Fall 2012: Early Adopter Plan for Teaching Concurrent and Distributed Systems

Jawwad A. Shamsi
Computer Science , FAST National University , Karachi, Pakistan. Jawwad.shamsi@nu.edu.pk

Abstract— **This paper is focused on describing methodology for teaching Parallel and Distributed Computing topics to undergraduate students. Through an elective course of Concurrent and Distributed Systems, the course incorporate recommended PDC topics. There are many important pedagogical goals of this course. First the course is aimed in promoting interest of the students. Second, the course is designed to develop programming and system development skills. In addition, the contents of the course have been formulated to equip students with many emerging topics. Through programming-based skills, interactive teaching style, and incorporating student evaluations, the course receives high student ratings and teaching evaluations. The paper describes the course methodology and explains mechanism for course design and implementation.**

Keywords- Early Adapter Plan, Concurrent and Distributed Systems, TCPP recommended Curriculum

I. INTRODUCTION

The curriculum of Parallel and Distributed Computing (PDC) entails a wide range of topics. Further, the subject matter includes both theoretical contents and programming techniques. These characteristics imply that PDC can be taught by using numerous methodologies and by incorporating a wide range of curriculum contents. A significant requirement in teaching PDC curriculum is to adopt the contents and teaching practice according to the interest of the students.

Therefore, teaching PDC topics requires numerous considerations related to contents, teaching style, and delivery methodology. The purpose of this paper is to briefly describe the course of Concurrent and Distributed Systems, which has received early adaptor status in Fall 2012. The focus has been on effective teaching while incorporating recommended topics.

II. COURSE DESIGN

The course is designed to include recommended PDC topics while adopting to effective teaching methodologies. Since the course is offered to students of CS major, a major focus of the course is to teach PDC topics with the aid of programming skills and methodologies.

A. *Pedagogical Goals*

The course is aimed towards following pedagogical goals

G1: Explain fundamental and theoretical concepts
G2:Develop programming skills in PDC
G3: Incorporate student feedback and evaluations
G4: Teach applied and advanced topics

B. *Course Structure*

The structure of the course is based on TCPP recommended topics of PDC. Emphasis is on two important tracks of 'programming' and 'cross cutting'. Table 1 describes the bloom-level implementation of the recommended PDC topics. Table 2 shows a list of programming assignments which have been included while considering the bloom-level learning of course contents.

Table 1 – Plan for Teaching PDC topics

Topics	Bloom-Level	Expected Outcome
Client Server, task/thread spawning	A	Using TCP and UDP sockets for creating client/server applications. Using fork and exec for task creation and handling.
Tasks and Threads, Synchronization, Concurrency defects	A	Ability to write multi-threaded and multi-processes programs using pthreads and fork. Using locks, and semaphores to handle critical region. Comprehension of deadlocks, race conditions, join, and wait. Methods for Inter-process communication.
Security	C	Network authentication mechanisms. Denial of Service Attacks, Kerberos, Single Sign on Systems
Cloud	K/C	Understanding of Cloud Architecture, virtualization, on demand scaling, and resource provisioning.
P2P	A	Overlay nodes, Development of P2P-based look up service.

Table 2 – Programming Assignments

Assignment Description	
A1	Simple UDP-based client Server program
A2	Request/Response based network monitoring tool using multithreaded semantics.
A3	Develop Unix Shell
A4	Develop P2P-based file lookup Service

Table 3 – Weekly Plan

Week	Topics
1	Introduction to Distributed Systems. Socket Programming.
2	Socket Programming: UDP and TCP Programming. Assignment 1
3	Quiz 1. Concurrent Programming. Socket Options, Thread Programming
4	Thread Programming, Mutual Exclusion,
5	Thread Programming. Assignment 2. Project Proposals Due.
6	MidTerm1. Race Condition, Deadlocks
7	Process Programming, exec.
8	Quiz 3. Process Programming
9	Signals
10	Pipes. Assignment 3
11	Midterm 2
12	Applications of Distributed Systems: Overlay Networks
13	Assignment 4. Applications of Distributed Systems: Network Security
14	Quiz 4. Applications of Distributed Systems: Network Security
15	Cloud Computing and Distributed File Systems
16	Project Evaluations.
	Final Examination

Table 4 – Overall marks given by the students

Max=10, Min=7.5, Mode and Median=8, Average=8.44

1) Programming

To maintain students' interest many programming topics are covered. Class lectures are based on teaching methodology to develop effective concurrent and distributed systems. Contents such as socket programming, thread programming, locks, process, methods for inter process communication, and signals are incorporated.

2) Cross Cutting

Cross cutting topics include peer to peer system development, theory for P2P systems, network security, DoS attacks, Single Sign on Systems, Cloud Computing, and Distributed File Systems are included.

C. Evaluation of Students

Student evaluations are integral part of the course. For this purpose, programming, assignments, quizzes, projects, and mid-term and final examinations were conducted.

D. Student Feedback

Three set of feedback were conducted. Initial feedback was conducted to assess the students' requirements and capabilities. A mid-semester feedback was taken to assess the strengths and weaknesses of the course and incorporate remedial actions.

At the end of the course, a final student survey is conducted to assess the achievement of goals and to evaluate the overall success of the course. 34 out of 36 students participated in the course, anonymously. Results in table 4 confirm that the course methodology is liked by all the students. During the survey, the students also answered about the approaches adopted during the course.

Programming is the most liked approach of the students. Out of the four programming assignments, the Unix-shell assignment (A3) turned out to be the most difficult assignment for the students. In that, only a few students were able to implement the redirection operator using Unix pipes. Comparatively, the P2P assignment received the highest scores. This is perhaps they were inspired by the topic of P2P.

Students also acknowledged that interactive teaching style, group discussions, and peer learning contributed successfully towards their learning. Specifically, the students mentioned that Socket, Thread, and P2P Programming, Security, Cloud Computing, are the most interesting topics.

The students also suggested that the topic of Cloud Computing should be covered in more depth. Further, many students believed that programming aspect should be incorporated in the topic of network security.

III. LESSONS LEARNED

Following are the important lessons learned.

- Programming provides an effective method for learning.
- TCPP recommended curriculum is well designed.
- Multiple courses are needed in order to comprehensively cover the course.

- Cutting edge topics such as Cloud computing, network security, and distributed file system are very popular among students
- Interactive learning, peer discussion, and group discussions are effective in teaching.

ACKNOWLEDGMENTS

The author is thankful to TCPP for providing recommended topics. In addition, the author is thankful to Intel and NSF for facilitating grant through early adapter plan.