Developing a Scientific Computing Cluster Course for the Undergraduate Curriculum

Stephen Lien Harrell, Benjamin J. Cotton, Michael E. Baldwin, Andrew L. Howard

Purdue University

Abstract

The demand for scientific computing is ever-increasing, making those who can use such resource very valuable. Unfortunately, undergraduate curricula too often focus on science or computing without addressing the intersection of the two fields. This paper describes efforts at Purdue University to create scientists who can effectively use scientific computing resources and system administrators who can provide resources to meet the needs of science. In this paper, we describe early offerings of a scientific computing course. After several offerings, certain areas for improvement were evident. We describe the changes to address the early shortcomings and offer a glimpse of the future work to expand efforts beyond a single course.

Introduction

Incorporating scientific computing into the undergraduate curriculum is not a new concept. As early as 1994, Dartmouth College was planning a freshman-level course in parallel computing [1]. In the nearly two decades since, the literature has not presented a comprehensive best practices guide. This may be due in part to the difficulty of developing course content that is simultaneously approachable for both computer science and domain science students. Students in domain sciences generally require background education in computer science topics before being able to understand the application of scientific computing to their field. Similarly, computer science students do not understand the broader context of the scientific domain and so require introduction to the applied domain. [2]

PURDUE UNIVERSITY

Motivation

Some efforts in course development have chosen to focus on the computer science students (e.g. [3] and [4]). The focus of this course is on bridging the gap between the domain science and system administration in order to create a new generation of system administrators and domain scientists who can effectively communicate with each other. The goal is to provide a big-picture view of scientific computing to all participants.

The impetus to teach scientific computing and its administration came from a competition that is held each year at the Supercomputing Conference(SC). Purdue University has brought a team to every Student Cluster Competition at SC since its inception. This activity has been used to train students to repair and understand large scale compute resources as well as train them for their careers after graduation. The competition was the impetus to create a class which was run for two years [5]. When teaching the class it seemed that two disparate subjects were being taught; scientific computing and large installation system administration. During the classes there were few attempts to connect the two, the courses lacked coherence. It was during that time it became clear that a new curriculum was needed that was not tied to the competition directly.

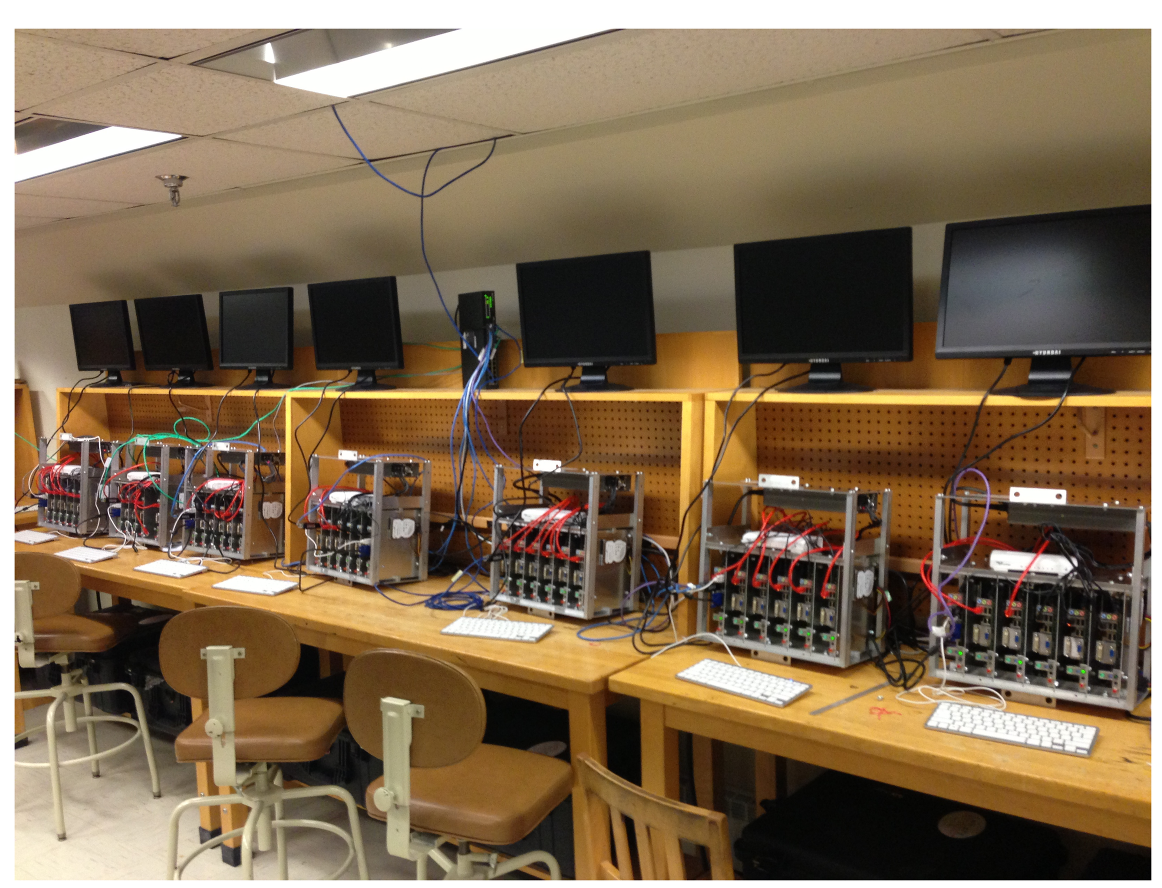

Figure 1: Lab space for LittleFE instructional clusters

Initial Course Offering

The overarching idea of the class was going from a box of unassembled commodity hardware to running a forecast model and pulling out basic data from the output, such as what the daily maximum temperature. In this way it was possible to introduce students to both the system administration of and consumption of compute resources for scientific computing. Each student started this course with four desktop computers connected with gigabit Ethernet for themselves. The auto installer infrastructure was setup for the students and was lectured on briefly. Once the machines were installed, the students installed basic packages such as torque and MPI libraries. The students then compiled the Weather Research and Forecasting (WRF) model with MPI. Once WRF was compiled and running for all students, basic workflow management and data processing using Python was taught.

Revised Class

The initial class was well-received, but had room for improvement. During the second semester the class content was tweaked to focus on making those improvements.

We redefined the outcomes for the current class and they fall into two basic categories. The topics in the first group of outcomes are running and maintaining a small computational science resource. For the second group the topics encompass usage of scientific applications on a parallel computational resource.

This first set of outcomes fill an oft-overlooked niche in many parallel and distributed computing curricula. The ability to stand up large scale compute resources, specifically for scientific computing, is a need at Purdue and elsewhere. In addition for the science student who will go on to schedule or program these resources these outcomes will also be useful as a view of the internals of the systems are exposed to them.

SESA '13

Conclusion

Based on our experiences and the research of others, we recommend introducing students to scientific computing concepts early in their academic careers. The scientific models should be used as tools to teach the science instead of teaching science to understand the models. Students often do not realize what resources are available, and they expect getting started to be more difficult than it is. Courses like ours serve to help expose students to the scientific computing resources and help build confidence that those resources are within reach.

References

[1] D. Johnson, D. Kotz, F. Makedon *et al.*, "Teaching parallel computing to freshmen," in *Conference on Parallel Computing for Undergraduates. Colgate University*, 1994.

[2] D. Burhans and G. Skuse, "The role of computer science in undergraduate bioinformatics education," in *ACM SIGCSE Bulletin*, vol. 36, no. 1. ACM, 2004, pp. 417–421.

[3] M. Chen and T. Li, "Construction of a high-performance computing cluster: A curriculum for engineering and science students," *Computer Applications in Engineering Education*, vol. 19, no. 4, pp. 678–684, 2011. [Online]. Available: http://dx.doi.org/10.1002/cae.20352

[4] C. Nevison, "Parallel computing in the undergraduate curriculum," *Computer*, vol. 28, no. 12, pp. 51–56, 1995.

[5] S. L. Harrell, P. M. Smith, D. Smith, T. Hoefler, A. A. Labutina, and T. Overmyer, "Methods of creating student cluster competition teams," in *Proceedings of the 2011 TeraGrid Conference: Extreme Digital Discovery*. ACM, 2011, p. 50.

Contact Information

Email: sharrell@purdue.edu