A Proposed Model for Teaching Advanced Parallel Computing and Related Topics

Katharine Cahill
Ohio Supercomputer Center
Columbus, U.S.A.
kcahill@osc.edu

Steven I. Gordon
Ohio Supercomputer Center
Columbus, U.S.A.
sgordon@osc.edu

Scott Lathrop
Shodor Education Foundation, Inc.
Durham, U.S.A.
lathrop@shodor.org

Abstract- The heterogeneous, multi-core and many-core architecture of modern computers offers opportunities to solve critical problems in science and engineering as well as parallel programming challenges to those who wish to use them. This problem includes computer science programs but extends to a wide variety of STEM programs where computation has become an integral part of the research infrastructure. Students from science and engineering majors often have limited programming experience. Instructors from these fields are often not knowledgeable enough about the techniques used for efficient parallel computing to create and run these courses independently. Even in places where there is faculty expertise to teach the courses, they tend to be low in enrollment and thus are less likely to be offered. This leaves students who wish to learn the advanced concepts of parallel and large scale computing to seek resources elsewhere or stumble on blindly in their field and pick things up as they go. Although some of the required content is available online as self-paced or MOOC style courses, those approaches are not effective in getting the majority of the target students the level of expertise they require. Both the Blue Waters and XSEDE projects have experimented with collaborative online courses that have proved to be an excellent model for resolving some of these problems. In this paper, we define the problems more fully and then describe our proposed collaborative model. We then conclude with a discussion about the challenges of implementing the model more permanently and at scale.

Keywords – HPC curriculum, flipped-classroom, MOOC, SPOC, team teaching, online learning, workforce development, diversity in STEM

I. Introduction[1]

The problems associated with providing adequate advanced computing curriculum at the university level vary from institution to institution, but in order to examine the efficacy of the existing programs, we have listed several common issues that need to be addressed.

A. Instructors do not have the expertise to create and run their own courses.

Today's heterogeneous HPC architectures pose a number of unique challenges to those seeking to optimize their codes to run efficiently to conserve the oversubscribed hardware resources. Most of the science and engineering codes are developed by researchers and their students who have limited computer science training and expertise. Thus, there are a limited number of institutions where the faculty have sufficient expertise to teach a course focused primarily on efficient parallel computing coding techniques. In addition, those faculty are primarily responsible for teaching courses in their subject area domains and often have little time to undertake such an offering.

At the same time, the graduate students undertaking research that requires the use of parallel computing techniques to bring their projects to completion. They are then left to train themselves in the relevant programming techniques. The

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. The XSEDE collaborative course is currently supported by a grant from the National Science Foundation award ACI-1650912.

results will then vary - from a few projects that successfully optimize their codes to many where the codes are suboptimal and inefficient. This both wastes valuable computational resources and hampers future research efforts.

B. If student interest in the course does not meet a certain threshold, the course is not offered.

Although computer modeling has become a key aspect of research in the sciences and engineering, on all but the largest campuses, there are only a handful of graduate students at any one time that need to acquire the necessary coding skills to carry out their research. For this reason, it is often difficult to offer the relevant courses in light of campus rules for the minimum number of students in a course or for such a course to be offered in lieu of other courses with higher enrollment demands. Even on large campuses where the faculty have the requisite expertise to teach a course, this often means that the courses will not be taught or will be taught very intermittently, often leaving students on their own to learn the required material.

C. Difficulty of offering these topics falls most heavily on Minority Serving Institutions (MSIs), reinforcing lack of diversity in HPC workforce.

Supercomputers were introduced in the 1960s. Ever since that time, high performance computing (HPC) which for the purposes of this paper will also include high throughput computing (HTC), has been critical to science and engineering research. Today, applications of HPC have mushroomed to support discovery across all fields of study. The National Science Foundation established the supercomputing centers program in 1986, funding as many as five supercomputing centers at one time. Today there are well over 15,000 people utilizing the NSF funded HPC systems across the country, and tens of thousands more using campus computing facilities.

And yet, for all of the demand for HPC resources, there are few formal education programs preparing today's workforce to make effective use of these rapidly evolving systems. Today, there needs for courses to prepare students across the spectrum of science, technology, engineering, mathematics, humanities and social sciences in at least three major areas:

- Code developers working to maximize the potential of large-scale HPC systems who need to learn about software engineering, MPI, OpenMP, OpenACC, code profiling and optimization, large data and I/O handling, software libraries, workflows, visualization, verification and validation, ethics, security
- Applications researchers utilizing community codes to accelerate the "*time to science*" who need to learn about software engineering, applications selection, scientific visualization, workflows, meta-data, large data and I/O handling, verification and validation, ethics, security
- IT professionals designing, building and supporting HPC centers of excellence who need to learn about benchmarking techniques, building design including power and cooling, networking, security, operations tools, data storage and archives, system fault recovery, consulting models, training methods, budgeting, setting policies and standards

The Society for Industrial and Applied Mathematics (SIAM) website [1] identifies that there are 41 graduate programs and 7 undergraduate Computational Science programs across the US, although this may be an undercount due to the creation of new programs. There are 23 colleges/universities with master's degrees in data science [2] and a growing number offering such degrees.

As a result, there are insufficient HPC related course offerings across the nation, especially among non-PhD granting institutions, to prepare an adequate workforce to address the workforce gap in the country. The report issued by the Department of Energy, "HPC Workforce Challenges at Labs and University Centers" in April of 2015 [3], described the critical need for a skilled workforce in computing sciences. The lack of a workforce graduating from two and four-year academic institutions, EPSCoR institutions, and Minority Serving Institutions, will exacerbate the need to prepare a large and diverse workforce able to contribute to modern day science and engineering environments in academia, business, industry, and government.

D. MOOCs and other online self-driven courses do not offer credit, have high drop-out rates, and lack the instructional support needed by students from varied backgrounds.

Massive Open Online Courses - (MOOCs) are becoming very prevalent and popular. As described on the Coursera website, participants may "*Earn official recognition for your work, and share your success with friends, colleagues, and employers.*" For example, Johns Hopkins offers a non-credit *Data Science* series [4]. There are numerous institutions offering accreditation at the institution delivering the MOOC content, such as the University of New Mexico. Other institutions, like the University of Illinois, welcome exploring partnerships and collaborative agreements with corporations, school districts and group cohorts to develop credit or non-credit educational programs. As reported on the MOOCs for Credit web site [5], "*Seven MOOCs offer transferable college credit, and there are more planned. You enroll in any of the seven courses on edX, and add the Verified Certificate option (which costs $49). If you pass, you pay Arizona State University a fee of $600 per course and they give you the credits (which you can transfer to any university).*"

There are no MOOCs, to our knowledge, that allow students to enroll and directly earn formal course credit at their home institution for a course taught at a different institution.

More importantly, MOOCs have been criticized for the high drop-out rates related to a combination of lack of instructional support for the students and lack of incentive to complete the course. [6-10]

II. Implementing a Blended Online Collaborative Course Model

To address the challenge of providing students with credit at their own institution, the Blue Waters team launched the Virtual School for Computational Science and Engineering (VSCSE). The VSCSE program was modeled on the Big10 Academic Alliance CourseShare program [11] whereby faculty from one institution teach an online course which students from other member institutions may enroll. The Big10 member institutions have developed agreements to allow for the transfer of course credits among the member institutions. The Big10 program provides mechanisms for balancing the institutional financial costs of students enrolling in courses at other member institutions, as long as each institution offers a similar number of courses with overall similar levels of enrollment. However, this model is not easily adaptable to interested academic institutions across the country, and has not experienced sufficient interest to be readily extensible in all science and engineering departments. The VSCSE program has to date provided 633 graduate students at 28 US institutions in 16 states, as well as institutions in Canada, Cyprus, Puerto Rico, and Saudi Arabia with the knowledge and skills to advance computation and data-enabled discovery. Among the collaborating institutions were 7 institutions in EPSCoR states/jurisdictions, and 2 MSIs. A secondary outcome was providing the collaborating faculty at 32 institutions with the knowledge, materials, resources and confidence to be able to teach the course content on their own. Subsequent to the courses being taught, we have made the materials available to any faculty member that requests access. The VSCSE offered seven courses as described in Table 1.

Each of these courses focused on a set of HPC software- and hardware-related skills. Recorded lectures introduce the course materials. Then a mixture of quizzes and assignments help students apply those skills. Descriptions of some of the courses offered below provides an overview of the technical content of each.

TABLE 1. VSCSE COURSES OFFERED TO DATE.

Year	Faculty Member	Topic	Institu-tions (MSI)	Students
Spring 2013	Dr. Wen-mei Hwu	*Algorithmic Techniques for Scalable Many-core Computing*	2	54
Fall 2014	Dr. Wen-mei Hwu	*Algorithmic Techniques for Scalable Many-core Computing*	4	74
Spring 2015	Dr. Bill Gropp	*Designing and Building Applications for Extreme Scale Systems*	3	49
Spring 2015	Dr. Han-Wei Shen	*High Performance Visualization for Large-Scale Scientific Data Analytics*	7 (1)	131
Spring 2016	Dr. Bill Gropp	*Designing and Building Applications for Extreme Scale Systems*	11 (2)	130
Fall 2016	Dr. Wen-mei Hwu	*Algorithmic Techniques for Scalable Many-core Computing*	5	56
Fall 2016	Dr. David Keyes	*Introduction to HPC*	10 (2)	139

Algorithmic Techniques for Scalable Many-core Computing

This course was designed to allow students to master commonly used algorithm techniques and computational thinking skills for scalable, many-core/many-thread programming. It included coursework in the following areas: many-core hardware limitations and constraints; desirable and undesirable computation patterns; practical algorithm techniques to convert undesirable computation patterns into desirable ones. Regular quizzes and labs were included as well as a final group project implemented on Blue Waters. Familiarity with CUDA programming was a prerequisite.

Designing and Building Applications for Extreme Scale Systems

In this course, students learned how to design and implement applications for extreme scale systems, including analyzing and understanding the performance of applications, the primary causes of poor performance and scalability, and how both the choice of algorithm and programming system impact achievable performance. Dr. Gropp covered multi-and many-core processors, interconnects in HPC systems, and parallel I/O. Students who take this course are expected to have a strong knowledge of C, C++, or Fortran, including writing, debugging, and optimizing an application as well as some parallel programming experience. Weekly homework and final group projects were completed using Blue Waters.

High Performance Visualization for Large-Scale Scientific Data Analytics

To equip students with the ability to analyze very large-scale data sets, this course provided an in-depth discussion of the state-of-the-art in large scale scientific visualization algorithms and systems. In addition to the fundamental visualization techniques, parallel implementation of selected algorithms for high-performance architectures was covered. Hands-on visualization projects included work on Blue Waters.

Introduction to HPC

This course focused on high performance computing algorithms and software technology, with an emphasis on using distributed memory systems for scientific computing. The main topics covered in this course include computer architecture, state-of-the-art discretization techniques, solver libraries, and execution frameworks. The programming assignments use MPI and PETSc on local resources and Blue Waters culminating in an independent project leading to an in-class report.

The XSEDE project launched a similar distributed model of online teaching in collaboration with Dr. James Demmel, University of California, Berkeley. The course, Applications of Parallel Computers, is intended for graduate students from a variety of disciplines to learn about the algorithms and patterns associated with creating efficient parallel computing programs on modern computing architectures. Over the four years of offering this course as a collaborative online course, 573 students from collaborating institutions have completed the course. As shown in Table 2, the course has impacted a large number of institutions including a number of minority serving institutions that otherwise might not have been able to offer the course. The participating institutions were from 19 different states, Puerto Rico, the Virgin Islands, and six other foreign institutions.

The VSCSE and XSEDE models selected instructors (faculty with the appropriate computing expertise) to provide online courses generally not available at academic institutions through a blended approach of online delivery of content and the active participation of collaborating faculty at remote institutions. The lead instructor also taught the course on their own campus. While teaching on their own campus is not a requirement for the model to work, it does afford the faculty member with a local audience to better gauge the experiences of the students enrolled in the course.

The VSCSE and XSEDE projects announced the course offerings, and solicited collaborating faculty who wished to offer the course on their own campus. The collaborating faculty were not expected to teach any sessions, but they were expected to convene the course on their own campus, ensure that the students viewed the lectures, facilitated local discussions with their students, and awarded final grades to their own students. The collaborating faculty generally established a special seminar graduate course at their institution. Students were able to enroll in the course offered at their own institution.

TABLE 2. APPLICATIONS OF PARALLEL COMPUTERS PARTICIPANTS

Year	Institutions	Minority Serving	Students completing
2014	18	4	143
2015	10	2	108
2016	11	2	215
2017	13	2	107

Each of the campuses with enrolled students followed different academic schedules and there were multiple time zones to address. As a result, the courses could not be delivered in real time. Therefore, the instructors recorded the lecture material for viewing at a later time by the students. These teaching models used the open source learning platform, Moodle [12], to host content and provide forums for discussion.

The collaborating faculty had the option to have their students view the lectures during their regularly scheduled class time, with the faculty member able to help answer questions of their own students during this class time. However, many faculty used a flipped classroom model, whereby the students viewed the lectures on their own, and the faculty used the scheduled classroom time for open discussions on the video content. The collaborating faculty and students found the flipped classroom model to be a richer learning experience, and improved the experience for both the faculty members and students.

III. CREATING A CONSORTIUM

There are numerous models for teaching online courses that are actively being pursued including Small Private Online Course (SPOC) and Massive Open Online Course (MOOC) styles of teaching. The VSCSE and XSEDE chose to use the SPOC model to ensure that the students gain in-depth understanding of the material with direct support and mentoring from their local faculty. The SPOC model has demonstrated much higher retention rates among students compared with the MOOC approach.

We suggest a consortial approach using this SPOC model: A series of courses that cover the fundamental skills of HPC used for research. This would bring the positive experiences of the Blue Waters and XSEDE approaches to more institutions and provide greater structure for both the faculty and students. Several examples of this model have been recently reported with a similar type of consortial program [13, 14], they describe using a MOOC approach, but the activity described is more like a SPOC model in its focus on providing materials to a small number of institutions at a time.

Clearly the collaborative efforts by both projects have been well-received and by the participating institutions. However, there remain some challenges to creating a more permanent, on-going collaboration using this model. Discussions with the participating faculty has indicated several problems that need to be resolved:

1. Planning for participation in such a consortium requires a timeline of two to three years. Faculty have trouble inserting courses even one or two semesters in advance given the constraints on course scheduling and workload.
2. The courses have not reflected a true sequence but instead have been created based on the opportunities to find willing instructors. Any future consortial approach should more carefully design a sequence with more complete prerequisites and learning goals.
3. None of the participating faculty aside from the lead instructors are willing to take the responsibility to create an entire online course in the future. Most are willing to participate more actively but would prefer to be responsible for only a portion (such as one month) of a course.
4. The institutional arrangements for the collaborative courses has been ad hoc. Individual faculty have received permission at the department level to offer the courses as they arose. In some cases, faculty took on the course as an overload without compensation. This is not sustainable in the long run. Any future consortium will require a more definitive approval at the institutional level.

These issues need to be addressed in the potential formation of one or more ongoing consortia of institutions participating in their implementation. We expect the solutions would be negotiated by the members of consortium and would vary with the needs of the specific institutions. This will involve the collaborative design of a two- to three-year sequence of relevant courses. With that design, the participating faculty will then need to commit to lead a portion of the instruction as well as to have their students participate in the course sequence.

With such a curriculum design in place, it should then be possible to obtain formal institutional commitment to the consortium including the integration of the courses into the regular curriculum at each participating institution. Of course, such an approval process at the diverse set of participating institutions will not always go smoothly. Nevertheless, this collaborative model may be one of the best ways to address the gap in the workforce by providing the appropriate skills in parallel and high performance computing to a broader, more diverse population.

IV. Conclusion

Making HPC curriculum available in more institutions is a critical need in order to train the future workforce in these necessary skills. Creating new avenues to bring these courses to under-served communities is an important step to ensuring that more, and more diverse, students have access to these skills.

References

[1] Society for Industrial and Applied Mathematics, Graduate Programs in Computational Science. https://www.siam.org/students/resources/cse_programs.php
[2] 23 Great Schools with Master's Programs in Data Science. http://www.mastersindatascience.org/schools/23-great-schools-with-masters-programs-in-data-science/
[3] Chapman, Barbara. 2015. HPC Workforce Challenges at Labs and University Centers https://studylib.net/doc/9244417/hpc-workforce-challenges-at-doe-labs-and-university-centers
[4] Coursera: Data Science Specialization https://www.coursera.org/specializations/jhu-data-science
[5] Lequerica, Anuar. 2016. MOOCs for Credit: All the Different Ways You Can Get Credit from MOOCs https://www.class-central.com/report/moocs-for-credit/
[6] Dacey, James. 2015. The MOOC Point. Physics World Archive, March 2014. IOP Publishing Ltd 2015 ISSN: 0953-8585.
[7] Daniel, Sir John. 2012. Making Sense of MOOCs: Musings in a Maze of Myth, Paradox and Possibility. Journal of Interactive Media in Education, JIME http://jime.open.ac.uk/2012/18.
[8] DeJong Peterson, Rachelle. 2014. MOOC Fizzles. Acad. Quest. 27:316–319, DOI 10.1007/s12129-014-9434-3.

[9] Biemiller, Lawrence. 2013. MOOC Busters. Chronicle of Higher Education. Vol. 60 Issue 15, pA23-A23, 2013.

[10] Gordon, Demmel, Destefano, Rivera, 2015. "Extending Access to HPC Skills Through a Blended Online Course", XSEDE15.

[11] Big10 Academic Alliance CourseShare program https://www.btaa.org/projects/shared-courses/courseshare/introduction

[12] Moodle Pty Ltd. Perth, Australia. Moodle - Open-source learning platform | Moodle.org

[13] Liu, Ludu, Klein, Spector, Ikle, "Innovative Model, Tools, and Learning Environments to Promote Active Learning for Undergraduates in Computational Science & Engineering" JOCSE, Accepted, 2017.

[14] Mullen, Byun, Gadepally, Samsi, Reuther, Kepner. 2017 "Learning by Doing, High Performance Computing education in the MOOC era", J. Parallel Distrib. Comput. 105, pp105-115.