Early Adopter : Multiprocessor Programming in the Undergraduate Program

NSF/TCPP Curriculum: Early Adoption at the University of Central Florida

Narsingh Deo Damian Dechev Mahadevan Vasudevan

Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816

May 16, 2011

Parallel Programming Education at UCF

COP 4520: Concepts in Parallel and Distributed Computing

- Elective senior computer programming class
- Prerequisites
  - COP 3503: Sequential Algorithms and Data Structures
  - COP 3402: Systems Software
- Carries: *3* semester hours, *45* in-class instruction hours
- Spring *2011: 33* enrolled students
- Scope
  - parallel graph algorithms
  - principles of distributed computing
  - programming models
  - frameworks for parallel processing

TCPP Topics and Integration Plan

COP 4520: place for highly motivated and ambitious undergraduate students

Prior to Spring 2011

- Fundamentals (3 weeks)
  - taxonomy and architectures
  - data vs. control parallel approach
  - algorithm analysis, running time, speedup, cost/work, efficiency
  - Brent's scheduling
  - Amdahl's law, Gustafson's law
- Cluster Computing with MPI (5 weeks)
  - overview of cluster architecture, granularity constraints
  - message passing, frequently used functions
  - sample applications, using the UCF cluster
  - parallel software development
  - team project involving substantial parallel programming

TCPP Topics and Integration Plan (cont'd)

COP 4520:

place for highly motivated and ambitious undergraduate students

Prior to Spring 2011

- Designing, Implementing, and Evaluating Parallel Algorithms (remainder of the class)
  - Prefix-sums / list-ranking, finding the max of a set, sorting
  - matrix problems: matrix partitioning, matrix multiplication
  - Gaussian elimination
  - graph problems: all-pairs shortest paths: Dijkstra's, Warshall-Floyd, minimum spanning tree
  - performance comparisons
- Overview (last week of classes)
  - Recent advanced: programming models, architectures
  - P-completeness:
    a glimpse of the problems that resist parallelization

TCPP Topics and Integration Plan (cont'd)

COP 4520:

the paradigm shift in our core computing architecture requires a fundamental change in how we program

Spring 2011

- Core Topics I
  - introduction to multi-threading and multiprocessor synchronization
  - design of highly concurrent data structures and algorithms
  - lock-free synchronization
  - software transactional memory (STM) models
  - programming tools and techniques for parallel computing
  - program analysis tools (such as Intel Pin, Valgrind, ROSE Compiler)

TCPP Topics and Integration Plan (cont'd)

COP 4520:

the paradigm shift in our core computing architecture requires a fundamental change in how we program

Spring 2011

- Core Topics II
  - emerging parallel programming models (Intel TBB, Intel Ct, STM)
  - recent advances and future trends in concurrent programming
  - validation and verification of parallel processes
  - industrial applications
  - heterogeneous platforms (CPUs, GPUs, FPGAs)
  - hardware-software co-design
  - advanced simulation tools

TCPP Topics and Integration Plan (cont'd)

COP 4520:

the paradigm shift in our core computing architecture requires a fundamental change in how we program

Spring 2011

- Lectures I
  - mutual exclusion
  - concurrent objects, consistency and semantics
  - shared memory data structures
  - synchronization primitives, transactional memory
  - spin-locks, read-write locks, contention
  - nonblocking data structures: linked-lists, queues, vectors, hash tables
  - hazardous concurrency bugs: ABA Problem, race-conditions

TCPP Topics and Integration Plan (cont'd)

COP 4520:

the paradigm shift in our core computing architecture requires a fundamental change in how we program

Spring 2011

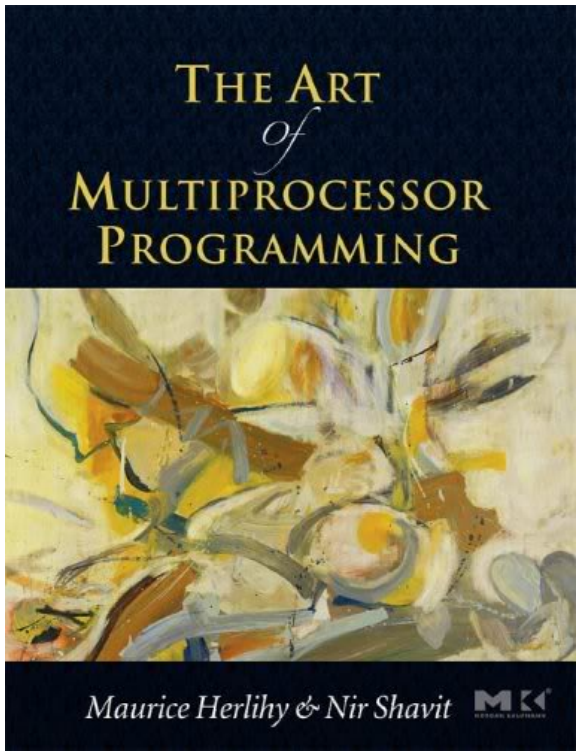
- Lectures II
  - hazardous concurrency bugs: ABA Problem, race-conditions
  - progress guarantees, linearizability
  - validation and verification of multi-processor algorithms
  - scheduling and work distribution
  - real-time systems, HPC applications, advanced simulations
  - programming language
    support for concurrency: new languages and language standards
  - the application of static and dynamic program analysis
  - new programming models for multi-core computing

TCPP Topics and Integration Plan (cont'd)

Spring 2011

The Art of Multiprocessor Programming

Nir Shavir and Maurice Herlihy
Morgan Kaufmann 2008, ISBN 978012370591.



TCPP Topics and Integration Plan (cont'd)

COP 4520:

rapidly expanding set of important topics in parallel computing and our desire to provide to our students a dynamic curriculum

After Spring 2011

- introduce multiprocessor programming earlier in the curriculum create a sophomore Parallel Programming Course in C++ class
- offer a sequel elective junior class in Parallel Graph Algorithms and Design Patterns
- offer a class on Parallel Computer Organization and Architectures
- split COP 4520 into two advanced classes:
  - a class on multiprocessor synchronization and lock-free programming
  - a course on the more traditional distributed computing and MPI programming models

Evaluation Plan

- Serve two purposes
  - Collect meaningful feedback about the current state-of-the art parallel programming technique
  - Estimate the level of preparedness of our students for applying their skills to modern industrial projects
- Main method:
  a carefully crafted survey with multiple choice and short-answer questions sent to graduates 1-3 years after graduation
- Establish the relevance of the current set of topics in our curriculum

Thank You !