A Comprehensive Parallel Computing Curriculum: From Second Year to Professionals

Peter E. Strazdins*,
* Research School of Computer Science, The Australian National University,
Email: Peter.Strazdins@cs.anu.edu.au

***Abstract*—The Australian National University has had a long tradition of teaching all aspects of Parallel Computing. It begins with introductory elements of parallelism in a second year computer organization course, and is followed up by a later second year course in Concurrent and Distributed Systems. The curriculum progresses with a High Performance Scientific Computation course in 3rd year, and is capped with a Parallel Systems course in 4th year. A specialty Multicore Computing course was also taught at this level. Recently, we also began delivering one-week intensive courses to external professionals; these have been in Shared and Distributed Memory HPC.**

I. Overview

Since the early 2000's there has been a comprehensive parallel computing curriculum throughout the undergraduate computer science program.

The relevant curriculum for the course begins with COMP2300[1] *Introduction to Computer Systems*, which introduces parallelism within and between instructions. Concurrency issues are introduced in the course COMP2310 *Concurrent and Distributed Systems*; in particular, the concepts of threads and shared objects. These two courses comprise the assumed knowledge for the further parallel computing courses.

The third year course COMP3320 *High Performance Scientific Computation* covers data modelling, programming (including shared memory models) and program performance issues for high performance scientific applications.

The capstone course is however the fourth year course, COMP4300 Parallel Systems, which provides a practically oriented introduction to the special case of concurrency of shared and distributed memory parallel computers. It mostly deals with 'classical' forms and paradigms of parallelism, and then compares and contrasts these to state-of-the-art systems.

A further capstone course, COMP8320 *Multicore Computing: Principles and Practice*, [1] is a specialized multicore computing course available for advanced undergraduate and postgraduate students. It provides an in-depth study of the principles, the architecture and programming paradigms of contemporary and emerging multicore (parallel) processors. As such, it was the most advanced and specialized course in our curriculum. The course ran for a first time in the second half of 2009 and again in 2011, with approximately a dozen students each time. This course must necessarily specialized in multicore aspects due to the existence of COMP3320 and COMP4300. It covered the most advanced aspects of computer architecture, concurrent programming and computer performance themes. It also exposed students to the depths of the then start-of-the-art multicore technology: The UltraSPARC T2, the Kepler GPU and the Intel Single Chip Cloud Computer [1].

II. Design Philosophy: Capstone Courses

COMP4300 and COMP8320 were designed in order to prepare advanced undergraduate and postgraduate students for the rapidly unfolding future of parallel / multicore computing. The material is based on relevant research and practice of the Computer Systems group at the ANU, and the method of delivery is termed as *research-based education* [2]. The student numbers are intended to be small in order to facilitate more personal contact with the instructor, using the *Cognitive Apprenticeship* approach [3].

While these courses emphasize computer architecture and operating system concepts, they are based on the context of a programmer undertaking the task of writing correct, portable and efficient programs. A key goal is to teach how architectural effects relate to changes in program performance when a program or system parameter is varied. In order to better achieve this aim, the courses also had a strong emphasis on the use of software tools (e.g. profilers, instrumented runtimes), which can provide more detailed insights.

III. Intensive Parallel Computing Courses fro Professionals

In recent years, we were approached by two computing professional organizations to teach HPC courses in the form of non-assessed one-week intensive courses. We designed separate courses for shared and distributed memory HPC, based upon an expansion of the material in COMP4300. We found the format of one hour lecture followed up by one hour hands on exercises (based on the lecture) worked well. Web pages for the distributed memory course can be found at http://cs.anu.edu.au/courses/distMemHPC.

References

[1] P. E. Strazdins, "Experiences in Teaching a Specialty Multicore Computing Course," pp. 1271–1276, May 2012.

[2] A. Jenkins, R. Breen, and R. Lindsay, *Reshaping Teaching in Higher Education: Linking Teaching with Research*. UK: Kogan Page, 2003.

[3] A. Collins, J. S. Brown, and A. Hollum, "Cognitive Apprenticeship: Making Thinking Visible," *American Educator*, Winter 1991, http://www.21learn.org/arch/articles/brown_seely.html.

[1] Web pages are at http://cs.anu.edu.au/courses/COMP2300; most of our courses have pages with similar URLs.