and Reusable (*refers to the FAIR data principle*)

FSTF: Facilities Software Task Force

HPC: High Performance Computing

HPD: High Performance Data

HPDF: High Performance Data Facility

HPN: High Performance Network

INCITE: Innovative and Novel Computational Impact on Theory and Experiment

IPv6: Internet Protocol Version 6

IRI: Integrated Research Infrastructure

LAPACK: Linear Algebra PACKage

LCF: Leadership Computing Facility

LLM: Large Language Model

ML: Machine Learning

MPI: Message Passing Interface

NAIRR: National Artificial Intelligence Research Resource

NERSC: National Energy Research Scientific Computing Center

NNSA: National Nuclear Security Administration

NRE: Nonrecurring Engineering

OLCF: Oak Ridge Leadership Computing Facility

OS: Operating System

OSTP: Office of Science and Technology Policy

PE: Programming Environment (*also used in this report as a shorthand to the programming environments, tools, and libraries focus area*)

POSIX: Portable Operating System Interface

R&D: Research and Development

RFI: Request For Information

RFP: Request For Proposal

RPM: RedHat Package Manager

RSE: Research Software Engineer

SC: Office of Science