AC 2010-803: HANDS-ON HIGH PERFORMANCE COMPUTING: DEVELOPING A CLUSTER COMPUTING COURSE FOR REAL WORLD SUPERCOMPUTING

Thomas Hacker, Purdue University, West Lafayette

Developing a Course for Hands-on High-Performance Computing

Abstract

High-performance computing (HPC) based on commodity hardware and open-source software has become the dominant paradigm for supercomputing today.[1, 2] Thus a great unmet need exists for skilled students and practitioners who can design, develop, deploy, and operate HPC-based systems to support discoveries in industry and academe.

To address these needs, we have developed two courses in HPC, one for undergraduates and one for graduate students, that provides students with hands-on experience in designing, developing, and testing commodity-based supercomputing systems. In this paper, we describe a cost-effective and scalable approach that we developed for this course, which has been successfully delivered over several semesters. We describe the curricular context, pedagogical approach, and outcomes along with a detailed description of the approaches and strategies we used to develop a hands-on laboratory component that can be replicated by others seeking to develop similar courses. We believe that our results will be useful to departments and institutions interested in developing curricula to answer the increasing needs presented by HPC and cyberinfrastructure.

Introduction and Motivation

In the race for global competitiveness in technology, manufacturing, science, and engineering, computer-based design and simulation have become critical elements in producing higher quality and less-expensive products. Computer simulation and data analysis are central to this effort. Computer simulation is used by automobile manufacturers to design better products in shorter times at lower costs, to discover new and previously overlooked sources of oil and gas, and to improve industrial processes.[3] Data analysis, which involves sifting through terabytes of data to discover trends and unexpected patterns, is another emerging area of computation that is helping to improve product design, science, and engineering research.

Algorithms that form the "computational engine" and heart of these analysis and simulation efforts have grown in capability and complexity as the physical and temporal scales and resolutions of the problems posed by researchers increase. The increasing computational scale and complexity have created an increased demand for greater computational capacity and storage resources. This trend in computational demand has naturally led to a significant increase in the use of HPC and supercomputing to provide an effective computational platform on which to solve these problems.[4-9] In the past, supercomputing systems were designed and built exclusively from custom-engineered components, and thus they were rare and expensive. The powerful trends of increasing computational power and decreasing costs of commodity computer components driven by the home computing and gaming markets have completely changed the supercomputing industry over the past 15 years. As noted in a recent National Research Council Report[10]: "…over the last several years such commodity supercomputers have rapidly come to dominate the supercomputer marketplace." The dominance of commodity-based cluster computing systems is evidenced by the growing fraction of cluster-based systems on the Top 500 list – a semiannual list of the 500 fastest supercomputers.[1] In November 2000, a few years after the introduction of commodity-based Beowulf clusters,[11] cluster-based architectures comprised only 5.6% of the Top 500 list.[1] Since then, as of November 2009 the percentage of cluster-based architectures has grown to more than 83%.[1] Not only have the Top 500 computers, which represent the peak of the high-performance computing pyramid, become dominated by cluster architectures, but the quantity and diffusion of high-performance computing systems have also significantly increased, driven in large part by the decreasing costs and increasing performance of commodity-based systems. In today's high-performance computing marketplace, which includes lab-sized clusters as well as the Top 500 supercomputers, platforms built on the Linux operating system and commodity off-the-shelf technology (COTS) are the dominant platform.[12] This increased demand for HPC systems has generated also an increasing demand for skilled practitioners with the required knowledge and experience to build and utilize HPC systems for problem solving.

The design, construction, and operation of high-performance computing and supercomputing systems have traditionally been "on the edge" of the field of computer science and information technology; thus limited training and education resources have been available for students seeking to learn new skills. The result is a gap that has emerged between supply and demand, one in which the lack of skilled practitioners and available training in the development and use of HPC systems have become serious impediments in the adoption of HPC and supercomputing technology.

The specific needs for training in this area fall into several broad areas: architecture, applications, system sizing and cost minimization, performance, open-source supercomputing, and technology trends and forces. In the area of architecture, students need training in supercomputer and cluster-computing technology components and in the interrelationships among these components, as well as the cost/benefit tradeoffs of specific technologies and components used. In the area of applications, education in understanding the types of applications in use today (e.g., data intensive, computational fluid dynamics, embarrassingly parallel), cyberinfrastructure, and the specific architectural needs of each type of application are needed. On the subject of system sizing and cost minimization, the essentials are in the areas of assessing specific application needs and requirements to determine the optimal system size and configuration, and the best way to distribute funds among a range of technology options. For performance, students need a clear understanding of the inherent performance capabilities and constraints of architectures and technological components. One major subject is the use and development of open-source operating systems, libraries, and applications designed for use in developing and operating a high-performance computing infrastructure and environment. Another area is the impetus of market and commodity trends and forces shaping the technological development of commodity-computing components (such as CPUs, memory, and disk drives), and the effects of these trends on current and future supercomputing architectures and environments. Lastly, pulling these elements together, applying a working knowledge of how to put this training into use in developing and operating cost-effective and application-effective high-performance computing systems, is a critical need.

In this paper, I describe our approach to developing an undergraduate course and one for graduates in HPC systems that seeks to provide the necessary training and education to meet the training needs described above. The specific problem we seek to address is how to increase the population of skilled practitioners who have the requisite training, knowledge, and skills to design, develop, deploy, and operate HPC systems to support discoveries and analyses in industry and academe.

Approach

Our group approach to address this problem is to develop a curriculum that leverages existing courses and skills within the department. This effort, described in detail in Hacker[13], seeks to provide training in systems and applications. The goals of the curriculum are to increase student awareness and use of cyberinfrastructure technologies for problem solving. Within this context, I developed new courses in high-performance computing systems and parallel data systems that focus on the aspects of designing, building, and testing HPC systems. This paper focuses on the undergraduate course and graduate course.

Course Objectives

The long-term objectives of these courses are to increase the number of skilled practitioners with *demonstrated abilities* to design, build, and operate a high-performance computing system to support scientific applications. The short-term objective of the courses is to train students in the design, deployment, and use of commodity-based HPC systems. These courses have several specific short-term educational objectives (summarized from previous work[13]) that seek to ultimately meet the long-term objectives:

(1) Problem Assessment – the students should be able to assess a computational problem in the domains of science and engineering and determine if the application of high-performance computing could make a net positive impact on solving the problem. Students should be able to determine whether the problem has a feasible cost/benefit solution.

(2) Technology Selection – the students should be able to identify and survey a set of potential technology solutions and deliver an integrated high-performance technology solution that fits the constraints of specific application requirements, facility constraints, and budget at the lowest reasonable cost.

(3) Design, Implementation, and Operation – the students should be able to design a high-performance computing system from a collection of commodity off-the-shelf technology components and vendor subsystems that can meet specific performance, reliability, budget, and facility constraints and goals. They should also be able to deploy and maintain the system for a community of users over the useful lifetime of the system.

(4) Application – the students should have the skills needed to collaborate with a domain specialist in science or engineering to assess a computational problem within that discipline. Further, They should be able to apply HPC technologies to create a computational solution that is efficient, that can easily scale to a large system, and that uses equipment on hand or that can be purchased at a reasonable cost.

Pedagogical Approach

Providing one form of education through lecture and reading alone is known to be an effective approach for only a subset of learners. In Kolb's theory of experiential learning,[11] the most effective learning environment provides learning in several modes: abstraction conceptualization, concrete experience, reflective observation, and active experimentation. The pedagogical approach to the courses seeks to provide a range of learning experiences in each of these areas through a combination of lectures, discussions, laboratory assignments, and challenges requiring students to solve at least one difficult problem in the laboratory during the semester. Students enrolled in the courses are expected to have prior experience installing and using Linux and also some with building or upgrading desktop computers. Because of the prevalence of home computers and community computing (such as “LAN parties” and distributed gaming), the vast majority of students come prepared with knowledge they have gained on their own. Furthermore, many of the students have previously completed a rigorous course sequence in network and systems administration. The pedagogical approach for the courses is based on connecting course content with prior knowledge the students have from both personal experience and classroom education. By building forward from this base of knowledge in a progressive fashion, the students are aided in efforts to integrate new knowledge with existing knowledge and experience. They are expected to build on prior knowledge to solve a sequential series of problems involved in designing, building, and operating a commodity-component-based high-performance computing cluster utilizing resources in the lab that mirror the types of resources available to them in the real world. The courses involve at least one project in which students are presented with a problem within the context of a competition to achieve the best possible performance. This approach includes components of Problem Based Learning (PBL), which challenge students to apply their knowledge and skills as individuals and as a community to solve a difficult problem.

To help students achieve the educational objectives of the course, its activities are focused in several areas. First, a traditional lecture is provided, using a combination of slides, in-class discussion, and resources available on the Internet. Lectures cover history, real-world HPC applications, component technologies, trends and forces in commodity computer hardware, HPC application types and the specific needs of each type, reliability, data center design, commodity cluster design process, and how to design, build, and benchmark a high-performance computing cluster. Material is presented on the history of supercomputing to provide historical context for current technologies and trends and to demonstrate the manner in which technologies can change while the underlying principles remain constant. An Internet resource that has proven to be effective in communicating the history of technology is an extensive archive of historic computer vendor demonstrations and short clips from the Computer History Museum that shows early generations of computing hardware at work. One excellent example is a YouTube early IBM marketing video showing the first generation RAMAC drive at work and a mercury-based delay-line memory. These historical contexts help to initiate classroom discussions on how technology has advanced, but principles and approaches (such as the use of circular disk platters) remain unchanged. The classrooms in which lectures and discussions take place are located within the High-Performance Computing and Cyberinfrastructure Research Laboratory, which includes a small high-density data center. As part of a lecture, the instructor can take the students across the hall to provide a hands-on demonstration of technologies and infrastructure. This approach helps students immediately connect information shared in the lecture with real-world implementations of the technology.

In the graduate High-Performance Computing Systems course, students must design and construct a heterogeneous HPC system by using a mixture of commodity desktop systems, blades, and rack-mounted servers over a series of gigabit Ethernet switches that link the components together. Besides lectures, the course covers several seminal papers in the field, and each paper is discussed in class with the students. The papers cover storage technologies and parallel applications.[9, 14-20] Beyond similar objectives from the undergraduate course, there are several additional objectives for the graduate course. The first one is to teach students how to conceptualize and conduct a research project, which is a critical skill needed for discovery activities. The second is to train students to clearly communicate research results within a framework of hypothesis-based science. To meet these objectives, students propose and conduct a semester-long research project with the aim of producing a publication-quality research paper; they present results of the research to their peers at the end of the semester.

The commodity desktop systems used in the course are recycled Dell Optiplex desktop computers that are about four years old. The original use of the systems was in university computer labs. The second use was to provide a university-wide research cluster. Once these systems were retired from the cluster, we acquired them for use in the courses. And this third use has proven to be ideal for several reasons. First, the older equipment has a better level of support in Linux, since the open source community has had sufficient time from the initial release of the hardware to develop reliable drivers and patches. Second, because the hardware was older, students could open the cases and tinker with it without fear of damaging the equipment, since replacement parts are ubiquitous and plentiful for a system of that age. In practice, we have never had an instance of damage – the students' prior experience with their own computers provided sufficient background training. Third, the age of the equipment causes occasional failures (especially in memory and disk drives). As part of the problem-based learning, students learned to test, diagnose, and repair failing components in their systems during the semester. The learning that students gathered from this experience was excellent training for the conditions they may encounter in industry, since many early efforts by organizations to develop HPC capabilities are usually based on recycled computer equipment.[21, 22]

To support the courses developed as part of the HPC and cyberinfrastucture curriculum area, the Purdue College of Technoogy and the Department of Computer & Information Technology developed a comprehensive facility dedicated to research and education in HPC and Cyberinfrastructure. The Purdue High Performance Computing and Cyberinfrastructure Research Laboratory (HPC-CRL), described in more detail in Hacker[23], is a comprehensive facility dedicated to research and education in HPC and Cyberinfrastructure. The lab opened in August 2008, and is operated in partnership with IBM, Force-10 Networks, and Hewlett-Packard. The HPC-CRL is designed to support research, collaboration, and sharing in three main activity areas: the Data Center Research Laboratory, the Collaboration and Research Project area, and the Sharing area. The Data Center Research Laboratory is designed to house dense high performance computing equipment, and provide state-of-the art computing, power, and cooling systems to support research projects and student coursework. The second area is the collaboration area, which is a flexible reconfigurable space designed to support the needs of students, researchers, vendor partners, and faculty. The third area is the sharing space, which is a small space that can be used for meetings, courses, and seminars.

The assignments for the courses are cumulative. In the first assignment, students design and specify the architecture of the high-performance computing system they will be building, developing estimates for power, space, and cooling consumption. They are required to describe in detail the individual components they will be using, the networking and storage infrastructure, and the physical layout of the system. In the second assignment, students construct and wire their cluster; they also load the head node of the cluster with the open source CentOS operating system.[24] CentOS is an open source Linux distribution that seeks to closely emulate the functionalities and features of a commercially marketed and supported version of Linux (Red Hat Enterprise Linux). Although other Linux distributions (such as Debian) could be used, one advantage of CentOS is that it will fully accept Red Hat binary RPM (Red Hat Package Manager) installation packages, and it is fully supported by the OSCAR environment and works with OSCAR packages.[25] For the third assignment, students utilize the OSCAR cluster tool kit[26, 27] to create a computational node image, discover and load the operating system on the computational nodes, and test the basic functionality of the cluster. This assignment is the most challenging for the students because they need to integrate a variety of individual components into a properly functioning whole, and then pass an assessment of correct functioning that is provided by OSCAR. By this time, many of the students have developed relationships with other students in the course, and they often collaborate to share ideas and devise solutions to the problems they encounter as they build and load their high-performance computing cluster. The final project is based on using High-Performance Linpack (HPL[28]), a standard supercomputing benchmark program, to assess the performance of their cluster and to achieve a reasonable percentage of theoretical peak performance of the system. For this assignment, students work on identifying and eliminating bottlenecks in their system that impede performance; they also configure and tune the benchmark applications. For part of this assignment, if time permits, students can work in teams to observe the performance effects of scaling up their clusters with a larger number of processors.

Outcomes and Lessons Learned

The objectives of these courses are focused on increasing the number of skilled practitioners with a *demonstrated ability* to design, build, and operate a high-performance computing system to support scientific applications. The short-term objective is to train students in the design, deployment, and use of commodity-based high-performance computing systems. In terms of lessons learned, one observation is that the prevalence of high-quality open-source software and commodity computers makes it possible today to develop an engaging hands-on course in high-performance computing. The approach described in this paper was based on the maximal reuse of low-cost recycled and redirected computing equipment that can often easily be obtained from the property disposition stream of an organization. This approach provided a high educational impact from a pedagogical and learning perspective at a reasonably low cost. Students completing the course have developed a demonstrated ability to construct a cost-effective HPC system from open-source software and commodity components. Given the wide availability of the high-speed Internet and the ubiquity of computer components, students completing the course gain an easily transportable skill that can provide them with a significant occupational advantage and a competitive edge for employers seeking to adopt high-performance computing technology.

REFERENCES

1. *Top500 Supercomputer Sites*. [cited September 2009]; Available from: http://www.top500.org.
2. Feitelson, D.G., *The supercomputer industry in light of the Top500 data.* Computing in Science & Engineering [see also IEEE Computational Science and Engineering], 2005. **7**(1): p. 42-47.
3. Council on Competitiveness. *Full Vehicle Design Optimization for Global Market Dominance,* May 2005; Available from: http://www.compete.org/.
4. Council on Competitiveness. *Keeping the Lifeblood Flowing: Boosting Oil and Gas Recovery from the Earth,* 2005; Available from: http://www.compete.org/.
5. Council on Competitiveness, *Auto Crash Safety: It's Not Just for Dummies.* 2005; Available from: http://www.compete.org/.
6. Council on Competitiveness, *Spin Fiber Faster to Gain a Competitive Edge for U.S. Textile Manufacturing.* 2005; Available from: http://www.compete.org/.
7. Council on Competitiveness*, Customized Catalysts to Improve Crude Oil Yields: Getting More Bang from Each Barrel.* 2005; Available from: http://www.compete.org/.
8. Council on Competitiveness, *Full Vehicle Design Optimization for Global Market Dominance*. 2005; Available from: http://www.compete.org/pdf/.
9. Baker, M. and R. Buyya, *Cluster computing: the commodity supercomputer.* Software-Practice and Experience, 1999. **29**(6): p. 551-76.
10. Graham, S., M. Snir, and C. Patterson, *Getting up to speed: The future of supercomputing*. 2005: Natl Academy Pr.
11. Ridge, D., et al. *Beowulf: Harnessing the power of parallelism in a pile-of-pcs*. 1997.
12. Vacek, G., D. Mullally, and K. Christensen, *Trends in High-Performance Computing Requirements for Computer-Aided Drug Design.* Current Computer-Aided Drug Design, 2008. **4**(1): p. 2-12.
13. Hacker, T.J., et al., *Developing a Curriculum for High Performance Computing and Cyberinfrastructure Education.*, in *Proceedings of the American Society for Engineering Education Conference for Industry and Education Collaboration*. 2008: New Orleans, LA.
14. Patterson, D., G. Gibson, and R. Katz. *A case for redundant arrays of inexpensive disks (RAID)*. 1988: ACM New York, NY, USA.
15. Gray, J. and P. Shenoy. *Rules of thumb in data engineering*. 2000: IEEE Computer Society Press; 1998.
16. McKusick, M., et al., *A fast file system for UNIX.* ACM Transactions on Computer Systems (TOCS), 1984. **2**(3): p. 181-197.
17. Kleiman, S. *Vnodes: An architecture for multiple file system types in Sun UNIX*. 1986: Citeseer.
18. Schroeder, B. and G. Gibson. *Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you*. 2007.
19. Hey, T. and A. Trefethen, *The Data Deluge: Grid Computing - Making the Global Infrastructure a Reality.* 2003: John Wiley & Sons.
20. Wang, F., et al. *File system workload analysis for large scale scientific computing applications*. 2004: Citeseer.
21. Hargrove, W., F. Hoffman, and T. Sterling, *The do-it-yourself supercomputer.* Scientific American, 2001. **285**(2): p. 62-9.
22. Montante, R., *Beowulf and Linux: an integrated project course.* Journal of Computing Sciences in Colleges, 2002. **17**(6): p. 10-18.
23. Hacker, T. and K.M. Madhavan. *Developing a Research and Education Laboratory for High Performance Computing and Cyberinfrastucture*. in *Proceedings of the 2009 American Society for Engineering Education Conference*. 2009. Austin, TX. .
24. Membrey, P., et al., *The definitive guide to CentOS*, in *The expert's voice in open source*. 2009, Apress: Berkeley, Calif.
25. Team, C., *HOWTO: Create an OSCAR package, January 2004*.
26. Sloan, J., *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI (Nutshell Handbooks)*. 2004: O'Reilly Media, Inc.
27. Brim, M., T. Mattson, and S. Scott. *OSCAR: Open Source Cluster Application Resources*.

28. Petitet, A., et al., *Hpl-a portable implementation of the high-performance linpack benchmark for distributed-memory computers.* Innovative Computing Laboratory, University of Tennessee. http://www. netlib. org/benchmark/hpl.