Project Summary

S4PST: Sustainability for Node Level Programming Systems and Tools

Keita Teranishi (Principal Investigator); Oak Ridge National Laboratory
William Fredie Godoy (Co-Investigator); Oak Ridge National Laboratory
Pedro Valero-Lara (Co-Investigator); Oak Ridge National Laboratory

Objectives Our goal is to enable a more predictive ecosystem for the sustainability of high-performance computing (HPC) software for node level programming systems and tools (PST). The scope includes exploratory work to identify technical, economic and social requirements important to the HPC community and to plan for a sustainable strategy to deliver robust, trustworthy software in a thriving community invested in addressing the challenges of the current and future heterogeneous computing landscape.

Description We propose a planning and path-finding effort to identify the important sustainability thrust areas from HPC software projects, in particular those funded by the US Department of Energy (DOE) Exascale Computing Project (ECP), while defining a vision for emerging technologies. We called it the ecosystem the Sustainability for node level Programming Systems and Tools (S4PST). Our main deliverables would be: i) the formulation, analysis and publication of a comprehensive survey study, ii) a workshop organized among project collaborators, and iii) a comprehensive report outlining the research and direction to understand and reduce technical, economical, and social barriers to support a sustainable ecosystem.

Methodology Our methodology to achieve our deliverables includes the following main activities: (a) Identify technical, economical, social requirements and barriers for software sustainability. (b) Define direction and a hierarchy of priorities and tasks to craft a sustainability proposal that will create a larger impact in the DOE HPC software community. (c) Explore reward mechanisms to promote new ideas and people in the community executing the tasks required for software sustainability. (d) Balance existing and explore future technologies in programming models and systems for extremely heterogeneous systems.

We will form activity teams focusing on the following thrust areas: inclusivity and training, vendor independent software quality assurance, methodologies for feature tracking, test-case synthesis, and packaging, and DOE-aligned efforts on PST for emerging technologies. We will disseminate our findings through a paper and a workshop report for crafting an ecosystem for the future of DOE's portfolio for node-level programming systems and tools and the community driving these efforts.

Impact S4PST expects that a more predictive and prescriptive ecosystem can reduce the current technical, social and economic debt of using the current scattered corrective approach in delivering sustainable HPC software. We will continue the ECP long-term vision for a coordinated and synergistic approach to prepare for the post-Moore, extremely heterogeneous world and deliver a sustainable software ecosystem of node-level programming systems and tools to future generations.

Team Our team is comprised of experts from academia, government and industry with a proven track record of delivering sustainable HPC software, diversity initiatives, and research on programming systems and tools. Many of the teams members are currently funded by ECP and work closely with DOE HPC systems. An inclusive and diverse composition of the team ensures a wider view of the requirements and the vision for software sustainability in the DOE.

S4PST: Sustainability for Node Level Programming Systems and Tools

Funding Opportunity Announcement Number: DE-FOA-0002844
DOE/SC Program Office: Advanced Scientific Computing Research (ASCR)
DOE/SC Program Office Technical Contact: Dr. Hal Finkel and Dr. William Spotz

Applicant/Institution:
Oak Ridge National Laboratory
1 Bethel Valley Road
Post Office Box 2008
Oak Ridge, Tennessee 37831

Lead Principal Investigator:
Name: Keita Teranishi
Telephone: +1 865.341.2324
Email: teranishik@ornl.gov

Administrative Point of Contact:
Name: Bailie Roy
Telephone: 865-574-5538
Email: roybc@ornl.gov

Proposed Collaborative Budget

Institution Point of Contact	Institution	Year 1 ($K)	Total Budget ($K)
Keita Teranishi	Oak Ridge National Laboratory	44	44
Sameer Shende	University of Oregon	15	15
Sunita Chandrasekaran	University of Delaware	10	10
Alan Edelman	Massachusetts Institute of Technology	10	10
Hartmut Kaiser	Louisiana State University	10	10
Damian Rouson	Lawrence Berkeley National Laboratory	12	12
Michel Schanen	Argonne National Laboratory	12	12
Ignacio Laguna	Lawrence Livermore National Laboratory	12	12
TOTAL		**125**	**125**

Contents

1 Introduction

1.1 Motivation

The US Department of Energy (DOE) Exascale Computing Project (ECP) has fostered and strengthened the use of modern software engineering practices for the development of applications and libraries [7,11], resulting in the coordinated and interoperable E4S[1] and xSDK[2] ecosystem stacks. While this is a cost-effective strategy, the technologies underpinning our HPC software are developed using programming systems and tools (PST). Today, our major PST stack consists of traditional high-performance computing (HPC) programming languages (Fortran, C, C++, Python) that offload ecosystem aspects to third-party implementations motivated by use-cases outside science due to the broader nature of these communities. The result is a massive number of specifications and variants creating an expensive "many-ecosystems" orchestration adding overhead costs to the consumers near end of the pipeline in our development cycles [22].

Despite efforts such as ECP, this dependence on legacy design decisions continues to lead the broader HPC community as a whole to a scattered, and "corrective" maintenance model of the ecosystem aspects to enable performance portability, productivity, correctness, and reproducibility when building our software. These aspects combined with the niche "one-off" nature of scientific software resulted in a separate and uncoordinated development model that was identified early on as unsustainable when crafting the vision for exascale computing software [10]. This technical debt is expected only to grow, and be paid by the future workforce in the pipeline, with the increasingly heterogeneous computing landscape in the post-Moore era [23].

1.2 Requirements for Sustainability

The latest generation of programming languages, e.g. Julia [8] and Rust [18], embrace critical ecosystem aspects (e.g. packaging, tooling, instrumentation) as part of the overall development of these modern languages, leading to a more productive experience addressing today's needs. A noteworthy effort coming out of ECP in this direction is Spack [12], as a unifying package manager targeting HPC facilities. Productivity gains are obtained through a rich, common layer structure that enhances the orchestration of the packaging and deployment of our HPC-targeted software replacing the previously uncoordinated facility –and system– specific efforts. Another success example is the adoption of LLVM [17] by major vendors and highly productive languages (e.g. Julia, Rust and Python/Numba) as the common compiler backend. LLVM empowers different communities by providing a common and coordinated development effort which leads to more productive experience as novel hardware architectures emerge and programming models and languages evolve.

As a result, there are ecosystem aspects that are worth providing a common structure that fosters a culture of collaboration among the broader and diverse HPC community. Nevertheless, for a successful "buy-in" and adoption of sustainable practices and ecosystem from different communities, it is required to understand the uniqueness of HPC software and its nontrivial and highly specialized tasks [20]. In this environment, it is key to understand societal requirements to expand the current pool of talent as the people behind HPC efforts are highly skilled individuals, far from any type of commodity resource. Hence, sustainability requirements for this context can be classified at three different levels: technical, economical and social.

Technical requirements imply that modern software must be well tested, validated, deployed and that results can achieve a level of reproducibility to trust the scientific end products at a wide range of scales and targeted heterogeneous platforms in the DOE HPC landscape. The PST for the DOE users are unique in a variety of open-source and vendor specific APIs for parallelism and concurrency, requirements for correctness, validation and verification (V&V) process for application output, and support for rich stack of legacy software and applications. Today, each project must address many of these aspects individually

[1] https://e4s-project.github.io
[2] https://xsdk.info

generating siloed processes and interactions, thus bringing an excellent opportunity to rethink codesign for the future [7]. Additionally, the current landscape of extreme heterogeneity in the post-Moore's era and the data avalanche of AI data-driven workflows has led to an inflection point in the future of HPC disrupting the successful monolithic model [21].

Economical requirements imply that the process behind the software lifecycle must have reasonable costs as more heterogeneous hardware and programming models become available in HPC. This process includes source code refactoring and software testing. More importantly, a larger cost is incurred with documentation and design decision or standardization process such as feature addition and deprecation. The cost for outreach activities such as training and consultation can be expensive without enough participants. Current costs in a corrective maintenance model are only expected to increase — this includes the costs of interactions between software maintainers, the community and the accumulation of technical and social debt. The community should consider amortization of these debts by proactively introducing and rewarding new practices and PST designs that embrace ecosystem aspects as their goals for sustainable software to be handed to the next generation of HPC experts.

Social requirements imply that the next generation of HPC practitioners need to continue fostering a diverse and inclusive culture of predictive software quality, productivity, reproducibility, and maintenance to solve the important scientific challenges of tomorrow. DOE is in a unique position to contribute and reduce social barriers towards a sustainable reward, retention, and community model impacting the pipeline of people behind these efforts, and investing and promoting the future workforce that will bring new ideas to this new landscape.

The present proposal plans to expand this "proactive", rather than corrective, view towards a rich ecosystem that includes: software system specification, validation/verification, quality assurance, and interoperability for the sustainability of node-level programming systems and tools.

1.3 Our Vision

Our vision is to define a work plan to make the access to HPC for science more cost-effective by lowering technical, economical and social barriers to enable sustainability. The proposed community effort will deliver a comprehensive view and study that prioritizes the needs of DOE mission. We propose a coordinated software ecosystem, S4PST, as an investment opportunity to make our programming systems and tools more accessible as we enter an inflection point in HPC. The current uncoordinated, corrective and vendor-driven programming languages, models and ecosystem approach is only expected to become more expensive down the pipeline as more heterogeneous components are deployed in the future landscape of HPC. The current monolithic model is impacted by the end of Moore's Law, energy and economic bounds, and the wealth of data-driven AI workflows. Democratizing access through a sustainable ecosystem will allow the growth of the HPC community to include traditional underrepresented groups while maximizing the nation's strategic computing investments for science.

2 S4PST Objectives / Sustainability Thrust Areas

2.1 Sustainable Community

We will draft a strategy for career development and retention of the workforce behind these efforts [13]. We will enable different levels [14] of quality assurance to reduce currently scattered sustainability costs across users. Through these activities, we will define a project-specific reward mechanism (e.g. badging) to promote not only the sustainability and robustness of a single programming system but also a sustainable improvement of the interoperability of multiple programming systems and their tools. We will also create a venue for inviting emerging programming systems, languages and tools to promote technical inclusiveness in the community. The existing initiatives in the ECP [2, 6] have established a set of policies and standards, improving the software quality for building large production software suites. This approach, however, eliminates the opportunities for the community to explore emerging, yet immature, programming systems and tools. Instead, we will develop policies and reward mechanisms to promote early adapters with new flexible metrics of software quality assurance that clarify risks and mitigation to the PST community and users.

2.2 Community-Wide Technical Support

We will define a set of efforts to accomplish the goals for a sustained node-level PST ecosystem, such as i) track implementation and the latest capability, ii) establish vendor and open-source points of contact, iii) facilitate dependency tracking and deployment via package managers (e.g., Spack), iv) track the interoperability of individual languages, programming systems and tools. These types of information are often scattered without any coordination across different supercomputing facilities or small groups of experts. Having organized archives such as tables describing language feature requirements for certain programming systems (such as C++17, specific compiler versions) and interoperability for major libraries (such as MPI) facilitate for quick referencing. We will implement a pilot projects for OpenMP and OpenACC as part of the survey discussed in the proposed work.

2.3 Training and diversity

We propose an aggressive community engagement and training effort, specifically targeting traditionally underrepresented minorities to diversify the pipeline of people. We will ensure that technical and social debts are minimized for future generations while sustaining and expanding the existing solid connections built between DOE and the broader scientific communities. Our goal is to define a training program focused on teaching modern programming techniques to train students and application developers on modern techniques to make code safer (e.g., smart pointer vs. raw pointer), cross-language and cross-platform interoperability, and more importantly principals and best practices of software testing and packaging. These ideas have been implemented in the IDEAS-ECP project [3], and we will further explore opportunities in non-HPC venues as seen in the new scientific computing track of CppCon 2022 Conference [1], and other programming systems conferences such as RustCon [5] and JuliaCon [4].

2.4 Vendor-Independent Verification and Validation Suite

We will craft a strategy to reduce vendor dependence and identify gaps (e.g. bugs) in vendor and S4PST stacks. It is also our goal to continue building synergies with vendors on through early proactive verification and validation engagements that is independent, but complementary, to vendor efforts. We will propose the application of proxy apps [19] to identify important workloads and a hierarchical approach to perform different levels of validation: functionality, accuracy, scalability. The goal is to ensure the DOE's interests for HPC programming systems and tools are represented in open-source, vendor-independent, verification and validation test suites. Such suites will capture important use cases from DOE HPC applications and, for those use cases, check that programming systems and tools conform to relevant programming model standards, support interoperability across multiple programming systems and runtime systems, and support portability across HPC's increasingly heterogeneous hardware ecosystem. Results from these suites will be useful to HPC users, programming systems and tools vendors, hardware vendors, and DOE program manners as they evaluate the suitability of programming systems and tools for DOE's needs. While some relevant suites already exist [15, 16], we will propose efforts to identify gaps, create new suites, and extend existing suites as needed to ensure the DOE's evolving requirements continue to be represented.

2.5 Emerging Technologies

We believe that one of the major *S4PST* objectives should be the definition of mechanisms able to drive (R&D, collaborative, shepherd, ...) efforts to guarantee that DOE's priorities are part of the PST ecosystem. This will help to develop a more robust, functional, and sustainable PST ecosystem. Important DOE's Research Priorities [7, 11, 23], such as performance portability, extreme heterogeneity, automatic verification, among many others, will help to build a valuable long-term PST ecosystem exceeding the current sustainability capacity to contribute significantly in key scientific milestones and transformational discoveries. While DOE, and ASCR specifically, maintains a very important software stack, it is part of ASCR's identity and our mission to continue to propose new ideas and technologies that can be integrated into DOE's portfolio. The value and return of investments for software sustainability on emerging technologies such as modern LLVM-based high-productivity languages and ecosystems (e.g. Julia, Rust, Python/Numba) and modern build systems (e.g. Meson) need to be understood in the DOE context in conjunction with their success in the broader field of computing. This understanding will take into consideration previous PST related efforts, such as the Defense Advanced Research Projects Agency (DARPA) High Productivity Computing Systems (HPCS) program [9]. This is crucial in the convergence of AI + HPC, as AI is driven by different community and business needs not necessarily focusing on the scalability and science mission aspects of DOE's HPC.

3 Proposed Work / Management Plan

3.1 Work Plan

Our planned work (see Figure 1) and structure can be divided in three main activities:

- Outline a plan of action for each identified sustainability thrust area.
- Organize a workshop among community members of S4PST.
- Disseminate our finding in a workshop report and position paper that will serve as a basis for a larger proposal plan for DOE.

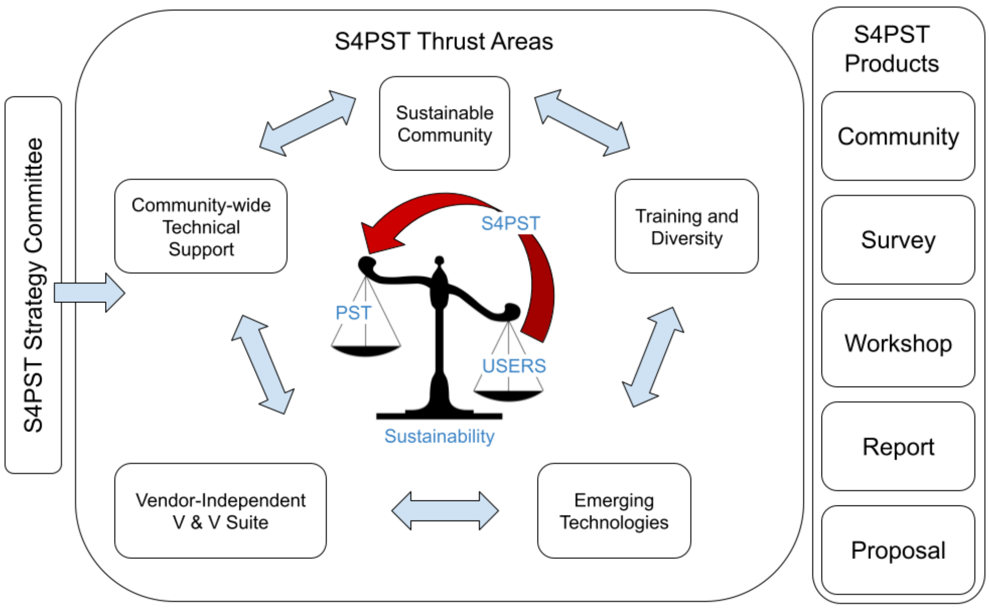

Figure 1: S4PST diagram.

3.1.1 Sustainability Thrust Areas

Our goal is to craft and provide the guidelines for a robust, inter-operable and sustainable PST eco-system and its community to serve the foundation for the applications and software eco-systems at DOE. We will elaborate a cost-effective work plan to address the different thrust areas (already defined) from DOE software stack eco-systems to identify their current and future needs.

3.1.2 Survey

The target of this effort is to identify technical, economical and social requirements for a sustainable PST eco-system. Understanding the landscape is crucial to prioritize detailed aspects of the thrust areas described. The survey will be a comprehensive study on understanding what are the priorities of the DOE HPC community. Our survey covers not only the major programming languages, runtime systems and tools such as C++, Fortran, OpenMP and Kokkos, but also other emerging high-productivity programming languages and systems such as Chapel, Julia, Python/Numba and Rust from the sustainability and productivity viewpoints and the broader landscape of computing driven by AI workflows. The survey also includes the experience from the pilot project of the five major thrusts ideas applied to OpenMP and OpenACC. The outcomes of this survey are:

- Technical requirements: The survey will focus on key areas for software sustainability, such as, hardware and software interoperability and dependencies of PST, important metrics used for sustainability (quality assurance), opportunities in research and development, potential risks and their mitigation.
- Economical requirements: The survey will provide a few economical model of introduction, packaging, testing, documentation and maintenance. In particular, we will focus on the amortization cost for sustainability efforts, such as CI, release schedule, documentation, debug and test support, adoption of new features for the users with a spectrum of proficiency.
- Social requirements: The survey will discuss the identification of PST communities and their current capacities, demography, expectation to users and developers, rewards and incentives for sustainability (e.g. badging and awards), training, diversity and standardization.

The conclusions will be published in a subsequent report and peer-review paper to be delivered to the community.

3.1.3 Workshop

S4PST will fund regular virtual coordination meetings and an all-hands workshop to draft the S4PST roadmap towards a robust, inter-operable and sustained PST eco-system. This workshop will be organized by our university partners. The overall goal is to gather the S4PST community members, discuss survey results, collect our viewpoints on software sustainability and craft a unifying vision for DOE. We expect that most of the funding would be used for this activity after the first half of the funding period with two .

3.1.4 Strategy Committee

Our efforts will be guided by a governance council (see Table 2). This committee, composed of application developers, software developers and vendors, will set priorities, direction and opportunities, trough a governance model that is inclusive of all stakeholders.

3.1.5 Toward FY24 Activities — S4PST Report / Proposal / Community

The success of this project will provide a comprehensive report on PST activities and research involving technical, economical, and social aspects. Based on this report, we will define a work plan including multiple research and activity teams and focus on (1) diverse and inclusive PST community for training and outreach, (2) research on vendor independent software quality assurance for PST, (3) research on software engineering methodology such as automated feature tracking, test-case synthesis and packaging for sustainable PST, (4) research on PST for emerging technology and imminent application needs of DOE, and (5) coordination with other software sustainability research and initiative of DOE. These activities will consolidate the underpinnings of scientific computing research and software development of DOE.

3.2 Management Plan

3.2.1 S4PST team, budget, and coordination

Our proposed *S4PST* project will cover one year as in the expected performance period. As listed in Table 1 along with their institutions, our funded team members are leading experts in the design and implementation of multiple PSTs. The PI for the project, Keita Teranishi, will be responsible for accomplishing all project goals in close coordination with the DOE stakeholders.

In addition, we list the community participants interested in providing their input and advice to our effort that are not necessarily funded in Appendix 6, Table 4. The *S4PST* team is located at Oak Ridge National Laboratory (ORNL), Hewlett Packard Enterprise (HPE), Innovative Computing Laboratory - University of Tennessee at Knoxville (ICL), Codeplay Software (CS), University of Delaware (UD), University of Oregon (UO), Massachusetts Institute of Technology (MIT), Louisiana State University (LSU), National Aeronautics and Space Administration (NASA), Argonne National Laboratory (ANL), Lawrence Livermore National Laboratory (LLNL), Siemens Digital Industries Software (SDIS), Julia Computing Inc. (JCI), NexGen Analytics (NGA), and Lawrence Berkeley National Laboratory (LBNL). Because the team will be

Last name	First name	Position	Institution
Chandrasekaran	Sunita	Associate Professor	UD
Shende	Sameer	Research Professor	UO
Edelman	Alan	Professor and Group Leader	MIT
Blaschke	Johannes	Application Performance Specialist	LBNL
Rouson	Damian	Group Leader	LBNL
Schanen	Michel	Computer Science Specialist	ANL
Laguna	Ignacio	Group Leader	LLNL
Gamblin	Todd	Distinguished Member	LLNL
Kaiser	Hartmut	Adjunct Professor	LSU
Diehl	Patrick	Researcher	LSU
Teranishi	Keita	Group Leader	ORNL
Valero-Lara	Pedro	Computer Scientist	ORNL
Godoy	William	Senior Computer Scientist	ORNL

Table 1: *S4PST* Funded Members and Institutions.

distributed geographically, we will hold monthly project video/teleconferences to ensure progress. We will hold two face-to-face all-hands meetings, at least. Individual task teams will meet approximately monthly to discuss their individual progress.

3.2.2 S4PST governance

The primary focus of the *S4PST* Principal Investigators (PI) and Senior Personnel (SP) is defined in Table 2. Every primary focus (see Section 2) is led by a team. These teams are composed by two members, at least, from different institutions. Some of all these efforts will be part of the deliverable list presented in the previous subsection, for instance, the "survey" effort. Other tasks will be part of the final *S4PST* report, such as "sustainable community", "community-wide tech support", "training and diversity", "vendor-independent verification and validation suite" , and "emerging technologies". The "workshop" task will focus on the organization of the *S4PST* workshop. Finally, the "strategy committee" is composed by the members of the *S4PST* strategy committee, who will participate in all the other tasks. All these efforts will be coordinated by the task leaders of the "*S4PST* project" effort. This effort will focus on accomplishing the different project goals in close coordination with the other task teams and DOE stakeholders. The *S4PST* team is able to create ad hoc tiger teams to address high-priority project topics for the sponsor. We will have mailing lists for the management and entire project team.

3.2.3 Schedule

Table 3 lists the milestones for all *S4PST* activities. Each milestone has an associated deliverable and leader.

Table 3: *S4PST* milestones.

Start of Table 3				
Month	Task #	Milestone	Deliverable	Lead
4	2	*S4PST* Survey	Report	LBNL, ORNL
6	1	*S4PST* half-year all-hands meeting	Event	UO, ICL, ORNL
9	3	*S4PST* Workshop	Event	UO, ICL, ORNL
12	4	*S4PST* Report (Proposal)	Report	ORNL
End of Table 3				

3.3 Broader S4PST Community

Last name	First name	Institution	Role
–*S4PST* project–			
Teranishi	Keita	ORNL	PI
Godoy	William	ORNL	Co-I
Valero-Lara	Pedro	ORNL	Co-I
–Sustainable Community–			
Pébaÿ	Philippe P.	NGA	SP
Doerfert	Johannes	LLNL	SP
Teranishi	Keita	ORNL	PI
–Community-Wide Tech Support–			
Gamblin	Todd	LLNL	SP
Rizzi	Francesco	NGA	SP
Blaschke	Johannes	LBNL	SP
–Training and diversity–			
Rouson	Damian	LBNL	SP
Parete-Koon	Suzanne	ORNL	SP
Diehl	Patrick	LSU	SP
–Vendor-independent verification and validation suite–			
Chandrasekaran	Sunita	UD	SP
Denny	Joel	ORNL	SP
–Emerging Technologies–			
Laguna	Ignacio	LLNL	SP
Schanen	Michel	ANL	SP
Valero-Lara	Pedro	ORNL	Co-I
–Survey–			
Blaschke	Johannes	LBNL	SP
Godoy	William	ORNL	Co-I
–Workshop–			
Shende	Sameer	UO	SP
Teranishi	Keita	ORNL	PI
–Strategy committee–			
Edelman	Alan	MIT	SP
Chapman	Barbara	HPE	SP
Mehrotra	Piyush	NASA	SP
Kaiser	Hartmut	LSU	SP
Vetter	Jeffrey	ORNL	SP

Table 2: *S4PST* PI & SP Personnel.

Last name	**First name**	**Position**	**Institution**
Chapman	Barbara	Director of Programming Environment	HPE
Mehrotra	Piyush	Division Chief	NASA
Anzt	Hartwig	Director	ICL
Wong	Michael	Distinguished Engineer	CS
Chandrasekaran	Sunita	Associate Professor	UD
Shende	Sameer	Research Professor	UO
Edelman	Alan	Professor and Group Leader	MIT
Churavy	Valentin	Research Assistant	MIT
Kaiser	Hartmut	Adjunct Professor	LSU
Diehl	Patrick	Researcher	LSU
Schwinge	Thomas	Software Engineer	SDIS
Kruse	Michael	Computer Scientist	ANL
Monsalve	Jose M.	Postdoctoral Associate	ANL
Schanen	Michel	Computer Science Specialist	ANL
Gamblin	Todd	Distinguished Member	LLNL
Laguna	Ignacio	Group Leader	LLNL
Doerfert	Johannes	Computer Scientist	LLNL
Rouson	Damian	Group Leader	LBNL
Blaschke	Johannes	Application Performance Specialist	LBNL
Shah	Viral	Chief Executive	JCI
Besard	Tim	Software Engineer	JCI
Pébaÿ	Philippe P.	Chief Executive Officer	NGA
Rizzi	Francesco	Chief Technology Oficcer	NGA
Teranishi	Keita	Group Leader	ORNL
Godoy	William	Senior Computer Scientist	ORNL
Valero-Lara	Pedro	Computer Scientist	ORNL
Denny	Joel	Computer Scientist	ORNL
Lee	Seyong	Senior Computer Scientist	ORNL
Gonzalez-Tallada	Marc	Senior Computer Researcher	ORNL
Jin	Zheming	Software Engineer	ORNL
Parete-Koon	Suzanne	High Performance Computing Engineer	ORNL
Vetter	Jeffrey	Corporate Fellow	ORNL

Table 4: *S4PST* Community Members.

Appendix 1 Bibliography and References Cited

References

[1] CppCon web page. cppcon.org. Accessed: 2022-12-11.

[2] E4S Project web page. https://e4s-project.github.io/. Accessed: 2022-12-11.

[3] ECP-IDIAS web page. https://ideas-productivity.org/ideas-ecp/. Accessed: 2022-12-11.

[4] JuliaCon web page. https://juliacon.org/. Accessed: 2022-12-11.

[5] RustCon web page. https://rustconf.com/. Accessed: 2022-12-11.

[6] xSDK web page. https://xsdk.info. Accessed: 2022-12-11.

[7] James Ang, Andrew A. Chien, Simon David Hammond, Adolfy Hoisie, Ian Karlin, Scott Pakin, John Shalf, and Jeffrey S. Vetter. Reimagining codesign for advanced scientific computing: Report for the ascr workshop on reimagining codesign. https://www.osti.gov/biblio/1822199, 4 2022.

[8] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, January 2017.

[9] Jack Dongarra, Robert Graybill, William Harrod, Robert Lucas, Ewing Lusk, Piotr Luszczek, Janice McMahon, Allan Snavely, Jeffrey Vetter, Katherine Yelick, et al. DARPA's HPCS program: History, models, tools, languages. In *Advances in Computers*, volume 72, pages 1–100. Elsevier, 2008.

[10] Jack Dongarra et al. The International Exascale Software Project roadmap. *The International Journal of High Performance Computing Applications*, 25(1):3–60, 2011.

[11] Hal Finkel and Ignacio Laguna. Report of the workshop on program synthesis for scientific computing. https://arxiv.org/abs/2102.01687.

[12] Todd Gamblin, Matthew LeGendre, Michael R. Collette, Gregory L. Lee, Adam Moody, Bronis R. de Supinski, and Scott Futral. The Spack Package Manager: Bringing Order to HPC Software Chaos. Supercomputing 2015 (SC'15), Austin, Texas, USA, November 15-20 2015. LLNL-CONF-6698https://rustconf.com/s90.

[13] Michael A. Heroux. BSSW 4: Improving Reproducibility Through Better Software Practices. https://figshare.com/articles/journal_contribution/Improving_Reproducibility_Through_Better_Software_Practices/5593339, 11 2017.

[14] Michael A. Heroux. Productivity Initiative. https://github.com/trilinos/Trilinos/wiki/Productivity---Initiative, 1 2019.

[15] Thomas Huber, Swaroop Pophale, Nolan Baker, Michael Carr, Nikhil Rao, Jaydon Reap, Kristina Holsapple, Joshua Hoke Davis, Tobias Burnus, Seyong Lee, David E. Bernholdt, and Sunita Chandrasekaran. Ecp sollve: Validation and verification testsuite status update and compiler insight for openmp, 2022.

[16] A. M. Jarmusch, A. Liu, C. Munley, D. Horta, V. Ravichandran, J. Denny, and S. Chandrasekaran. Analysis of validating and verifying openacc compilers 3.0 and above, 2022.

[17] C. Lattner and V. Adve. LLVM: a compilation framework for lifelong program analysis amp; transformation. In *International Symposium on Code Generation and Optimization, 2004. CGO 2004.*, pages 75–86, 2004.

[18] Nicholas D Matsakis and Felix S Klock. The Rust Language. *ACM SIGAda Ada Letters*, 34(3):103–104, 2014.

[19] Satoshi Matsuoka, Jens Domke, Mohamed Wahib, Aleksandr Drozd, Andrew A. Chien, Raymond Bair, Jeffrey S. Vetter, and John Shalf. Preparing for the future—rethinking proxy applications. *Computing in Science Engineering*, 24(2):85–90, 2022.

[20] Manish Parashar, Salim Hariri, Tomasz Haupt, and Geoffrey Fox. A study of software development for high performance computing. In Karsten M. Decker and René M. Rehmann, editors, *Programming Environments for Massively Parallel Distributed Systems*, pages 107–116, Basel, 1994. Birkhäuser Basel.

[21] Daniel Reed, Dennis Gannon, and Jack Dongarra. Reinventing High Performance Computing: Challenges and Opportunities, 2022.

[22] Dominik Straßel, Philipp Reusch, and Janis Keuper. Python workflows on hpc systems. In *2020 IEEE/ACM 9th Workshop on Python for High-Performance and Scientific Computing (PyHPC)*, pages 32–40, 2020.

[23] Vetter, Jeffrey S. et al. . Extreme Heterogeneity 2018 - Productive Computational Science in the Era of Extreme Heterogeneity: Report for DOE ASCR Workshop on Extreme Heterogeneity. https://www.osti.gov/biblio/1473756, 12 2018.