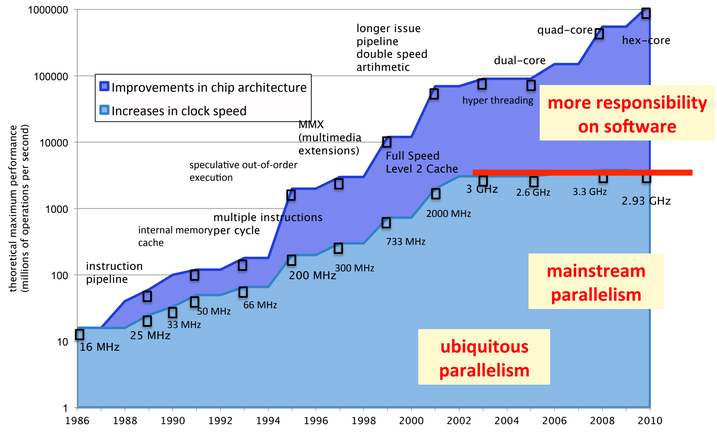
Preparing Computer Science Students for An Increasingly Parallel World
Teaching Parallel Computing Early and Often

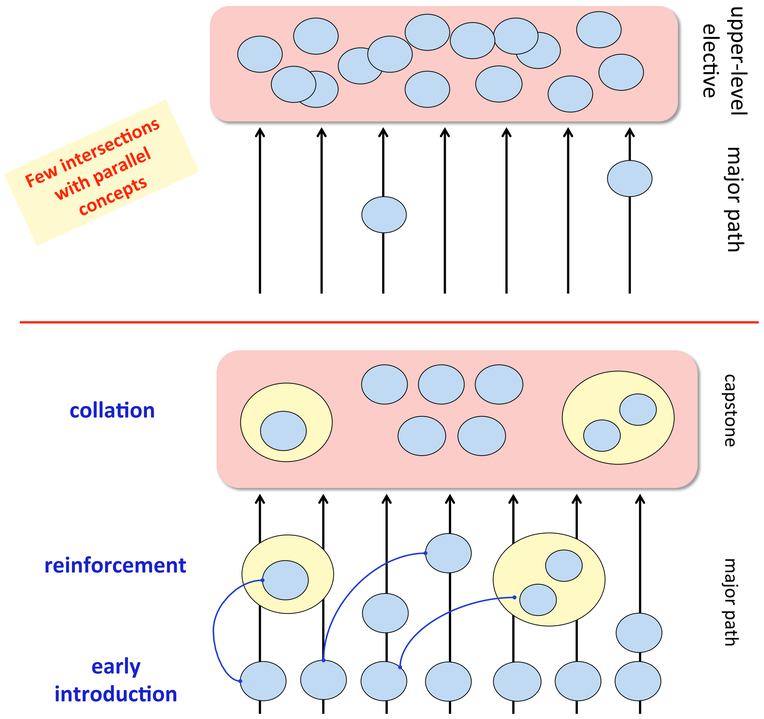
Martin Burtscher, Wuxu Peng, Apan Qasem, Hongchi Shi, Dan Tamir
Texas State University

The widespread deployment of multicore-based computer systems over the last decade has brought about drastic changes in the software and hardware landscape. However, most undergraduate computer science (CS) curricula have not embraced the pervasiveness of parallel computing. In their first years, CS undergraduates are typically exclusively trained to think and program sequentially. However, too firm a root in sequential thinking can be a non-trivial barrier for parallel thinking and computing. Thus, there is an urgent need to teach multicore and parallel computing concepts earlier and often in CS programs.

This project addresses the rapidly widening gap between highly parallel computer architectures and the sequential programming approach taught in traditional CS courses. It proposes to systematically integrate parallel computing into current undergraduate curricula. Specifically, its goals are to develop course modules and projects for introducing parallel computing concepts in several early computer science courses, to design an upper-level multicore programming course that serves as a capstone for parallel computing concepts, and to promote this model by making all relevant material freely available.



The Early and Often Approach

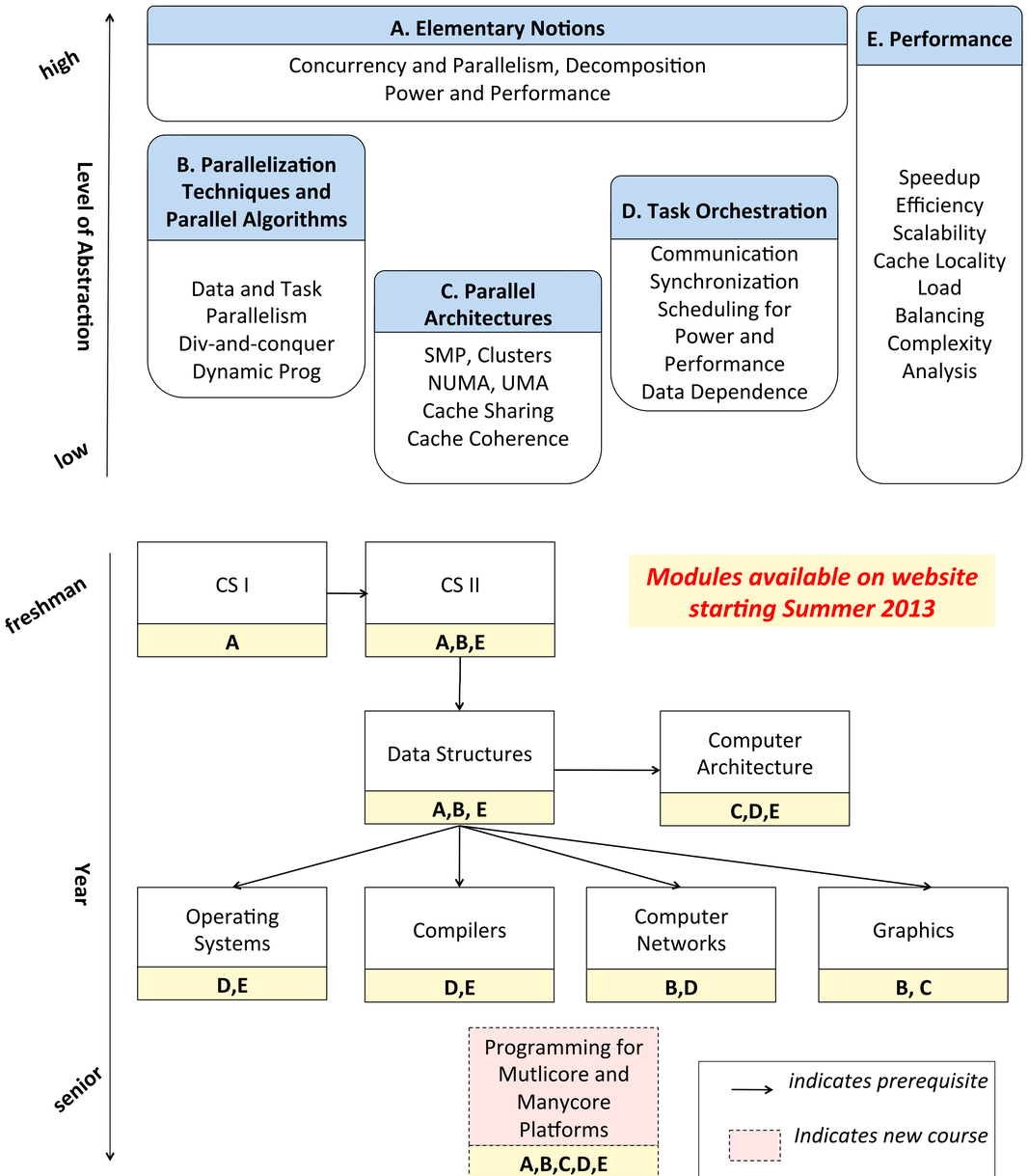


Breaks away from traditional mode of teaching parallel programming as an upper-level elective

- Introduce parallelism early in the curriculum in required courses
- Repeat key concepts in different courses throughout the curriculum
- Tie concepts together in an upper-level capstone course

Module Principles

1. modules should be classified and introduced based on **level of abstraction**
2. modules should provide parallel context to existing **content**
3. modules should be self-contained for **easy adoption** across different institutions



Modules available on website starting Summer 2013

Module Characteristics

Length : 1-3 lectures

Includes
- Lecture notes
- In-class examples
- Exercises and solutions
- Programming assignments
- Software support
- Learning outcomes
- Assessment guidelines

(follows CSinParallel model
http://cisinparallel.org)

Example Module

Name: Task Orchestration - scheduling and mapping

Topics: Scheduling algorithms for multi-processor systems, affinity-based scheduling, energy-aware scheduling

Length: 2 lectures (@ 1 hour and 20 minutes each)

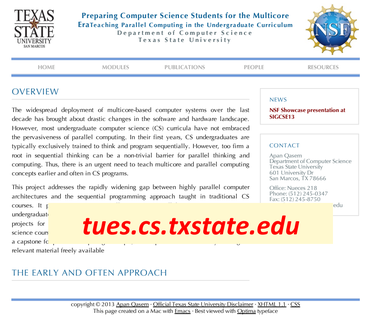
Projects: Derive an optimal affinity-based schedule for a loop-based application (e.g., an n-body simulation). OpenMP implementation and supporting framework provided with module

Course: Operating Systems

Context: OS Scheduling

Assessment: programming project, one exam question

Website



Project supported by the National Science Foundation under award DUE-1141022, CNS-1253292 and CNS-1217231

Contact
Apan Qasem
apan@txstate.edu
http://tues.cs.txstate.edu