Modernizing Early Computing Courses with Parallel and Distributed Computing

Sushil K Prasad, UT San Antonio
Alan Sussman, U. Maryland
Chip Weems & Neena Thota, UMass
R. Vaidyanathan, LSU
David Bunde & Jaime Spacco, Knox
Sheikh Ghafoor, April Crockett, & Jerry Gannod, TTU

SIGCSE-25, Feb 27-29, Pittsburgh
https://tcpp.cs.gsu.edu/curriculum/

Public Feedback on TCPP Curriculum & Contact
sushil.prasad@gmail.com

Sponsors

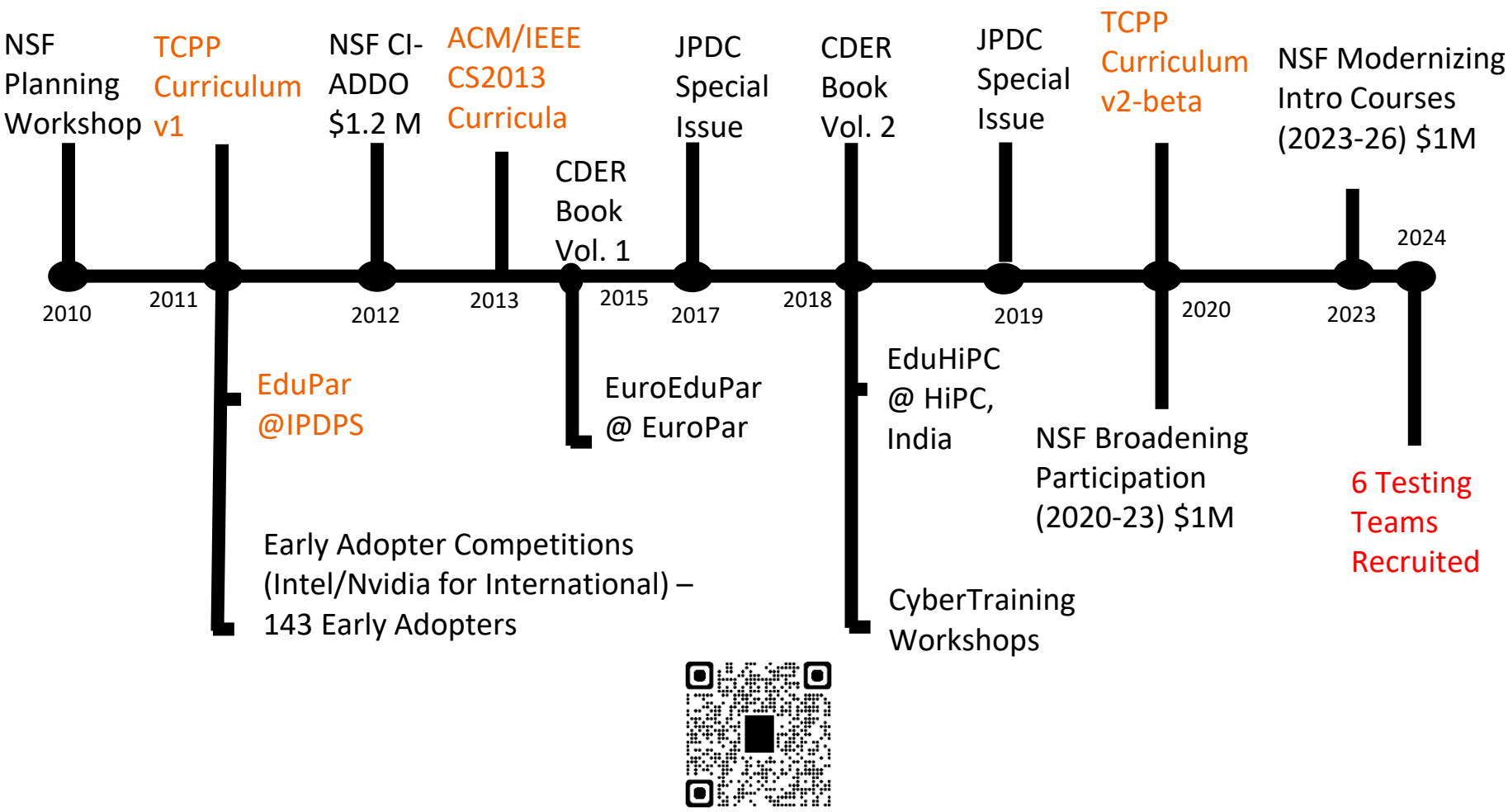
TCPP Curriculum Initiative

What should every Computer Science and Engineering Student know about Parallel and Distributed Computing (PDC)?
https://tcpp.cs.gsu.edu/curriculum/

- **Areas:** Programming, Architecture, & Algorithms
  - Version 1 – 2012
- **New Aspects:** Big Data, Energy, Distributed Computing, Pervasive topics
  - Version-2-beta released 2020

- **Companion Activities:**
  - CE-oriented TCPP Curriculum
  - Modernizing CS1/CS2 Exemplars
    - 2 Development Teams
    - Recruited 6 Testing teams
  - CDER Book Vol 3
    - Experience of Adopters
      - Exemplars + Resources
  - JPDC Special Issues

CDER - Center for Parallel and Distributed Computing Curriculum Development and Education Resources - Timeline



3 Curriculum Areas + Cross-Cutting
Architecture, Programming, Algorithms

TCPP Curriculum Example

Algorithms Topics	Bloom#	Course	Learning outcome and teaching notes
Algorithmic problems			Algorithmic problems section contains parallel algorithms for certain problems. The important thing here is to emphasize the parallel/distributed aspects of the topic
Communication and Synchronization			Understand (at the pseudo-code level) how certain patterns of communication can be implemented in a parallel/distributed model. Also appreciate the cost of communication in PDC.
Reduction and Broadcast for communication and synchronization	**C**	Data Struc/Algo	Understand, for example, how recursive doubling can be used to for all-to-one reduction, and its dual, one-to-all reduction, in log(p) steps. The same applies to all-to-all broadcast and all-to-all reduction. Recognize that all-to-all broadcast/reduction are synchronizing operations in a distributed (event-driven) environment.
Parallel Prefix (Scan)	**C**	Data Struc/Algo	Understand the structure of at least one simple parallel prefix algorithm. One could consider recursive or iterative approaches (such as those of Ladner-Fischer, Kogge-Stone, Brent-Kung)
Multicast	**N**		
Permutation	**N**		

Early Adopter and Training Programs

Over 200 early adopter and trainee institutions worldwide

- Spring-11 - Fall-15
- US, South America, Europe, Asia, and Middle East

NSF CyberTraining PDC Workshops - Summer 2018-25

- UMass; Tennessee Tech; LSU

Additional Training workshops

- **SIGCSE** 2023, 2024, **2025**
- **HiPC** 2022, 2023, **2024**

SIGCSE Tutorial

Modernizing the CS Introductory Sequence with Parallel and Distributed Computing (and some AI)

NSF Stipend: $400/trainee

Fri Feb 28, 2025
7:00 pm - 10:00 pm
at
Meeting Room 310

Some slots available

Register now!

Active, hands-on & unplugged activities

Integrating PDC topics in our courses CS1 & CS2

GenAI in CS1 & CS2

Call for Applications

NSF/CDER Instructor Training Workshop

MODERNIZING THE EARLY CS COURSES WITH PARALLEL AND DISTRIBUTED COMPUTING

July 14 - 18, 2025 hosted by UMass, Amherst

Application Deadline
March 31, 2025

$3000
Stipend/trainee

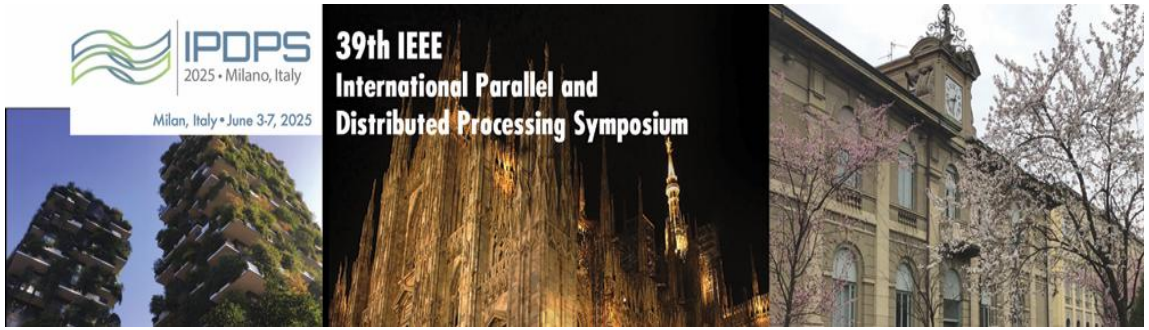
Apply now!

Workshop Agenda:

- PDC Unplugged & Plugged Activities
- Educational Evaluation Methodology
- Previews of modern exemplar first year courses
- Integrating PDC Topics in CS1, CS2, Data Structures and Algorithms, Systems/Comp Org

Edu* Workshop Series

- **EduPar-11** at IPDPS-2011
  - EduPar25 – June, Milan, Italy

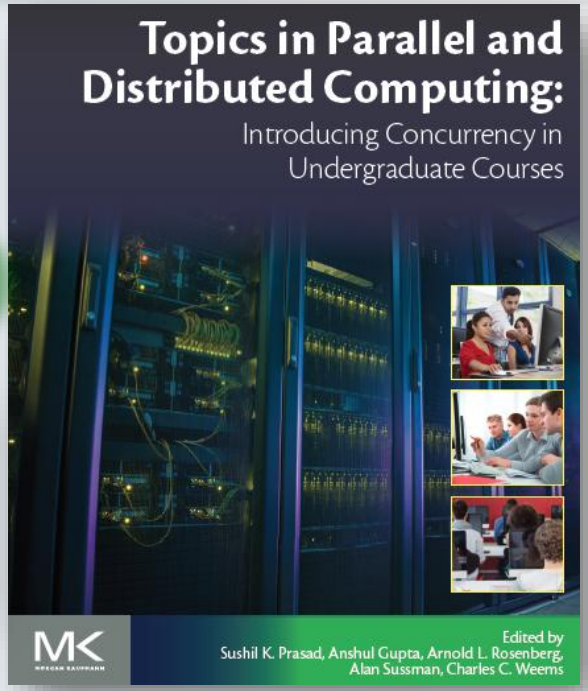


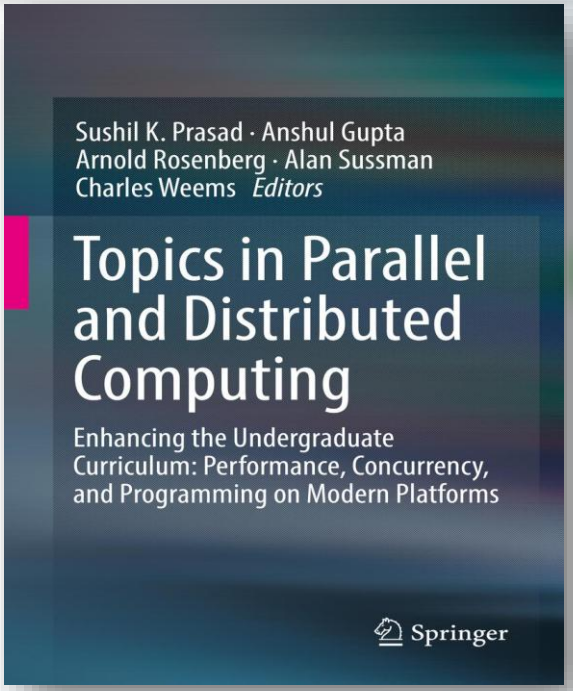
- **EduHPC** at SC-13
  - EduHPC-25 @ SC, St Louis, MO
- **EduHiPC 2018 @ HiPC in** India
  - EduHiPC'25 in Hyderabad, India

CDER Books - Topics in Parallel and Distributed Computing for Undergraduates

50K+ Chapter Downloads

Free Download





Free Download

PART 1 FOR INSTRUCTORS
5 Chapters

- ✓ Hands-on Parallelism with no Prerequisites and Little Time Using Scratch
- ✓ Parallelism in Python for Novices
- ✓ Modules for Introducing Threads
- ✓ Introducing Parallel and Distributed Computing Concepts in Digital Logic
- ✓ Networks and MPI for Cluster Computing

PART 2 FOR STUDENTS
4 Chapters

- ✓ Fork-join Parallelism with a Data-Structures Focus
- ✓ Shared-Memory Concurrency Control with a Data-Structures Focus
- ✓ Parallel Computing in a Python-Based Computer Science Course
- ✓ Parallel Programming Illustrated through Conway's Game of Life

PART 1 FOR INSTRUCTORS
5 Chapters

- ✓ What do we need to know about parallel algorithms and their efficient implementation?
- ✓ Models for Teaching Parallel Performance Concepts
- ✓ Scalability in Parallel Processing
- ✓ Energy Efficiency Issues in Computing Systems
- ✓ Scheduling for fault-tolerance: an introduction

PART 2 FOR INSTRUCTORS
5 Chapters

- ✓ MapReduce - The Scalable Distributed Data Processing Solution
- ✓ The Realm of Graphics Processing Unit (GPU) Computation
- ✓ Managing Concurrency in Mobile User Interfaces with Examples in Android
- ✓ Parallel Programming for Integrative GUI Applications

- CDER Book Vol 3 - upcoming

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

- open - Work in Progress

TCPP Computer Engineering Curriculum

- PDC Concepts
  - Concurrency, Asynchrony and Locality
  - Decomposition and Coordination
  - Performance and Pitfalls
- CE Courses identified
- Broad areas
  - Hardware and Architecture
  - Programming and Algorithms
  - Communications and Systems

Identifying topics in lower-level courses
in which PDC concepts can be incorporated

NSF Modernizing CS1/CS2 Exemplar Project

Vision: Create modern course exemplars for CS1 and CS2 courses to serve as national models	• Systems are more networked, parallel, and graphical than traditional courses • Software uses more libraries and APIs
2 Development teams	• Knox College (Java version) and Tennessee Tech (C/C++ version) • Piloting CS1 this term, CS2 in the Spring
6 Testing Teams	• Casper, Hawaii Pacific U., Montclair, U. Nebraska Lincoln, U. Southern Indiana, and Webster • Will teach the new courses in 2025-26