Enhancing HPC Curriculum through Competitions

Cristina Carbunaru
National University Of Singapore
cristina@nus.edu.sg

Sriram Sami
National University Of Singapore
sriramsami@nus.edu.sg

ABSTRACT

High Performance Computing (HPC) supports breakthroughs in artificial intelligence (AI), data-intensive science, and engineering. At the National University of Singapore (NUS), core parallelism concepts are currently taught through courses in Parallel Computing and Concurrent Programming, with additional domain-specific exposure in courses. While these offerings build strong theoretical foundations, they leave a gap in systems-level competencies essential for deploying, optimizing, and scaling applications on real HPC infrastructure.

We addressed this gap by initiating several projects meant to increase the knowledge in system-level skills for HPC. A main initiative is the participation in HPC student cluster competitions through which we integrated training in resource management, profiling, monitoring, containerized workflows, and distributed AI workloads for our selected students. This focus enables participants to bridge programming theory with operational expertise, preparing them to work effectively with clusters and heterogeneous architectures. Building on the interest around HPC competitions, the main curriculum in computer science is developing to include full-fledged HPC courses.

We faced several challenges in this process, including a steep learning curve with complex systems, limited access to costly and shared cluster resources, and a shortage of instructors with up-to-date expertise. Pedagogically, bridging theory and large-scale practice is difficult, especially in the HPC context where the access to resources is remote. Therefore, sustainable curriculum development calls for a gradual expansion of teaching topics and resources, coupled with the integration of hands-on, competition-driven learning to maintain engagement.

Formal HPC training enhances students' readiness for careers in computational science, promotes cross-disciplinary collaboration, and equips graduates with the advanced skills essential for solving complex challenges in AI and data-intensive fields.

KEYWORDS

Student Competitions, Artificial Intelligence, HPC Education

1 INTRODUCTION

With the increasing interest in high-performance computing generated by the need to manage and harvest computation resources for large scientific tasks and AI, we embarked on a journey to improve the HPC curriculum at the National University of Singapore (NUS). Foundational parallelism concepts are taught through courses such as CS3210 Parallel Computing, CS3211 Parallel and Concurrent Programming, providing strong theoretical grounding and programming skills, but they do not fully address the systems-level competencies required for deploying, optimizing, and scaling workloads on real HPC infrastructure.

This paper presents the existing courses related to HPC and illustrates the efforts of enhancing HPC training and education at NUS. We adopted an incremental approach by integrating hands-on training through final year projects (FYPs) and initiating participation in international student cluster competitions. These activities expose students to advanced topics such as resource management, distributed AI workloads, performance profiling, monitoring, and containerized workflows, which are skills essential for managing heterogeneous HPC clusters. The growing interest in HPC has motivated the proposal for a dedicated HPC course [1].

2 EXISTING PARALLEL COMPUTING COURSES

CS3210, CS3211, and CS5239 together form a strong foundation for HPC education at NUS by covering parallel architectures, programming paradigms, and performance optimization – key pillars in large-scale computing [2]. CS3210 Parallel Computing focuses on the theory and practice of parallelism, exposing students to shared-memory, distributed-memory, GPU, and heterogeneous systems. It teaches parallel computation models, algorithm design, scalability analysis, and performance measurement – skills directly applicable to HPC cluster workloads. CS3211 Parallel and Concurrent Programming deepens this by exploring concurrency in modern languages such as C++20, Go, and Rust. It addresses synchronization, safety, robustness, and performance trade-offs, enabling students to write efficient, scalable, and safe parallel software for emerging technologies and HPC environments. CS5239 Computer System Performance Analysis complements both by teaching latency, utilization, bottleneck analysis, and queuing theory. Its focus on measurement, workload characterization, and tuning mirrors the profiling and optimization work essential in HPC system and application performance engineering. Together with other related courses, such as Distributed Systems and Cloud Computing, they provide the theoretical, programming, and performance analysis skills needed for advanced HPC roles.

3 INITIATING SYSTEM ADMINISTRATION (DEVOPS) TRAINING

Managing technical resources for courses such as CS3210 Parallel Computing, CS3211 Parallel and Concurrent Programming, and CS5239 Computer System Performance Analysis presents notable challenges in ensuring effective use of the Parallel and Distributed

https://doi.org/10.22369/issn.2153-4136/17/1/8

Computing Lab cluster by about 400 students per semester, given its limited 24-node capacity. These challenges present rich opportunities for Final Year Projects (FYPs), where students can design and implement enhancements such as highly available, customizable monitoring infrastructures and automated software installation systems. We started training students on deployment, setting up, and monitoring technologies by engaging them in these undergraduate FYPs. These projects saw high interest from our students, which proved the interest in systems-level competencies beyond what the current curriculum could offer.

We proposed and developed a platform for managing the nodes for students and teaching staff. We created a lightweight and robust system that is customizable to support different course requirements, but at the same time, is easily shared by multiple groups of students and teaching staff. Trainees explored architecture design, software stack selection, integration difficulties, and potential operational issues, while their deployment and testing in real teaching environments would directly improve the educational experience of students. Students were trained on using tools such as Ansible (for deployment), Slurm (for job management), LDAP (for user management), Prometheus and Grafana (for monitoring and alerting). Mastering these tools equips students to manage HPC clusters efficiently – automating deployments, scheduling jobs, controlling user access, and monitoring performance in real time are skills essential for reliable, scalable, and secure operation of large-scale scientific and AI computing environments.

4 PARTICIPATION IN STUDENT CLUSTER COMPETITIONS

Participation in student cluster competitions directly enriches HPC curricula by providing hands-on, real-world experience that extends beyond classroom learning [3]. Students must design, configure, and optimize an actual HPC cluster under strict time and power constraints, applying concepts from parallel programming, systems performance, and resource management taught in courses like CS3210, CS3211, and CS5239.

Building on the training experience we gained through FYPs, we initiated the first NUS participation in Student Cluster Competitions in 2024. Our first NUS team, “Kent Ridge”, achieved immediate and notable success – winning the Hero Run (HPLinpack) at IndySCC@SC24 in Atlanta and securing third place in the Virtual SCC@ISC25 in Hamburg. These results showcase the technical and collaborative capabilities fostered through competition training.

The overarching goal is to build a platform where students can deepen and showcase their expertise in HPC applications, hardware, and software while fostering collaboration, innovation, and networking within the global HPC community. So far, approximately 20 students have been trained on advanced HPC topics not currently covered in the standard NUS School of Computing’s curricula, including cluster architecture, job scheduling, performance tuning, and advanced monitoring – bridging the gap between academic theory and professional HPC practice.

5 PROPOSING A HPC COURSE

The increasing interest in High Performance Computing (HPC) among students, generated by participation in cluster competitions and advanced project work, has motivated us to propose a dedicated HPC course at NUS. This course would go beyond the current parallel computing and performance analysis modules, providing students with hands-on experience in building, optimizing, and running applications on large-scale computing systems [4]. The course would focus on the integration of HPC with AI, including the design and deployment of distributed training infrastructure using frameworks such as Colossal-AI, and the development of efficient pipelines for large-scale model training. Students would also explore domain-specific HPC workflows in areas such as climate modeling, bioinformatics, and precision medicine, gaining exposure to real-world, data-intensive scientific challenges. The course would teach profiling and optimization techniques for large scientific applications, enabling students to identify performance bottlenecks and improve scalability across heterogeneous architectures. Practical work would leverage national and institutional HPC infrastructure, including the National Supercomputing Centre (NSCC) of Singapore and NUS IT resources, providing authentic experience with production-scale systems.

6 CONCLUSION

Empirical data collected from discussions with students and faculty shows a strong demand for enhanced HPC education at NUS. There is growing interest among students eager to deepen their knowledge in HPC technologies, coupled with a cohort of highly capable individuals ready to engage with advanced topics. This feedback reinforces the need to expand the curriculum with dedicated HPC courses and practical training opportunities, ensuring that graduates are well-prepared to meet the evolving demands of computational science, AI, and interdisciplinary research.

REFERENCES

[1] Weronika Filinger, Julie Mullen, Jeremy Cohen, Samantha Wittke, and Ann Backhaus. 2024. Building HPC Learning Pathways: Understanding our Community. In *Practice and Experience in Advanced Research Computing 2024: Human Powered Computing (PEARC ’24)*. Association for Computing Machinery. https://doi.org/10.1145/3626203.3670513

[2] Michael Gowanlock. 2025. Teaching parallel and distributed computing using data-intensive computing modules. *Journal of Parallel and Distributed Computingournal of Parallel and Distributed Computing* 202 (2025), 105093. https://doi.org/10.1016/j.jpdc.2025.105093

[3] Bryan Johnston, Lara Timm, David Macleod, John Poole, Lily de Melo, Sayfullah Jumoorty, Jonathan Faller, Reinhard Jansen van Vuuren, and Michael Beukman. 2025. From Student SIG to Success: The journey of a student HPC Special Interest Group towards sustainable training and success in Student Cluster Competitions. *The Journal of Computational Science Education* 16 (03 2025), 43–49. https://doi.org/10.22369/issn.2153-4136/16/1/9

[4] Sushil K. Prasad, Almadena Chtchelkanova, Sajal Das, Frank Dehne, Mohamed Gouda, Anshul Gupta, Joseph Jaja, Krishna Kant, Anita La Salle, Richard LeBlanc, Manish Lumsdaine, David Padua, Manish Parashar, Viktor Prasanna, Yves Robert, Arnold Rosenberg, Sartaj Sahni, Behrooz Shirazi, Alan Sussman, Charles Weems, and Jie Wu. 2011. NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: core topics for undergraduates. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, 617–618. https://doi.org/10.1145/1953163.1953336