Novel approach for teaching a course on "Parallel Computing Systems"

P.Chitra, Professor,

Department of Computer Science and Engineering

Thiagarajar College of Engieering, Madurai, India

pccse@tce.edu

***Abstract*—The course on Parallel Computing Systems (PCS) is designed to provide an understanding of the fundamental principles and engineering trade-offs involved in designing modern parallel computing systems as well as to teach parallel programming techniques necessary to effectively utilize these machines. Because writing good parallel programs requires an understanding of key machine performance characteristics, this course will cover both parallel hardware and software design. This poster hgihlights on the activity based learning approach for traching the course. Applying this methodology gave us a significant progress in parallel applications designed by students, where we have observed that the course outcomes are significantly met at the end of the course.**

I. INTRODUCTION

In the recent days, the scientific and engineering applications are designed with more complexity and precision, in order to obtain more accurate results. Hence, we need parallel environments to address this complex applicaions. Parallel application development has been included in various areas such as biology, physics, chemistry, etc. and this necessiated that parallel programming course had to be included in the teaching curriculum.

The major course outcomes of the course as follows:

- CO1: To select the appropriate parallel programming model for the given application.
- CO2: Apply the constructs of parallel programming model to convert a sequential program to parallel program.
- CO3: Design protocols for ensuring cache coherence using the directory based and snooping class of protocols.
- CO4: Develop parallel programs using OpenMP and MPI constructs.
- CO5: Characterize the benefits of using a GPU versus CPU for a typical parallel application.

II. SAMPLE ACTIVITIES CONDUCTED DURING THE COURSE

Table 1. Shows the different activites conducted during the course to make the student understand the different course outcomes.

TABLE I. ACTIVITIES CONDUCTED DURIONG THE COURSE

To make the understand the concept of thread and parallelism	• A question to find the youngest student of the entire class was posed • The students followed different strategies • The concept of threading was illustrated by grouping them according to the seating desk wise (row) and asking them to find the youngest among them. • The concept of computation and communication between the rows to arrive at the result was illustrated. • The importance of the performance metric Communication to Computation ratio was demonstrated.
To make the students understand the importance of cache memory in improving the performance	• Make the student visualize the kitchen in their home. • Identify the items to be placed on the Kitchen top near the stove. Illustrate the principle of locality of reference and mapping by placing the identified items on the kitchen top.
To make the students understand the cache coherence	A role play was organised with the students bearing the pluck cards titled "M", "E", "S", "I" and illustrating the corresponding changing states of cache.

Table 2 shows the sample Interim assessment methodologies used for assessing the progress of the students understanding and implementation skills.

TABLE II. INTERIM ASSESSMENT METHODOLOGIES

Category	Assessment Tool	Assessment Period	Course Outcome	Topics
Knowledge	Worksheets	Fourth Week	CO1 and CO2	Choice of appropriate strategies for parallelizing an application.
Communication Skill	Seminar	Mid of the semester	CO3	False sharing problem in recent parallel computing system
Professional Skill	Mini Project*	Two Weeks before Last working day	CO4 and CO5	Using one of Parallel programming paradigms

Mini project taken over by a team of 2–3 students. Topics assigned include: Parallel Implementation of Gauss-Seidel Algorithm for Linear Systems, and Mandelbrot Set, Odd-Even sorting,etc.

III. EFFECTIVENESS OF THE TEACHING METHODOLOGY

- Measured at the end of the term using an on-line survey,containing a set of short questions, inquiring about the level of achievement of the learning outcomes.Furthermore, the actual evaluation of the same course outcomes are also assessed using formative and summative assessment techniques.
- The items assessing the individual outcomes are evaluated separately and shown in the same table to see whether the student perceptions are consistent with the instructors' actual measured results. Of the 62 students enrolled in the course, 56 students completed the survey and both the results of student and instructor evaluations are reported. The table represents the perception of the students and the actual evaluation of the instructor(s) based on the Continous assessment tests (40%), assignments/project (20%), and the Final examination (40%).

TABLE III. EVALUATION OF THE TEACHING METHODOLOGY

Sample Course Outcome	Result of Survey (scale 1-10)	Instructor evaluation (scale 1-10)
Apply the constructs of parallel programming model to convert a sequential program to parallel program.	8.8	8.0
Design protocols for ensuring cache coherence using the directory based and snooping class of protocols.	8.1	8.5
Develop parallel programs using OpenMP and MPI constructs	8.3	7.8

IV. CONCLUSION

In this poster, an innovative approach to teach the parallel computing course. The proposed approach combined several ingredients to improve the effectiveness of learning. The students were engaged with visualizations, role play, and practical exercises. Moreover, we provided the students with informative notes and source code. We also tested the effect of active learning on students' understanding of parallel programming. Our evaluation shows a significant improvement in understanding theoretical and practical topics of parallel computing. Further, the active learning approach seems to give students an advantage in terms of allowing them to engage in the material and learn it more effectively. To complement our efforts, we allowed the students to practice parallelization topics, by asking them to use parallelism in their mini projects. Overall the result was positive with a majority of students successfully implementing parallel techniques into their own code.