Fall-11: Early Adoption of NSF/TCPP PDC Curriculum at Texas Tech University and Beyond

Yong Chen
Department of Computer Science
Texas Tech University
Lubbock, TX
yong.chen@ttu.edu

Yu Zhuang
Department of Computer Science
Texas Tech University
Lubbock, TX
yu.zhuang@ttu.edu

Noé López-Benitez
Department of Computer Science
Texas Tech University
Lubbock, TX
noe.lopez-benitez@ttu.edu

Abstract— **The Parallel and Distributed Computing (PDC) Curriculum developed by the National Science Foundation (NSF) and the IEEE Computer Society Technical Committee on Parallel Processing (TCPP) provides informative and insightful guidance in strengthening parallel and distributed computing education in computer science and computer engineering. This paper describes an early adoption effort of integrating the PDC Curriculum into the computer science undergraduate program at Texas Tech University. We present our curriculum change and design following the advice from the PDC curriculum along three lines: architecture, programming, and algorithms. We present our experiences, latest efforts, and observations as well.**

Keywords- Parallel and Distributed Computing Curriculum; computer science education; parallel architectures; parallel programming; parallel algorithms; Texas Tech University

I. INTRODUCTION

The advent of multicore processors has completely changed the landscape of computing. The serial computing era – when programmers could transparently and automatically take advantage of the microprocessors' performance improvement that follows the Moore's law – has gone [2]. Instead, parallel computing and distributed computing have now become universal, from the multicore computing and GPGPU (General-Purpose Graphics Processing Units) computing on chip, to the large-scale Grid computing and Cloud computing across wide-area networks [1]. All computer science undergraduate students need to study parallel and distributed computing principles and learn how to write parallel programs in order to prepare for their careers in this new computing era.

We believe that it is crucial for undergraduate students to grasp the parallel and distributed computing principles and problem solving skills, and thus initiated an effort to enhance the curriculum of parallel and distributed computing at the Texas Tech University (TTU). We were awarded an **Early Adopter status in Fall 2011** [1] for applying the Parallel and Distributed Computing (PDC) Curriculum developed by National Science Foundation (NSF) and IEEE Computer Society Technical Committee on Parallel Processing (TCPP) [3] to TTU. We found that the NSF/TCPP PDC Curriculum provides informative and invaluable guidance for enhancing the parallel and distributed computing curriculum, as well as a visionary long-term plan. In this paper, we present our curriculum change and design following the advice from the PDC curriculum. We discuss our experiences of adopting the PDC curriculum and latest efforts.

II. CURRICULUM DESIGN AND CHANGE

With the support of the NSF-TCPP Early Adopter Status award, we have incorporated PDC content into the undergraduate core course CS 3375 (Computer Architecture), and the elective course CS 4379 (Parallel and Concurrent Programming). Changes implemented include cutting-edge new technologies, APIs, and programming examples intended to increase students' enthusiasm about PDC. While CS 2350 (Computer Organization and Assembly Language Programming) remains a low-level course covering the basic concepts of computer architecture and organization, we have opted to revise this course to include new 64-bit architecture and organization models. These changes are necessary and consistent with the PDC curriculum. In addition, we have successfully introduced a new elective CS 4331 High Performance Computing in the summer of 2012 with a focus on hands-on experience and applications of PDC.

For CS 3375, we have expanded current content on instruction-level parallelism in the context of RISC machines, as well as added content on multiprocessor and multi-computer architecture. We dropped digital functional units while keeping data representation and I/O interfaces, memory and storage systems. These changes in CS 3375 were possible due to the content change of a prerequisite course the Electric & Computer Engineering's undergraduate course ECE 2372 (Modern Digital System Design), which has expanded coverage to include high-level circuitry and digital functional units beyond logic gates and transistor implementation of gates. The expanded coverage of ECE 2372 allows us to drop some hours of digital functional units in CS3375, making room for the inclusion of new PDC contents.

The major topics adopted from the PDC curriculum along the algorithm line include: notions of dependencies, task graphs (2 hours and 1 hour respectively), divide and conquer (2 hours), reduction (2 hours), communication including broadcast, scatter-gather (3 hours), and graph algorithms (3 hours). The major topics along the programming line include: concept of SPMD, data parallel model (1 hour), MPI model (1 hour). The topics covered along the architecture line include: data representation, arithmetic algorithms precision, rounding issues (3 hours), pipelining (3 hours), data and control hazards (1 hour), super-scalars (3 hours), Dynamic/Static scheduling (3 hours), Memory hierarchy (cached and virtual memory systems) (6 hours), multiprocessor systems (SMP, shared memory, message passing topologies) (3 hours), and multi-core systems (2 hours). We have also covered crosscutting topics such as the motivation of parallel and distributed computing, the power wall constraint, and performance issues.

For CS 2350, the contents of this course have been revised. As a result, topics related to computer architecture and organization includes now a software model (visible registers and memory organization) for 64-bit machines, specifically the Intel x64 model. This is an effort to include current changes in technology and already in the market. These changes are consistent with PDC objectives. However, due to timing and scheduling constraints, topics related to multi-tasking and task management have been dropped.

For CS 4379, we have redesigned the topics following the PDC curriculum. The major changes include moving multiprocessor and multi-computer architecture contents into CS 3375 (have been integrated into CS 3375), and increasing the focus on parallel programming, as well as strengthening the coverage of parallel algorithm basics. Programming topics cover programming paradigms for both shared-memory and distributed memory architectures, including shared address space as the programming paradigm for shared-memory machines and message passing for distributed architectures. For the shared address space paradigm, topics cover multi-thread programming using NVIDIA CUDA, and MPI for the message-passing paradigm. Matrix and graph problems, as well as algorithms, are included and adapted for programming teaching examples. Assignments are designed to help students understand major parallel performance-affecting factors and to gain hands-on programming experiences. Additionally, with the Early Adopter award support, **we have successfully made CS 4379 Parallel and Concurrent Programming a core course** in the undergraduate curriculum together with the Undergraduate Curriculum Committee. We believe this latest move will considerably help parallel and distributed computing education to undergraduate students at TTU.

We have also successfully offered **a new CS 4331 High Performance Computing elective course in summer 2012**. Even this course was introduced in the summer session, we received a high enrollment of 25 students (summer classes normally have 10-20 students). It was well received by students with an average evaluation of 4.3/5.0 and high attendance throughout the summer. This new elective course was introduced to focus on hands-on experience and applications of parallel and distributed computing for students. A group of undergraduate students who have taken the enhanced CS 3375/CS 4379 have found this CS 4331 elective is useful to apply the PDC knowledge to real-world high performance computing systems and applications. In addition, a team of eight undergraduate students with two females (six official members and two backups) trained through this course was selected **as one of finalist teams and participated in the internationally visible Student Cluster Competition at the 2012 ACM/IEEE Supercomputing Conference (SC12) in Salt Lake City**. In this competition, given fixed power consumption (26 amp power limit), a participating team will build a cluster to run given real scientific applications and High Performance Linpack (HPL) benchmark to compete for the performance and throughput. Even we did not win an award yet, students have found valuable experience with learning from PDC curriculum and applying the knowledge into real parallel and distributed computing problems. Two students from this team received job offers directly related with PDC trainings. Two other students have intended to continue advanced graduate education in PDC area.

III. Early Adoption Experiences

The NSF/TCPP PDC curriculum clearly provides a valuable guidance for designing and improving undergraduate curriculum for parallel and distributed computing education. It is the most comprehensive and systematic curriculum design and guidance in parallel and distributed computing education from our early adoption experience. Our experience has also confirmed that the PDC curriculum design including topics along the architecture, programming, algorithm, and crosscutting lines are not only constructive, but also lead students into a full awareness of the state of the art geared toward exploiting parallelism at all levels. The topics covered are essential, and the suggested hours and the Bloom level of expected learning outcomes are helpful as well. The discussions of the rationale of various topics are insightful. The suggestions on how to teach the topics are informative to assist the instructors to deliver the covered topics effectively. Given the importance of educating students with parallel and distributed computing knowledge, we believe that the PDC curriculum provides a timely, valuable, and comprehensive resource to assist institutions and instructors. Our experience of the early adoption effort suggests that the following aspects can potentially be further enhanced: 1) adopt real applications/real systems as the learning motivation and use them in teaching related topics (our experience with CS 4331 elective received positive feedback on this strategy); 2) provide project samples to have students reinforce and retain topics learned in class. These projects can even be a systematic cross-course project to assist the integration of multiple courses, which would be difficult to come up without a coordinated effort like the PDC curriculum design and implementation.

IV. Conclusion

The NSF/TCPP PDC curriculum has been adopted at Texas Tech University, and the early adoption experience has shown that the PDC curriculum provides a comprehensive and valuable resource for promoting parallel and distributed computing education. As parallel and distributed computing technologies become ubiquitous, it is important to elevate the PDC education across the nation. We have described our curriculum design and change, early adoption experience and feedback in this paper. We also seek further sponsorship or other grants to continue the PDC curriculum adoption effort with a goal of delivering quality parallel and distributed computing education at Texas Tech University.

Acknowledgment

This work is sponsored by the National Science Foundation and IEEE Computer Society Technical Committee on Parallel Processing via the NSF-TCPP Early Adopter Status award made to the Texas Tech University.

References

[1] Y. Chen, Y. Zhuang, N. López-Benitez. Early Adoption of NSF/TCPP PDC Curriculum at Texas Tech University. NSF-TCPP Early Adopter Status award. http://www.cs.gsu.edu/~tcpp/curriculum/?q=early-adopter-fall-11.html. 2011.

[2] W. Feng and P. Balaji. Tools and Environments for Multicore and Many-core Architectures. Special Issue of IEEE Computer, Dec. 2009.

[3] S. Prasad, A. Chtchelkanova, S. Das, et. al. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates, Dec. 2010 (latest release Dec. 27th, 2012).