Integration of PDC topics with Various Computer Science and Engineering Undergraduate Courses

Sandeep Kumar, Manoj Misra, Sudip Roy
Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India
sgargfec@iitr.ac.in, sandeepkumargarg@gmail.com

ABSTRACT

Parallel and Distributed Computing (PDC) is extensively used in these days in industries for designing various methods and algorithms. Hence, there is a continuous boost in the need of skilled persons in the concerned area. Motivated from this, we planned to integrate selected PDC topics from the TCPP Core Curriculum in teaching with various courses of Computer Science and Engineering at undergraduate level in our department. In this paper, we present our methodology that is adopted for this department-wide integration of the PDC topics with multi-courses.

Keywords

Parallel and Distributed Computing, PDC, CS Education, Curriculum.

1. INTRODUCTION

The usage of Parallel and Distributed Computing (PDC) based technology is finding its place in diverse sections of manufacturing and development. These days, PDC based technology is being used extensively in designing various methods and algorithms in government laboratories (such as DRDL, ISRO, DRDO etc.) as well as in private industries (such as NVIDIA, Intel, Facebook, Twitter, Adobe Research etc.). Also, a lot of public and private sector events and initiatives are emphasizing the need of adopting parallel thinking at the academic and research levels. So, it is felt that the formal training of the students and making them to ‘think in parallel’ while performing design or development can be beneficial to them in all their future endeavors.

Motivated from the TCPP Core Curriculum Initiative [1] and the market-needs, it is planned to integrate the PDC topics from TCPP Core Curriculum in teaching of various courses of Computer Science and Engineering at undergraduate level.

We initially analyzed the complete course structure of our undergraduate program. After the recent revision, we have incorporated some elective courses with the theme matching with the TCPP core curriculum such as Parallel and Distributed Algorithms, Cloud Computing, High Performance Computing, and Distributed Systems. Further, it is seen that some of the core courses such as Introductory Course on Computer Science, Introductory Course on Programming, Data Structures, Algorithm Design, Compiler Design, and Programming Languages mostly cover the topics related to PDC as per integration plan proposed in the TCPP Core Curriculum Initiative [1]. After going through the TCPP Core Curriculum Initiative document, we learned that some of the other remaining courses can be targeted for integration of PDC topics. Hence, we planned to integrate the topics from TCPP core curriculum into our courses on Computer Architecture, Computer Networks, Network Programming, Networks based Programming laboratory, Undergraduate Software Project, Software Engineering, and Industry-oriented project as part of the early adopter project. Out of these, the courses on Computer Architecture, Computer Networks, Network Programming and Networks based Programming laboratory are taught in Fall-semester at second year and pre-final year level to undergraduate students, and the remaining courses are taught in the Spring-semester to second year and final year students. Apart from the topics mentioned in the TCPP core curriculum for integration in these courses, we planned to have some project component in the course to enable integration of some more topics from PDC with this course. It enables the students to also understand the programming and software development practices to be kept into mind specifically for PDC related software development. Also, it enables the students to ‘think in parallel’ while problem-solving, algorithm design, architecture design, and developing the full projects. It is felt that this integration exposes the students to PDC from systems, programming, architecture, and algorithms perspectives. It is planned to conduct research-oriented teaching through discussion of emerging trends, assigning projects on research problems, conducting critical reviews, and discussing new ideas. We also planned to have the industry collaboration for taking feedback and conducting some invited lectures related to these courses from industry experts.

2. PROJECT OBJECTIVES

Besides the basic objectives of the related courses, we tried to accomplish many pedagogical objectives corresponding to the PDC course:

- To ensure that every undergraduate student of our department is exposed to parallel and distributed computing from systems, programming, architecture, and algorithms perspectives.
- To provide students with more opportunities to work with PDC related projects.
- To provide students with ample occasions to 'think in parallel'.

To fulfill these objectives, we analyzed the complete course structure of undergraduate level and have indentified the courses as mentioned in Section-1, which can be targeted for integration of PDC topics. We have chosen the courses which are compulsory courses to be learned by every student for proposed integration of PDC topics. The courses are taught in Fall and Spring semesters at second year, pre-final year, and final year level to undergraduate students. Through these courses, it is intended to cover many important PDC topics. Apart from these courses, various fundamental courses in our course-structure, as mentioned in Section-1, are already covering most of the fundamental topics related to PDC especially from the algorithms, systems, and architecture perspectives.

3. INTEGRATION PLAN

In our course curriculum, the emphasis is already on algorithmic problem solving through design and development. Many core level courses such as Introductory Course on Computer Science, Introductory Course on Programming, Data Structures, Algorithm design, Compiler design, and Programming Languages and some elective courses on Parallel and Distributed Algorithms, Cloud Computing, High Performance Computing, and Distributed Systems already have focus and coverage on some fundamental topics of PDC. Through this initiative, it is intended to widen the scope of integration of PDC topics at departmental level, in multi-semesters, and to add a flavor of parallel thinking in the other subjects especially in the subjects of Computer Architecture, Computer Networks, Network Programming, Networks based programming laboratory, Undergraduate Software Project, Software Engineering, and Industry-oriented projects.

We have gone through the document titled 'NSF/IEEE – TCPP Curriculum Initiative on Parallel and Distributed Computing-Core Topics for Undergraduates, Version-1, Dec 2012' [1]. We really appreciate the efforts in defining the detailed guidelines for integrating the various topics of PDC into many other knowledge areas. Our project is based upon the thinking underlying the TCPP initiative that exposure to PDC topics should not be through one independent course only but it should be in integration with several courses so that there are multiple opportunities and contexts for the students to apply and learn PDC. This project is highly encouraged from the thinking of the TCPP that mentions that if an instructor is talking about the parallel and distributed context of some topics under discussion in the class even if the topics have no obvious parallel or distributed content, then it will naturally expand the thinking of students in parallel and distributed terms. In line with this thinking of the TCPP, our team has prepared the course structures for courses of Computer Architecture, Computer Networks, Network Programming, Networks based programming laboratory, Undergraduate Software Project, Software Engineering, and Industry-oriented project by integrating PDC topics to be covered in class.

It is intended to deliver the teaching through various modes such as discussion, assignments, laboratory programs, problem solving, and projects. It is planned to conduct research-oriented teaching through discussion of emerging trends, assigning projects on research problems, conducting critical reviews, and discussing new ideas.

4. EVALUATION

Each of the courses is of one semester length and evaluation of the students is done through continuous assessment via various modes: mid-term written examinations, quizzes, mini-projects, assignment, presentations, and end term written examinations.

In addition to the regular evaluation for the corresponding subjects, one of our aims is to assess the students' ability in thinking effectively in parallel and distributed concepts. It is assessed through specific questions and problems during regular assessment. The mid-term written examination (1.5 hours) is given 35% weightage for Computer Networks and 25% weightage for other mentioned courses, while the end-term written examinations (3 hours) is given 50% weightage. Rest of the 15% weightage for Computer Network course/25% for the other mentioned courses is devoted to the class work performance factor measured through various activities such as quizzes, mini-projects, assignments or presentations.

5. ACKNOWLEDGMENTS

The Authors are thankful to NSF, Intel, and the project coordinator for Early Adopter Award—Fall 2015 and the grant for presenting this paper.

6. REFERENCES

[1] Prasad, S. K., Chtchelkanova, A., Dehne, F., Gouda, M., Gupta, A., Jaja, J., Kant, K., Salle, A. L., LeBlanc, R., Lumsdaine, A., Padua, D., Parashar, M., Prasanna, V., Robert, Y. , Rosenberg, A., Sahni, S., Shirazi, B., Sussman, A., Weems, C., Wu, J. 2012. NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing: core topics for undergraduates. Version I, http://www.cs.gsu.edu/~tcpp/curriculum/?q=home.