Integrating Parallel and Distributed Computing Topics into Curriculum

Yingfeng Wang[1] and Xubo Wang[2]
[1]School of Information Technology, Middle Georgia State University
[2]Department of Mathematics, Middle Georgia State University

Abstract

To meet the need of teaching parallel programming to undergraduate students in a Bachelor program of Information Technology, we integrate Parallel and Distributed Computing (PDC) topics into the curriculum. In the adoption, we developed some teaching methods to respond to the challenges from both face to face and online courses. The experience of this study is expected to inspire other educators.

Introduction

The importance of parallel programming has been widely identified in computer education [1]. This study integrates PDC topics into the curriculum of the Bachelor program of Information Technology in School of Information Technology at Middle Georgia State University, which is a predominantly undergraduate university as part of Georgia University System. Before the adoption, the school of Information Technology had no course covering PDC topics.

This adoption includes integrating PDC topics to ITEC 3264 (Data Structures and Algorithm Analysis) and ITEC 4266 (C++ Programming) in Fall 2015 and Spring 2016, respectively, and developing a new course, ITEC 429F4 (Parallel Programming) in Summer 2016. We make slides, exercise questions, and videos for students to learn these topics.

Finished Work

In Fall 2015, we integrated some PDC topics into Data Structures and Algorithm Analysis (ITEC 3264), which had a face to face section and an online section in that semester. These PDC topics included the concept of time from parallel aspect, concepts of space and memory from parallel aspect, speedup, and the parallel version of divide and conquer algorithm.

To evaluate the adoption in ITEC 3264, we conducted some anonymous surveys. According to an anonymous survey of the online section, 100%, 86%, and 71% students agreed or strongly agreed that they understood the first, second, and third topics, respectively (Figure 1). Also 57% students agreed or strongly agreed they understood how to expose opportunities for parallel computation as what the parallel version of merge sort did. The anonymous survey of the face to face section shows 93% students considered they understood PDC topics introduced in the course, while half of them asked for more. Students were suggested to take the new parallel programming course in Summer 2016.

To strengthen students' understanding of PDC topics, an assignment of PDC topics was assigned. According to submissions of this assignment, all students were able to correctly explain at least three PDC topics, while 93.33% online and 85.71% face to face students correctly explained all PDC topics.

Based on the evaluation, most students understood PDC topics integrated into ITEC 3264. They also showed interests to learn more. The assignment of PDC topics successfully strengthened students' understanding in this area.

Ongoing Work

In Spring 2016, we work on integrating PDC topics to a C++ Programming course (ITEC 4266). Integrated topics include SIMD, concepts of shared memory and distributed memory, basic compiler directives and pragmas for shared memory programming (e.g. OpenMP), and creating parallel tasks using explicit tasking (e.g. PThreads). Because Students are recommended to use Visual Studio 2012 in this course, we briefly introduce how to set Visual Studio for enabling OpenMP, and also introduce a plugin for using PThreads on Visual Studio.

In this semester, ITEC 4266 is taught as an online course, in which the interaction and communication between the instructor and students is usually much less than that in traditional face to face courses. This is a challenge for diverse students' background. Some good students may be eager to learn more besides online regular teaching materials. To address this issue, we designed a special project for giving some students an opportunity for studying more upper level parallel programming skills through a realistic program.

When designing this project, we consider two factors. Firstly, this project should engage students' interests. Otherwise, students may easily feel boring. In the beginning of this semester, the news of AlphaGo [2] has attracted a lot of attentions. So, we pick Stockfish [3], one of the best chess software, as the target parallel program. Stockfish is open source software released under GPL license, which gives us enough flexibility in this teaching project. Secondly, this project is also designed to give students enough time to learn parallel programming skills.

Since students usually have no knowledge of C++ in the beginning of this course, students would have to work in a rush, if all goals of this teaching project are planned to accomplish within one semester. To overcome this problem, we plan to divide this teaching project into two parts. In Spring 2016, students focus on understanding the source code of Stockfish, especially the part of parallel programming. In the following semester, student will try to optimize its parallel programming code to adapt the hardware of a specific machine. Students, who are interested in this project, can register a special project course in Summer 2016 for continuing this project.

Future Work

In Summer 2016, we will develop a new course named parallel programming (ITEC 429F4). In this course, we will teach basic parallel computing, shared memory and distributed memory programming, and computation decomposition strategies, and illustrate some examples using parallel algorithms. In traditional parallel programming courses, C/C++ and Java are major teaching languages. However, these computer languages require a very high bar of programming skills for students. To guarantee the success of this course, we have to consider the situation of students at Middle Georgia State University, which is a predominantly undergraduate university. To help students quickly be engaged, Python will be used as the teaching language, which all students should have learned before taking this course. The syntax of Python is much easier than that of C/C++ and Java, and Python 3 provides mature features for multi-processing and multi-threading programming. We expect this adjustment will smooth the learning curve.

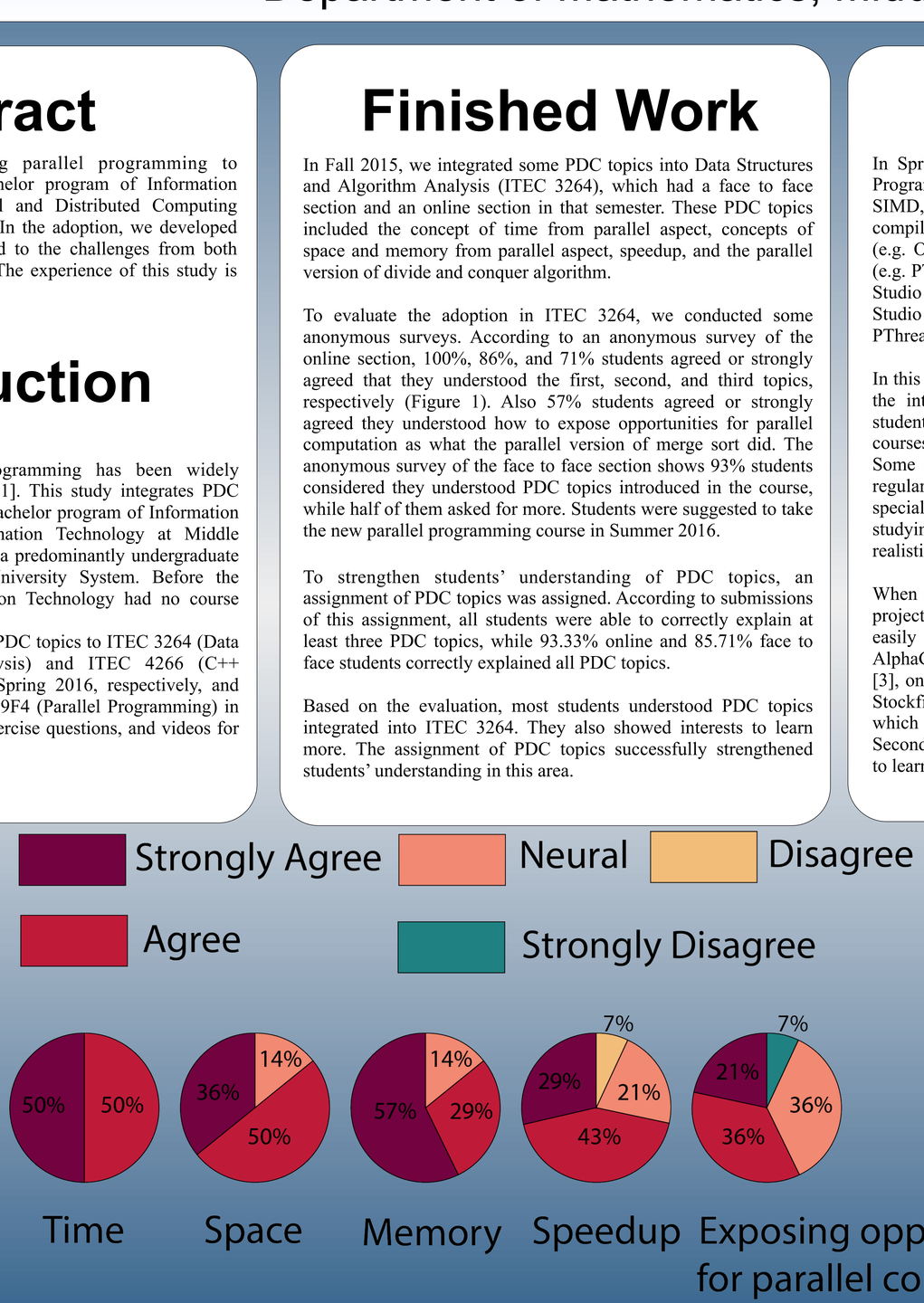

Figure 1: Survey results of students self-evaluation regarding the understanding of PDC topics in the online section of ITEC 3264. Students were asked whether they thought they understood related PDC topics and terms.

Conclusions

Updating curriculum by integrating PDC topics is challenging. This study applies several teaching methods in this adoption. We expect our experience will benefit other educators.

Acknowledgement

The authors thank Zhiyi Xie for his help in using Stockfish. This project is supported in part by Yingfeng Wang's NSF/IEEE-TCPP/CDER Center Early Adopter Award and Quality Enhancement Plan (QEP) Course Redesign funding.

Reference

1. **R. Miller, "The status of parallel processing education," Computer (Long. Beach. Calif)., vol. 27, no. 8, pp. 40–43, 1994.**
2. **E. Gibney, "Google AI algorithm masters ancient game of Go," Nature, vol. 529, pp. 445–446, 2016.**
3. **"Stockfish." [Online]. Available: https://stockfishchess.org/.**