Introduction to Volume 12 Issue 2: Special Issue on HPC Training and Education

Nitin Sukhija
Guest Editor
Slippery Rock University of Pennsylvania
Slippery Rock, PA

FOREWORD

High performance computing is becoming central for empowering scientific progress in the most fundamental research in various science and engineering, as well as societal, domains. It is remarkable to observe that the recent rapid advancements in today's and future computing and software environments provide both challenges and opportunities for cyberinfrastructure facilitators, trainers, and educators to develop, deliver, support, and prepare a diverse community of students and professionals for careers that utilize high performance computing to advance discovery. This special issue focuses on original research papers submitted to the Second Workshop on HPC Education and Training for Emerging Technologies (HETET20), which was held in conjunction with the ISC20 Digital conference, June 25, 2020; the third Workshop on Strategies for Enhancing HPC Education and Training (SEHET20), which was held in conjunction with the PEARC20 conference, July 27, 2020; and the Seventh SC Workshop on Best Practices for HPC Training and Education (BPHTE20), which was held in conjunction with the SC20 conference, November 11, 2020.

This special issue begins with an article by Purwanto et al. that presents a non-degree computational training program, DeapSECURE, that provides significant high-performance computing (HPC) and big-data foundations for cybersecurity students. The authors detail major improvements of the DeapSECURE lesson modules by grouping them into the "compute-intensive" and "data-intensive" categories, more tightly integrating the modules to streamline the learning experience. The assessment results of the cohort group trained indicate the need for further adjustments to improve learning experience and outcome. Moreover, the piloted workshop showed great promise to address some challenges encountered through the second year project.

The article by Chakravorty et al. reports on two educational approaches that were implemented in the informal program hosted by Texas A&M High Performance Research Computing (HPRC) in the Spring, Summer, and Fall semesters of 2020. The two approaches employed were: 1) traditional in-person sessions taught by the staff that focused on offering a lot of information online and 2) a primer approach involving a peer-learning environment engaging learners via shorter pop-up courses in which participants chose the topic matter and students taught and moderated the training sessions. These were supplemented with YouTube videos and continued engagement over a community Slack workspace. The authors conclude by highlighting the data collected as part of this study, indicating that the Primer format could be a suitable pedagogical approach that enhances learner engagement while scaling back on staff time.

The article by Dey et al. describes the efforts taken at the Texas Advanced Computing Center to develop a successful academic and training curriculum with the goal of making virtual classrooms more engaging, and more collaborative, thus delivering a better educational experience. The authors report on the approach to teaching with multiple instructors and integrating aspects of gamification, open curriculum, casual classroom, and flipped classroom along with spending more class time focused on applying learned concepts versus lecturing on concepts, resulting in much-needed teacher-student interaction to create a positive learning environment.

The article by Bungo and Wong describes the NVIDIA Deep Learning Institute (DLI) kits that offer a complete course solution to lower the barrier of incorporating AI and GPU computing in the classroom. The authors discuss the DLI University Ambassador Program that enables qualified educators to teach DLI workshops at no cost across campuses and academic conferences to faculty, students, and researchers. The authors conclude by illustrating real examples of leading academics leveraging Teaching Kits and Ambassador workshops in the classroom.

The article by Weeden describes the XSEDE EMPOWER (Expert Mentoring Producing Opportunities for Work, Education, and Research) program coordinated by the Shodor Education Foundation for the Extreme Science and Engineering Discovery Environment (XSEDE). The author discusses the goal of the program, which is to engage a diverse group of undergraduate students in the work of XSEDE, matching them with faculty and staff mentors who have projects that make use of XSEDE services and resources or that otherwise prepare students to use these types of services and resources. The author concludes by discussing the impact of the program on advancing careers and conference participation of the underrepresented undergraduate students.

The article by Backhaus et al. presents the challenges faced by the Pawsey Supercomputing Centre in making transition to virtual training, including how to creatively motivate and engage learners, build our virtual training delivery skills, and build communities across Australia. The authors detail the self-guided learning, using Nimbus cloud and containers for improving the training content, ensuring alignment with learning objectives and learning outcomes, and incorporating best practices in (virtual) learner interaction and engagement. The authors conclude by discussing that there is no universal, one-size-fits-all learning solution to address virtual training challenges and there exist various solutions and platforms that need to be carefully selected for different groups of learners.

Chen summarizes the development of a set of HPC courses to meet the needs of multidisciplinary students at the National University of Defense Technology. The courses emphasize both vertical understanding of HPC systems (parallel computer architecture, operating system/resource management system, compilation, library optimization) and the understanding of multiple HPC application areas.

The article by Bautista and Sukhija describes a new approach at National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory (LBNL) patterned after the apprenticeship program within the High Performance Computing domain. This approach requires an intern or apprentice to fulfill milestones during their internship or apprenticeship timeframe, with constant evaluation, feedback, mentorship, and hands-on work that allows candidates to demonstrate their growing skill that will eventually lead to winning a career position. The authors conclude by reporting the positive outcomes of the program such as recruitment of quality talent, improved retention, and encouragement to individuals to further their education.

The article by Ma et al. details the infrastructure and the outcomes of Ask.cyberinfrastructure.org, which is a collaborative, crowd-sourced Q&A site specifically curated for the research computing community. The authors discuss various technologies employed to build the site and the Locales, which allow institutions to create subcategories on Ask.CI where they can experiment with posting institution-specific content and use of the site as a component of their user support strategy. The authors report on lessons learned, plans to foster outreach efforts to reach out to other communities and mailing lists to expand Ask.CI's presence, and to invite any suggestions/recommendations from the community.

The article by Colbry presents a newly developed course at the Department of Computational Mathematics Science and Engineering (CMSE) at Michigan State University (MSU) for teaching parallel programming to undergraduates. The author describes the flipped classroom model and a "hands-on" approach used in the "Methods in parallel Programming" (CMSE 401) course for learning with multiple real-world examples from a wide range of science and engineering problems. The author concludes by discussing the feedback and challenges reported by the students and plans to improve the course.

The article by Peoples describes the learning outcomes that are focused on the transferable skills intended to be gained because of the assessment design. The author discusses assessments which were disseminated to a cohort of students on a Master of Science degree in Professional Software Development at Ulster University, United Kingdom. This Master's degree is a conversion degree into Information Technology for students from non-IT backgrounds. The author report that the creative assessment design helped to bridge these gaps by exposing students to state-of-the-art technology on an international basis, helping them to understand the software developments which are essential in their support at the back-end, and encouraging the application of knowledge in new way.

The article by Chen et al. presents the design of a parallel computing course offered at the College of Meteorologic Oceanography at the National University of Defense Technology in China. The authors discuss the design of the course, focusing on addressing the scalability challenges presented by non-computer science majors who lack a background in fundamental computer architecture, systems, and algorithms. The authors also present a set of assignments and projects that leverage the Tianhe-2A supercomputer for testing. The authors conclude by reporting in the result of the present pre- and post-questionnaires to explore the effectiveness of the class design.

The article by Hosseini and Lucas describes a quantitative methodology, POP, for the assessment of parallel codes at the Performance Optimisation and Productivity (POP) Centre of Excellence, funded by the EU under the Horizon 2020 Research and Innovation Programme. The authors detail the POP methodology as a scalable performance analysis methodology based on a set of hierarchical metrics, where each metric represents a common cause of inefficiency in parallel applications based on a set of hierarchical metrics, where the metrics at the bottom of the hierarchy represent common causes of poor performance. The authors conclude by illustrating the advantages of employing the POP methodology by describing two real-world examples that employ the POP methodology to help HPC users understand performance bottlenecks of their code.