# Approaching Exascale: Best Practices for Training a Diverse Workforce using Hackathons

Izumi Barker
NVIDIA
Santa Clara, CA
ibarker@nvidia.com

Mozhgan Kabiri Chimeh
NVIDIA
Santa Clara, CA
mozhgank@nvidia.com

Kevin Gott
National Energy Research Scientific Computing Center
Berkeley, CA
kngott@lbl.gov

Thomas Papatheodore
Oak Ridge National Laboratory
Oak Ridge, TN
papatheodore@ornl.gov

Mary P. Thomas
University of California San Diego
La Jolla, CA
mpthomas@sdsu.edu

## ABSTRACT

Given the anticipated growth of the high-performance computing market, HPC is challenged with expanding the size, diversity, and skill of its workforce while also addressing post-pandemic distributed workforce protocols and an ever-expanding ecosystem of architectures, accelerators, and software stacks. As we move toward exascale computing, training approaches need to address how to best prepare future computational scientists and enable established domain researchers to stay current and master tools needed for exascale architectures. This paper explores adding hybrid and virtual Hackathons to the training mix to bridge traditional programming curricula and hands-on skills needed among diverse communities. We outline current learning and development programs available; explain the benefits and challenges in implementing hackathons for training using experience gained from the Open Hackathons program (formerly the GPU Hackathons program); discuss how to engage diverse communities—from early career researchers to veteran scientists; and recommend best practices for implementing these events.

## KEYWORDS

HPC, Exascale, Hackathons, HPC Training, HPC Education

## 1 INTRODUCTION

The potential for high-performance computing (HPC) to accelerate science is limitless, making it essential to much of the research activities across academia, supercomputing centers, government laboratories, and industry. As the landscape of research changes, large scientific projects can no longer advance in isolation but are dependent on community-driven participation. This necessitates the need for scalability of data processing and analysis, input/output capabilities that match pace with computational capabilities, and sufficiently performance-portable and expressive programming models that can handle the ever-growing volume, complexity, and rapidity of current and future data sets. [7]

The overall outlook for the HPC market is strong. Growing at an overall market compounded annual growth rate (CAGR) of 6.9 percent, Hyperion Research reported HPC spending (on-premise, cloud, and AI) for 2021 neared $35 billion (USD) and is on track to reach nearly $40 billion in 2022 and $50 billion by 2026. The rise of exascale and near-exascale systems has also seen tremendous growth, increasing from one near-exascale system in Japan in 2020 to five to eight exascale systems predicted by 2026 [9].

With the anticipated growth of HPC into exascale regions for both scientific computing and the broader enterprise, HPC is feeling the pressure of recruiting and retaining people. It faces the quandary of expanding the size, diversity, and skill of its workforce while simultaneously facing an expertise shortage. This scarcity of HPC experts is driven by several factors, such as the outflow of retirees exceeding the pipeline of new HPC staff, an increasing number of HPC sites worldwide, and the rising complexity of existing sites utilizing emerging technologies (i.e., AI, cloud, GPUs and other accelerators) that require different skill sets and leading to more systems per site [11].

As we move forward in exascale computing we must ask: How can we improve recruitment and better prepare future computational scientists for the upcoming challenges in exascale computing? How do we enable established domain researchers to stay current with the latest software and hardware trends and master the tools needed for the newer compute node architectures? How do we make exascale and HPC more accessible?

Traditionally, HPC has had a high barrier to use, owing in no small part to the shortfall of available expertise. Numerous training and development modalities exist, but often are independent of each other, lack standardization, or fail to incorporate real-world concepts and applications. Adding in-person and virtual hackathons to the training mix can bridge traditional programming curricula and hands-on skills needed among the diverse communities across national laboratories, supercomputing centers, and academic environments.

Hackathons and coding bootcamps have evolved from early coding and “bug discovery” sessions to become modern innovation events that combine agile programming and intense mentoring. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2023 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/14/1/3

collaborative approach of these events provides the critical accelerated computing skills needed by the scientific community and the professionals that support them and aids in preparing researchers to use current and upcoming supercomputing resources.

This paper explores adding in-person and virtual hackathons to currently available learning and development programs, outlines the benefits and challenges of these events; discusses community impact and engagement, and finally, recommends best practices for implementing hackathons for ongoing and sustainable development.

## 2 BACKGROUND: CURRENT TRENDS IN HPC TRAINING AND EDUCATION TODAY

HPC concepts have been taught in academic settings, through informal webinars and tutorials at HPC Centers, or self-taught on an "as-needed" basis." In traditional academia, HPC content is interwoven within accredited computer science, information science, or computer engineering degree programs. For students pursuing research in other disciplines that require significant computing resources, HPC education may be integrated into courses in a student's subject domain (i.e., physics) but the number of institutions offering HPC coursework is low [14]. Additional challenges such as the diversity and complexity of the subject domains and the limited or varied computer literacy of the students only serve to compound the problem.

Modeling and simulation, now so ubiquitous, have led to emerging fields such as Computational Science and Engineering (CSE). Combining computer sciences, applied mathematics and statistics, and domain sciences, CSE's multidisciplinary approach encompasses methods of HPC and has become a cornerstone for the development and use of computational methods for scientific discovery [27]. While the number of CSE courses and programs has grown, the overall availability is low as is the number of students pursuing this area of study or graduating from these programs. Current coursework fails to expose students to real-world applications thus limiting a true understanding of the complexities of the field, preventing the development of skills needed for modern scientific and technological enterprises, and inadequately preparing students to fully utilize powerful new supercomputers for scientific applications and innovation. Moreover, almost no universities have a curriculum specifically focused on exascale or petascale science as issues are largely unknown and unexplored [3].

HPC education is also commonly taught as brief, condensed workshops lasting a half-day to several days or through specialized training modules and events. These workshops are offered by a diverse ecosystem of providers, but, whether it is an institution looking to shore up the skills of their existing staff, a government initiative aimed at ensuring the country continues critical research or a professional organization dedicated to a specific area of practice, the explicit goal is to develop a workforce with HPC-specific skills.

A variety of training options are available, ranging from webinars, lectures, Massively Open Online Courses (MOOCs) [15], hands-on labs and tutorials, software carpentries [29], on-the-job and specialized events among others. These options are often disparate, unrelated, and not universally standardized. Many of these training activities are executed in accordance with specific projects or agendas that may or may not continue, such as in the case of XSEDE which concluded formal operations as a National Science Foundation (NSF)-funded project in August 2022 [30], or the Exascale Computing Project (ECP), a component of the DOE-led Exascale Computing Initiative (ECI), which is moving to completion in mid-2023. An overview of the Training Efforts in the Exascale Computing Project can be found in the paper by Marques and Barker [12].

Additional challenges to workshops and training include a limited pool of qualified and available trainers, a finite number of workshops and training activities offered for a given region within the calendar year, the efficacy of the training materials and modalities to meet individual learning needs at scale, and finally, training materials may not be appropriate for the individual's specific project or timeframe.

## 3 LEVERAGING HACKATHONS FOR TRAINING

Complex scientific challenges and priority research is pushing the increased demand for extreme-scale computational resources to support a range of workflows for modeling, simulation, and data analysis that enable new discoveries and new understandings. The role of software that is reusable is central to research; however, many of the software libraries and scientific codes have largely been developed organically and maintained by a diverse community without considering longer-term sustainability that supports interdisciplinary collaboration nor addresses rapidly changing computing architectures [10].

Hackathons are fixed-time events during which individuals form teams and intensively collaborate to advance or complete a specific project of interest. [24] Believed to be coined during an OpenBSD cryptographic development event in 1999 [13], the meaning and nature of these events have developed from early ad-hoc exploratory programming sessions to represent modern innovation events that offer new opportunities for cooperative research and scientific discovery. Growing in both popularity and success, hackathons foster learning, drive community engagement, increase networking and relationship-building, and are effective for addressing civic, environmental, and public health issues, leading to increased adoption across various fields from higher education to healthcare to business services [8] [24].

For the purpose of this paper, we focus on Open Hackathons (formerly GPU Hackathons) [22], which are managed by the OpenACC Organization [23] and are designed to help scientists, researchers, and developers accelerate and optimize their applications on a variety of data center architectures, enabling them to build the critical skills needed to take advantage of modern HPC compute resources. Started as a one-off training activity in partnership with Oak Ridge Leadership Computing Facility [20] and NVIDIA [4] in 2014, Open Hackathons have evolved into a global program with over 100 hackathon events executed worldwide since the program launch.

### 3.1 Benefits

Leveraging Open Hackathons to support HPC training initiatives offers benefits to attendees, the hosting organizations, and the community at large. For hackathon attendees, the first and foremost benefit is training and skills development. Where both academic settings and workshop/training offerings have challenges that perpetuate difficulties in meaningful HPC training at scale, Open Hackathons can address many of these limitations systematically.

At Open Hackathons, domain scientists are paired with experienced programming experts to receive dedicated guidance and mentorship for the course of the event. Attendees work with their mentors to strategically develop realistic goals for their codes and receive targeted recommendations and training in the HPC tools and resources relevant to those goals, allowing them to build hands-on skills such as learning how to compile their applications to identify computational bottlenecks or trying a new library or framework for a new approach to optimizing their code in short order.

Attendees are also given access to large heterogeneous HPC compute clusters that they may not otherwise have access to, for example, Ascent [21] the stand-alone 18 node system at OLCF with the same architecture and design as their Summit supercomputer, ranked in the top five of the Top500 list of most powerful commercially available computer systems since its debut in 2018 [25]. This enables an immersive experience that mirrors real-world environments so that attendees, particularly those who are students or early career researchers, can learn to navigate through many issues related to scalability, parallel efficiency, heterogeneous computing, parallel storage systems, and other issues [26]. Other computing platforms used during Open Hackathons include Cori at the National Energy Research Scientific Computing Center (NERSC), HiPerGator at the University of Florida, Juwels Booster at Forschungszentrum Juelich (FZJ), Piz Daint at the Swiss National Supercomputing Centre (CSCS), and Cirrus powered by EPCC [1] to name a few.

Since hackathon formats are intrinsically geared toward interdependent work, attendees benefit from collective knowledge sharing and increased opportunities for networking and recruitment as teams gain visibility to active projects and peers in different institutions and domains.

With the growing range of HPC workflows needing support and global exascale systems representing an $11 to $15 billion (USD) investment, it is imperative that HPC organizations compel full utilization of their existing systems and judiciously plan and prepare for upcoming needs. The motivations for organizations to host a hackathon are different from attendees; however, the benefits are aligned, focusing on skills development of their talent, system preparedness and utilization, talent recruitment, and competitiveness. To that end, Open Hackathons can assist.

Developing staff ability is paramount, since researchers cannot fully take advantage of computing systems without possessing the needed skills to do so. Hackathons help to facilitate quick and efficient skill-building through mentor engagement and guidance, hands-on team collaboration, and collective knowledge sharing. This is particularly effective, since the participants are actively working on their own specific codes or projects at the hackathon and therefore are deeply invested.

Many hackathons utilize the host organization's own compute cluster. This serves to assist staff in becoming more comfortable with the institution's available architectures and able to use new tools and techniques that allow them fully utilize the resource. Additionally, hackathons can help host institutions prepare for future system needs by giving them point-in-time snapshots of current research projects and their related applications across different domains of science, allowing the host to discover trends in the aggregate data and plan accordingly.

Lastly, as these events are most often open to the scientific community to participate in without regard to the participant's affiliation, hosting institutions can leverage hackathons to network, recruit new talent, as well as new users and projects of interest for computing allocation on their systems.

### 3.2 Open Hackathon Challenges

While there are numerous benefits to implementing Open Hackathons to augment HPC training, there are also challenges that need to be evaluated, including attendee preparedness, mentor availability and engagement, and system limitations.

Most Open Hackathon events are largely open to the scientific community for participation. This attracts a diverse applicant and attendee pool with varying levels of both domain-specific and technical skills and experience. This can lead to behaviors that affect participation, team dynamics, and overall outcomes. Applicants that are students or in early careers may feel intimidated or less able to fully contribute or participate without significant mentoring while more senior or seasoned attendees may be resistant to suggestions or new approaches. These behaviors affect team dynamics, impede progress and lead to lower satisfaction and learning outcomes.

Volunteer mentors are crucial to hackathon success: their expert guidance is needed to bridge gaps between domain knowledge and programming demands. With so many programming languages, libraries and frameworks, software development kits, and hardware choices available or utilized by the hackathon applications, having a large enough pool of qualified mentors can present a challenge when implementing a hackathon program. Given the intensive nature of hackathons, mentors must be available, engaged, and committed for the entirety of the event which can also pose challenges as they balance competing work priorities and schedules, particularly if they are affiliated outside the host organization.

Hackathons are great opportunities for training researchers on existing systems or helping to plan for upcoming system needs. They can also help stress test and evangelize systems that are newly online; however, this poses a challenge as well. A hackathon host should carefully consider the availability and "readiness" of the compute resource intended for the hackathon. Systems must be configured properly, be available and have adequate storage for the teams for the duration of the event, and provide any necessary information or instructions for access, containers, workflows and software stacks. Cluster support expectations should be understood and communicated. We have found that it's often best to have multiple systems available to mitigate risk associated with outages, new test systems, and other issues.

## 4 COMMUNITY IMPACT

The close integration of HPC simulation and data analysis continues to feed the development of new computer architectures and workflows, specialized software, and the growth of interdisciplinary teams, which are becoming more and more important for today's

HPC computing and scientific research efforts. Interdisciplinary research (IDR) is loosely defined as an effort conducted by teams that integrates information, data, techniques, tools, perspectives, and concepts from multiple disciplines to solve problems whose solutions are beyond the scope of a single discipline or area of research practice [19]. IDR is emerging to be a key concept of "convergence research," which is one of the NSF's "10 Big Ideas" for 2022 [6] and which the DOE Office of Science has made a Priority Research Area for SCGSR 2022. [18] Leveraging Open Hackathons as auxiliary HPC training programs will have a considerable, lasting impact on the research and development community by growing an interdisciplinary community, assisting with creating sustainable code, driving computer resource allocation, and advancing research. As a result, the Open Hackathon program has the potential to impact interdisciplinary scientific research and development.

The ethos of hackathons is collaboration and implementing these cooperative events for training grows the community by establishing wide-reaching, interconnected relationships between researchers and their projects. Participants can learn new perspectives, practices and technologies from each other, their mentors and their peers and are able to try new approaches in a safe environment. Strategic networks developed at these events can be instrumental for broader interdisciplinary knowledge exchange as well as raising the visibility of new collaboration opportunities and recruiting activities which is helpful for attracting new generations of HPC practitioners.

Scientific software is vital to research but faces difficulties in that it relies on an active community for continued development and distribution, but this community-driven approach can lead to an ecosystem of competing and collaborating products [10] as different contributors add to the codebase based on their own projects. Additionally, a sustainable approach to developing scientific software is sometimes overlooked by domain researchers as the focus is on publishing and not necessarily creating software [28]. As researchers work with mentors during the hackathon, not only is there a significant contribution to the code base but also an increased likelihood of a portable, production-ready, and sustainable code that can readily be used by the community since mentors are experienced programming experts and well-versed in creating reproducible, documented codes.

Open Hackathons increase community access to large-scale supercomputing systems enabling researchers and also act as feeders for additional initiatives and programs at hosting institutions as they solicit project proposals to allocate computing resources and cycles. One such example is the Innovative and Novel Computational Impact on Theory and Experiment, or INCITE program, jointly managed by Argonne Leadership Computing Facility and the Oak Ridge Leadership Computing Facility (OLCF) that awards allocations of supercomputer access to high-impact computational science projects across multiple disciplines [5]. Additionally, teams continuing work on large projects have participated in more than one hackathon, allowing them to access different compute systems (i.e., Ascent from ORNL [21] and Cori from NERSC [16]) aiding in scalability studies and comparisons.

Lastly, hackathons connect researchers to the right tools and technologies within an environment conducive to collaborative innovation and rapid optimization, making them very useful to advance research projects. To date, over 100 hackathons using this approach have been run worldwide and more than 550 scientific applications across multiple scientific domains have been accelerated wholly or in part at Hackathons. Examples include BerkeleyGW, Quantum ESPRESSO, CASTRO, Gkeyll, QUICK, CASTEP, and NWChem/NWChemEx. For additional information, please refer to the published paper: Best Practices in Running Collaborative GPU Hackathons: Advancing Scientific Applications with a Sustained Impact [2].

## 5 BEST PRACTICES FOR IMPLEMENTING HACKATHONS

Based on our experiences, we propose the following best practices in order to maximize the success and outcomes of hackathons and other training events.

### 5.1 Event Format

The Open Hackathon format centers around some guiding principles, including:

- detailed team application process involving hackathon hosts and Open Hackathon organizers to verify team capabilities and appropriate model to be studied,
- a minimum number of team members working on the same code to ensure a broader developer base behind the code [2],
- defined team goals for the hackathon [2], and finally,
- an approach that pairs teams of researchers with mentors and programming experts who are often experienced in the scientific domain.

An application process is utilized for participation in the hackathons where detailed information is collected, including code information such as programming language, programming model, algorithmic motif, code license, as well as team goals for hackathon and team members. Applications are reviewed by a jury composed of the host institution and program organizers in order to select those applications that 1) have a high impact project or code with domain relevance, 2) are technically feasible for the hackathon event with codes that are properly licensed, reproducible and documented, and 3) can be practically supported by available mentors in the network.

Most hackathon events run for a total of approximately five days; however, these days are not sequential but are separated over the course of two weeks to promote manageable and meaningful progress. "Day 0" occurs two weeks before the main hackathon event and introduces the team members and mentors, discusses the code, goals, and possible strategies to achieve these goals, and sets expectations between the participants. Day 0 also provides an overview of useful online tools as well as instructions for compute cluster access. "Day 1" occurs one week prior to the main event and introduces the participating teams, introduces all the mentors and their area of expertise, provides an overview of each project and code, and provides brief tutorials on the cluster, main tools such as profilers and libraries, and Q&A opportunities to encourage dialogue and knowledge sharing. The remainder of the hackathon ("Days 2-4") occurs during the last week where teams and mentors work collaboratively, loosely applying agile methodology and presenting progress in daily stand-up scrum sessions.

Small adjustments can be made dependent on whether the event is in-person or virtual, but this format balances flexibility and discipline for optimal progress.

## 5.2 Team Composition and Preparedness

The guiding principles are coupled with careful team selection and preparation to make sure that teams make progress throughout the event and beyond.

In terms of team composition, we have found that three to five members is the ideal number for hackathon participation, permitting equitable division of work without too much “down” time. All team members should be fluent in the code they are working on and committed to completely participating for the duration of the hackathon event. Lastly, while team composition can vary greatly–from students to senior scientists, from little to no GPU or accelerator programming skills to advanced CUDA or language fluency–a balanced mix produces the best outcomes. For teams composed of all students, a principal investigator or advisor should be tasked with supervising and regular check-ins to keep goals aligned.

The more prepared the team is, the better the experience and ultimately, progress. There are many tools available to prepare for the hackathon and we recommend that teams take advantage of these to the extent possible. Focused training can shore up knowledge gaps and provide attendees with fundamental understanding of techniques that will be used during the hackathon. Targeted bootcamps are short-format training events that teach basic skills in specific topics (i.e., how to accelerate a code via various programming models) through a combination of lectures and hands-on activities using mini-applications in a controlled environment. These training events help attendees to quickly gain introductory programming skills that they can apply to a real code, increasing their confidence and readiness to participate in the hackathon. Finally, to maximize time with mentors, teams are encouraged to profile their codes ahead of time so that computational bottlenecks are known and can actively be addressed.

## 5.3 Team Mentors

Mentor pairing is one of the most critical components for the success of a team. Mentors should be assigned to teams based on several factors, including their core competencies, skill level, expertise and work style, and we recommend two mentors per team for most hackathons but this is flexible depending on the experience of the mentor and the complexity of the code/project.

Successful mentors have both the technical skills and soft skills. From a technical perspective, mentors are experienced programmers with core competencies in a specific programming language or model and who oftentimes also have domain-specific knowledge. This helps the mentor to understand the context of the problem statements and offer guidance that is specific to the goals of the team, and helps the team have confidence in the mentor and builds trust. Soft skills are also important to facilitating open communication, increasing receptiveness to coaching and keeping teams focused and on-track. For optimal outcomes, mentors should reinforce the focus on learning and development rather than “project completion,” remaining in a mentoring role as opposed to project stakeholder. [17] This mentoring role mindset helps guide mentor interaction, informing mentors when to get “hands on” such as helping with small code samples, showing integration in the main code base, or profiling; and when to step back—allow team members to problem-solve or write code themselves to become self-sufficient.

Lastly, providing mentors with training and support helps increase success and satisfaction. Open Hackathons provides a variety of training options for mentors, including online courses, tutorials, industry training modules, peer-to-peer shadowing, and a mentor certification program.

# 6 CONCLUSIONS

High-performance computing (HPC) is critical to the continued advancement of science. As we approach the era of exascale computing, technology changes are creating opportunities and challenges, necessitating broadened approaches to training and developing the next generation of HPC users to be able to realize the full potential of emerging computing systems and architectures. Integrating hackathons into the training mix can bridge traditional programming curricula with real-world, hands-on skills to address the wide spectrum of computational needs and aptitudes and help stem the HPC talent shortfall.

# ACKNOWLEDGMENTS

The authors would like to acknowledge Sunita Chandrasekaran, Guido Juckeland, Jack Wells, and Julia Levites for their continued leadership and support of the Open Hackathons program. Additionally, we would like to thank OpenACC-Standard.org, NVIDIA, and our extensive partner network for their support, including: Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725; National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231; Helmholtz-Zentrum Dresden Rossendorf, Jülich Forschungszentrum, Swiss National Supercomputing Centre; Brookhaven National Laboratory; the San Diego Supercomputer Center under NSF awards for Expanse (#1928224) and the Extreme Science and Engineering Discovery Environment(XSEDE) (#ACI-1548562), among others.

# REFERENCES

[1] Edinburgh Parallel Computing Center. 2022. EPCC Cirrus System Page. Retrieved September 24, 2022 from https://www.cirrus.ac.uk/
[2] Sunita Chandrasekaran, Guido Juckeland, Meifeng Lin, Matthew Otten, Dirk Pleiter, John E. Stone, Juan Lucio-Vega, Michael Zingale, and Fernanda Foertter. 2018. Best Practices in Running Collaborative GPU Hackathons: Advancing Scientific Applications with a Sustained Impact. *Computing in Science & Engineering* 20, 4 (2018), 95–106. https://doi.org/10.1109/MCSE.2018.042781332
[3] Barbara Chapman, Henri Calandra, Silvia Crivelli, Jack Dongarra, Jeffrey Hittinger, Scott A. Lathrop, Vivek Sarkar, Eric Stahlberg, Jeffrey S. Vetter, and Dean Williams. 2014. DOE Advanced Scientific Advisory Committee (ASCAC): Workforce Subcommittee Letter. (7 2014). https://doi.org/10.2172/1222711
[4] NVIDIA Corporation. 2022. NVIDIA Home Page. Retrieved September 24, 2022 from https://www.nvidia.com/
[5] Department Of Energy. 2022. DOE INCITE Program. Retrieved September 24, 2022 from https://www.doeleadershipcomputing.org/
[6] National Science Foundation. 2022. Learn About Convergence Research. Retrieved September 24, 2022 from https://GOOGLE.COM

[7] Richard Gerber, James Hack, Katherine Riley, Katie Antypas, Richard Coffey, Eli Dart, Tjerk Straatsma, Jack Wells, Deborah Bard, Sudip Dosanjh, Inder Monga, Michael E. Papka, and Lauren Rotman. 2018. Crosscut report: Exascale Requirements Reviews, March 9–10, 2017 – Tysons Corner, Virginia. An Office of Science review sponsored by: Advanced Scientific Computing Research, Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High Energy Physics, Nuclear Physics. (1 2018). https://doi.org/10.2172/1417653

[8] Ahmed Imam, Tapajit Dey, Alexander Nolte, Audris Mockus, and James D. Herbsleb. 2021. The Secret Life of Hackathon Code Where does it come from and where does it go?. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. 68–79. https://doi.org/10.1109/MSR52588.2021.00020

[9] Hpyerion Research Joseph E. Sorensen. 2022. HPC Market Update Briefing During ISC22. Retrieved September 24, 2022 from https://hyperionresearch.com/hpc-market-update-briefing-during-isc22/

[10] Daniel S. Katz, Lois Curfman McInnes, David E. Bernholdt, Abigail Cabunoc Mayes, Neil P. Chue Hong, Jonah Duckles, Sandra Gesing, Michael A. Heroux, Simon Hettrick, Rafael C. Jimenez, Marlon Pierce, Belinda Weaver, and Nancy Wilkins-Diehr. 2019. Community Organizations: Changing the Culture in Which Research Software Is Developed and Sustained. *Computing in Science & Engineering* 21, 2 (2019), 8–24. https://doi.org/10.1109/MCSE.2018.2883051

[11] Hpyerion Research M. Riddle. 2022. ISC22 Market Update - HPC Talent Challenges. Retrieved September 24, 2022 from https://hyperionresearch.com/wp-content/uploads/2022/05/Hyperion-Research_ISC22-Market-Update_HPC-Talent-Challenges.pdf

[12] Osni Marques and Ashley Barker. 2020. Training Efforts in the Exascale Computing Project. *Computing in Science & Engineering* 22, 5 (2020), 103–107. https://doi.org/10.1109/MCSE.2020.3010596

[13] P. Mittai. 2022. A brief history of hackathon. Retrieved September 24, 2022 from https://content.techgig.com/codegladiators2021/a-brief-history-of-hackathon/articleshow/75291637.cms

[14] Julie Mullen. 2020. Interactivity, Engagement and Community Building in Online HPC Education and Training. Retrieved September 24, 2022 from https://sc20.supercomputing.org/proceedings/sotp/sotp_files/sotp124s2-file2.pdf

[15] Julia Mullen, Weronika Filinger, Lauren Milechin, and David Henty. 2019. The Impact of MOOC Methodology on the Scalability, Accessibility and Development of HPC Education and Training. *The Journal of Computational Science Education* 10, 1 (2019), 67–73. https://doi.org/10.22369/issn.2153-4136/10/1/11

[16] National Energy Research Scientific Computing Center (NERSC). 2022. NERSC Cori User Guide. Retrieved September 24, 2022 from https://www.nersc.gov/systems/cori/

[17] Alexander Nolte, Linda Bailey Hayden, and James D. Herbsleb. 2020. How to Support Newcomers in Scientific Hackathons - An Action Research Study on Expert Mentoring. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 25 (may 2020), 23 pages. https://doi.org/10.1145/3392830

[18] DOE Office of Science. 2022. Office of Science Priority Research Areas for SCGSR Program. Retrieved September 24, 2022 from https://science.osti.gov/wdts/scgsr/How-to-Apply/Priority-SC-Research-Areas

[19] National Academy of Science. 2005. Facilitating Interdisciplinary Research. Retrieved September 24, 2022 from https://www.doeleadershipcomputing.org/

[20] Oak Ridge Leadership Computing Facility (OLCF). 2022. Oak Ridge Leadership Computing Facility Home Page. Retrieved September 24, 2022 from https://docs.olcf.ornl.gov/systems/ascent_user_guide.html

[21] Oak Ridge Leadership Computing Facility (OLCF). 2022. OLCF Ascent User Guide. Retrieved September 24, 2022 from https://www.olcf.ornl.gov/

[22] OpenACC Organization. 2022. Open Hackathons Program. Retrieved September 24, 2022 from https://www.openhackathons.org/s/

[23] OpenACC Organization. 2022. OpenACC Organization Home Page. Retrieved September 24, 2022 from https://www.openacc.org/

[24] Ei Pa Pa Pe-Than, Alexander Nolte, Anna Filippova, Christian Bird, Steve Scallen, and James D. Herbsleb. 2019. Designing Corporate Hackathons With a Purpose: The Future of Software Development. *IEEE Software* 36, 1 (2019), 15–22. https://doi.org/10.1109/MS.2018.290110547

[25] Top 500 Project. 2022. Top 500 List. Retrieved September 24, 2022 from https://www.top500.org/lists/top500/

[26] Rajendra K. Raj, Carol J. Romanowski, John Impagliazzo, Sherif G. Aly, Brett A. Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I. Gordon, Cruz Izu, Shahram Rahimi, Michael P. Robson, and Neena Thota. 2020. High Performance Computing Education: Current Challenges and Future Directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 51–74. https://doi.org/10.1145/3437800.3439203

[27] Ulrich Rüde, Karen Willcox, Lois Curfman McInnes, and Hans De Sterck. 2018. Research and education in computational science and engineering. *SIAM Rev.* 60, 3 (2018), 707–754. https://doi.org/10.1137/16M1096840 arXiv:1610.02608

[28] Erik H. Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D. Herbsleb. 2014. Community Code Engagements: Summer of Code & Hackathons for Community Building in Scientific Software. In *Proceedings of the 18th International Conference on Supporting Group Work (GROUP '14)*. Association for Computing Machinery, New York, NY, USA, 111–121. https://doi.org/10.1145/2660398.2660420

[29] Greg Wilson. [n. d.]. Software Carpentry web site. http://software-carpentry.org. Main web site for Software Carpentry, replacing http://swc.scipy.org.

[30] XSEDE. 2022. XSEDE Project Home Page. Retrieved September 24, 2022 from https://www.xsede.org/