SIGCSE Workshop on Modernizing Early CS Courses with Parallel and Distributed Computing

Saturday, March 2, 3pm – 6pm

Sushil Prasad
NSF & Georgia State University

Alan Sussman
NSF & University of Maryland

Sheikh Ghafoor
Tennessee Tech University

Charles Weems
University of Massachusetts
Agenda

3:00p – 3:30p  Overview of TCPP Curriculum

3:30p – 4:00p  Unplugged Activity for Introducing PDC Concepts in CS1/CS2
   - Finding Youngest Person in Class
   - M & M Sorting
   - Card Sorting

4:00p – 4:20p  Introducing OpenMP in a CS2/DS or Systems Class (slides)

4:20p – 4:35p  Break

4:35p – 5:45p  Hands on Activity on Implementing Parallel Programs in CS1/CS2
   - Parallel Sum
   - Parallel Min/Max
   - Parallel Sort

5:45p – 6:00p
   - Resources for Introducing PDC in early CS classes
   - workshop survey
IEEE-TCPP Parallel and Distributed Computing Curriculum for Computer Science and Engineering Undergraduates

Sushil K Prasad

National Science Foundation
Georgia State University
Former Chair, IEEE Technical Committee on Parallel Processing (TCPP)

SIGCSE Workshop-19, Minneapolis

TCPP Curriculum Initiative:
http://www.cs.gsu.edu/~tcpp/curriculum/
Outline

• IEEE-TCPP Curriculum
  – Why this curriculum initiative and what are the opportunities for the audience?
  – Key Activities and Milestones
    • ACM/IEEE 2013 CS Curriculum Taskforce
      – provided direct link to us for rigorous coverage
  – How was the curriculum formulated?
  – How is it getting evaluated?
  – Current Activities
  – Resources
Who are we?

• Chtchelkanova, Almadena - NSF
• Dehne, Frank - University of Carleton, Canada
• Gouda, Mohamed - University of Texas, Austin, NSF
• Gupta, Anshul - IBM T.J. Watson Research Center
• JaJa, Joseph - University of Maryland
• Kant, Krishna – George Mason University
• La Salle, Anita - NSF
• LeBlanc, Richard, Seattle University
• Lumsdaine, Andrew - Indiana University
• Padua, David - University of Illinois at Urbana-Champaign
• Parashar, Manish- Rutgers
• Prasad, Sushil- Georgia State University
• Prasanna, Viktor- University of Southern California
• Robert, Yves- INRIA, France
• Rosenberg, Arnold- Northeastern
• Sahni, Sartaj- University of Florida
• Shirazi, Behrooz- Washington State University
• Sussman, Alan - University of Maryland
• Weems, Chip, University of Massachusetts
• Wu, Jie - Temple University
Why now?

• Computing Landscape has changed
  – Mass marketing of multi-cores
  – General purpose GPUs even in laptops (and handhelds)

• A student with even a Bachelors in Computer Science (CS) or Computer Engineering (CE) must acquire skill sets to develop parallel software
  – No longer instruction in parallel and distributed computing primarily for research or high-end specialized computing
  – Industry is filling the curriculum gap with their preferred hardware/software platforms and “training” curriculums as alternatives with an eye toward mass market.
How was the curriculum formulated?

Why would they come?

*Field of Dreams (1989)*: "If you build it, he will come"
Curriculum Planning Workshops at DC (Feb-10) and at Atlanta (April-10)

• Goals
  – setup mechanism and processes which would provide periodic curricular guidelines
  – employ the mechanism to develop sample curriculums

• Agenda:
  – Review and Scope
  – Formulate Mechanism and Processes
  – Preliminary Curriculum Planning
    • Core Curriculum
    • Introductory and advanced courses
  – Impact Assessment and Evaluation Plan

Main Outcomes

- Priority: Core curriculum revision at undergraduate level

- Preliminary Core Curriculum Topics

- Sample Intro and Advanced Course Curriculums
Some Participants at the Planning Workshop, Washington DC, Feb 5-6, 2010
Weekly Tele-Meetings on Core Curriculum (May-Dec’10; Aug’11-Feb’12)

**Goal:** Propose core curriculum for CS/CS graduates

- *Every individual* CS/CE undergraduate must be at the proposed level of knowledge as a result of their *required* coursework

**Process:** For each topic and subtopic

1. Assign **Bloom’s classification**
   - K = Know the term (basic literacy)
   - C = Comprehend so as to paraphrase/illustrate
   - A = Apply it in some way (requires operational command)

2. Write **learning outcomes**
3. Identify core CS/CE courses impacted
4. Assign number of hours
5. Write suggestions for “how to teach”
## TCPP Curriculum Example

<table>
<thead>
<tr>
<th>Algorithms Topics</th>
<th>Bloom#</th>
<th>Course</th>
<th>Learning Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic problems</td>
<td></td>
<td></td>
<td><strong>The important thing here is to emphasize the parallel/distributed aspects of the topic</strong></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>broadcast</td>
<td>C/A</td>
<td>Data Struc/Algo</td>
<td>Represents method of exchanging information - one-to-all broadcast (by recursive doubling)</td>
</tr>
<tr>
<td>multicast</td>
<td>K/C</td>
<td>Data Struc/Algo</td>
<td>Illustrate macro-communications on rings, 2D-grids and trees</td>
</tr>
<tr>
<td>scatter/gather</td>
<td>C/A</td>
<td>Data Structures/Algos</td>
<td></td>
</tr>
<tr>
<td>gossip</td>
<td>N</td>
<td>Not in core</td>
<td></td>
</tr>
<tr>
<td>Asynchrony</td>
<td>K</td>
<td>CS2</td>
<td>Asynchrony as exhibited on a distributed platform, existence of race conditions</td>
</tr>
<tr>
<td>Synchronization</td>
<td>K</td>
<td>CS2, Data Struc/Algo</td>
<td>Aware of methods of controlling race condition,</td>
</tr>
<tr>
<td>Sorting</td>
<td>C</td>
<td>CS2, Data Struc/Algo</td>
<td>Parallel merge sort,</td>
</tr>
<tr>
<td>Selection</td>
<td>K</td>
<td>CS2, Data Struc/Algo</td>
<td>Min/max, know that selection can be accomplished by sorting</td>
</tr>
</tbody>
</table>
How is the Curriculum being evaluated?

Early Adopter Program
EduPar/EduHPC/Euro-EduPar Workshop series
Early Adopter Program

• Over 100 institutions worldwide
  – Spring-11: 16 institutions; Fall’11: 18;
  – Spring-12: 21; Fall-12: 25 institutions, Fall-13: 25 institutions, Fall-14: 25, Fall-15: 13
  – Most from US (4 year to research institutions, one high school)
  – Some from South America, a few from Europe, fewer from Asia (India, China, Indonesia, Singapore), Middle East

• Next competition: Spring/Summer 2019
  – NSF/Intel funded Cash Award/Stipend up to $1500-5000/proposal
  – Which course(s), topics, evaluation plan?

• Instructors for core CS/CS courses such as CS1/2, Systems, Data Structures and Algorithms – department-wide multi-course multi-semester adoption preferred
  – Elective courses; graduate courses
Early Adopter Status Awarded - Fall 2013

1. DePauw University
2. Tennessee Technological University
3. NCSU
4. University of Central Florida
5. Washington State University
6. Texas State University
7. Michigan Technological University
8. University of Houston - Clear Lake
9. Northeastern Univ.
10. Portland State University
11. Jackson State University
12. Ursinus College and Haverford College
13. Miami University
14. University of Utah
15. Louisiana School for Math, Science, and the Arts – High School

1. Indian Institute of Technology, Kharagpur
2. Indian Institute of Technology, Patna
3. Russian Federation: Nizhni Novgorod State University
4. Hong Kong: City University of Hong Kong
5. India; Walchand College Of Engineering, Sangli
6. Pakistan: FAST- National University of Computer and Emerging Sciences
7. Germany: TU Braunschweig
8. China: University of Electronic Science and Technology of China
9. Israel: Bar-Ilan University
10. Egypt: AASTMT
Courses updated by Early Adopters - Fall 2012

- **Swarthmore College**
  - CS31 Introduction to computer systems, CS40 Computer graphics, CS41 algorithms, CS45 Operating Systems, and CS87 Parallel computing

- **Oklahoma City University**
  - CS1, CS2, CS3, Software Engineering

- **Singapore University** of Technology and Design
  - Introduction to Algorithms - departmental core; Computer System Engineering - departmental core; and Graph Theory and Algorithm - elective.

- **Purdue University** - ECE 264 - Advanced C Programming

- **Huazhong University of Science and Technology**
  - Parallel Programming Principle and Practice; Parallel Data Structure and Algorithm

- **Wilberforce University** - IDS L, university wide core, and a co-op course.

- **Carnegie Mellon University** & Pittsburgh Supercomputing Center
  - Introduction to Computational Physics and Advance Computational Physics

- **Louisiana State University** - Digital Logic I, II
Edu* Workshop Series

- **EduPar-11** at Alaska, IPDPS-2011
  - Receive feedback from the Adopters
  - Stimulate discussion of curricular and other educational issues.
- **EduPar-12** at Shanghai, IPDPS-2012
  - A regular satellite workshop of IPDPS
- **EduPar-13** in Boston + **EduHPC** Workshop at SC-13 + BOF at SIGCSE-14
- **EduHPC-14** @ SC-14, Nov – New Orleans; EduHPC-15 in Austin, EduHPC-16, EduHPC-17, EduHPC-18 in Dallas
- **EduPar-15** @IPDPS, May, India; EduPar-16, Chicago, EduPar-17 in Orlando; EduPar-18 in Vancouver
- **EduHiPC 2018 @ HiPC in Bangalore** – for India and the region
  - Monday, Dec 2018
- **EduPar-19** @ IPDPS in Rio in May’19
  - Still accepting for Peachy Assignments
CDER: Center for Parallel and Distributed Computing Curriculum Development and Educational Resources

• Develop **PDC core curricula** flexible enough for a broad range of programs and institutions; collaborate with all stakeholders
  – Version I released Dec 2012; version 2 expected Spring/Summer 2018
• Develop, collect, and synthesize pedagogical and instructional materials for teaching PDC curriculum topics*
  – Website setup and populated mostly from early adopters
  – Book project Volume I published; Volume 2 upcoming
• Facilitate access to state-of-the-art **hardware and software resources** for PDC instruction and training by instructors and students*
  – CDER cluster - Multi-core, GPU, Shared/Distributed Memory
    • request accounts for Spring-19 and Summer-19
• Organize Early Adopter Competitions and EduPar workshops, and related events
CDER Courseware Website

Upload and Search Course Material

• **Type:**
  – Slides, Syllabus, Tutorial, Video
  – Animation, Article, Award, Blog, Book, Competition
  – Course Template, Course Module, Data
  – Hardware Access, Software/Tools
  – Proposal, Report

• **Courses:**
  – CS1, CS2, Systems, Data Structures and Algorithms, ...

• **NSF/TCPP Topic/Subtopic Classification:**

  **ALGORITHMS**
  Parallel and Distributed Models and Complexity
  Algorithmic Paradigms
  Divide & conquer (parallel aspects)
  Algorithmic problems

  **ARCHITECTURE**
  **PROGRAMMING**
  **CROSS-CUTTING**

- **Now open** - Work in Progress

Prasad/SIGCSE-19
CDER Book Project

• Lack of suitable textbooks to integrate PDC topics into the core courses
  – CS1, CS2, Systems, and Data Structures and Algorithms

• Part I - For instructors: Basic Concepts and References on what and how to teach

• Part 2: For students: Supplemental teaching material for core courses

• 9 chapters
  – over 27K chapter downloads – free downloads

• 2nd Volume – Published Nov’19
  – Vol 3 – Early Adopter course and topic exemplars and accompanying resources

Prasad/SIGCSE-19
CHAPTER 1 Editors’ Introduction and Road Map

PART 1 FOR INSTRUCTORS

- CHAPTER 2 Hands-on Parallelism with no Prerequisites and Little Time
- CHAPTER 3 Parallelism in Python for Novices
- CHAPTER 4 Modules for Introducing Threads
- CHAPTER 5 Introducing Parallel and Distributed Computing Concepts in Digital Logic
- CHAPTER 6 Networks and MPI for Cluster Computing

PART 2 FOR STUDENTS

- CHAPTER 7 Fork-join Parallelism with a Data-Structures Focus
- CHAPTER 8 Shared-Memory Concurrency Control with a Data-Structures Focus
- CHAPTER 9 Parallel Computing in a Python-Based Computer Science Course
- CHAPTER 10 Parallel Programming Illustrated through Conway’s Game of Life

Free Preprint version of this CDER book:
http://grid.cs.gsu.edu/~tcpp-curriculum/?q=cedr_book
## Curriculum Version II Activities

<table>
<thead>
<tr>
<th></th>
<th>Areas</th>
<th>Architecture</th>
<th>Algorithms</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New Aspects</strong></td>
<td><strong>Area Lead/Aspect Lead</strong></td>
<td>Chip Weems</td>
<td>Arnold Rosenberg</td>
<td>Alan Sussman</td>
</tr>
<tr>
<td><strong>Exemplars</strong></td>
<td><strong>Sushil Prasad</strong></td>
<td>Karen Karavanic, Eric Freudenthal</td>
<td>Erik Saule, Duane Merril, David Bunde</td>
<td>David Brown, Eric Freudenthal</td>
</tr>
<tr>
<td><strong>Distributed</strong></td>
<td>Vaidyanathan Ramachandran</td>
<td>Vaidyanathan Ramachandran, Manish Parashar</td>
<td>Vaidyanathan Ramachandran, Costas Busch, Denis Trystram</td>
<td>Alan Sussman, Chi Shen</td>
</tr>
<tr>
<td><strong>Big Data</strong></td>
<td>Trilce Estrada</td>
<td>Craig Stunkel</td>
<td>Cynthia Phillips, Debzani Deb</td>
<td></td>
</tr>
<tr>
<td><strong>Energy</strong></td>
<td>Krishna Kant, Craig Stunkel</td>
<td>Craig Stunkel, Karen Karavanic</td>
<td>Denis Trystram</td>
<td>John Dougherty</td>
</tr>
<tr>
<td><strong>Crosscutting</strong></td>
<td>Sheikh Ghafoor</td>
<td>Craig Stunkel, Eric Freudenthal</td>
<td>Robert Robey, Martina Barnas</td>
<td>Sheikh Ghafoor, Eric Freudenthal</td>
</tr>
</tbody>
</table>
Programming Topics - Overview

• Main goal is to introduce parallel programming topics into intro programming, data structures, and systems classes
  – Secondary goal is to target upper-level classes
• High-level themes include:
  – **Paradigms and notations** – SIMD, shared memory, message passing, client/server, big data stack, threads, tasks, data parallel, etc.
  – **Semantics and correctness** – synchronization, concurrency defects, ...
  – **Memory models** – sequential consistency, weak consistency, ...
  – **Performance and energy** – computation and data decomposition, scheduling/mapping, data layout and locality, tools and metrics
• Most topics at a shallow level (Bloom level C or K) for intro courses, but at a deeper level for upper-level courses (or deferred to upper-level completely, so at N Bloom level for intro courses)
Programming Topics - Updates

• Incorporated new programming topics related to distributed computing (e.g. client/server), big data (e.g., MapReduce), and power/energy

• Eliminated some topics from original guidelines completely, since no longer relevant

• Added small number of other topics missed in original guidelines, or newer ideas (e.g., accelerator programming)
Architecture

• Scaling and efficiency (e.g., big data, energy)
• Power and energy
• Hardware/Software threads
• Atomicity, ordering, consistency, weak forms
  – Extend to distributed issues
• Accelerators as more general paradigm
Algorithms

- **Goal:** Encourage parallel and distributed thinking early on

- **Suggested Methodology:** Gently alter existing courses to incorporate parallel thinking

- **Introduce ideas in the tradition setting so they can be elaborated in advanced/elective courses**
  
  - **Big Idea 1:** *Parallel actors are inherently asynchronous.* Asynchrony may be useful, or may need to be tamed.
  
  - **Big Idea 2:** Recursive *divide-and-conquer* can be mapped to parallelism.
Three main topic areas:

- **Parallel and Distributed Models and Complexity**
  - basis for analyzing parallel algorithms and their performance

- **Algorithmic Techniques**
  - algorithmic kernels that can be used as building blocks for parallel algorithms

- **Algorithmic Problems**
  - parallel algorithms for solving common problems such as sorting, searching, multi-agent communication
Sponsorship Acknowledgements

– NSF
  • NSF/TCPP Curriculum Initiative
  • Early adopter competitions (stipend, travel)
  • EduPar/EduHPC workshop series
  • CRI-ADDO CDER (2012-15)

– Intel (2011- )
  • International early adopter institutions (stipend, travel)
  • Curriculum Development, Courseware, Exemplars

– nVIDIA
  • GPU cards to 50+ early adopters from Spring'11, Fall'11 and Spring'12 rounds.

– IEEE TCPP, IBM
  • Keynotes in the past
Conclusion

• Need to inculcate “parallel thinking” to all

• Core Curriculum Revision is a community effort
  – Curriculum Initiative Website:

• EduPar-19 @ IPDPS in Rio, Brazil
  • PC Chair: Noemi Rodriguez
  • Peachy Assignments

• CDER Resources
  – Book Project - download free preprints
  – MPI/Spark Cluster - free access for courses
  – Courseware