

# EduHiPC-22 Training Workshop

**Welcome**

**+**

## **IEEE-TCPP Parallel and Distributed Computing Curriculum Initiative**

Sushil K Prasad, UT San Antonio

Sheikh Ghafoor, Tennessee Tech

Ashish Kuvelkar, CDAC

**Dec 17, 2022**

**TCPP Curriculum Initiative:**

<https://tcpp.cs.gsu.edu/curriculum/>

# Welcome

- Our Goal
  - To improve PDC awareness, knowledge, and interest in undergraduate (CS/CE/Engineering) students by exposing them to PDC concepts and issues in their regular course of study (workforce development)
- *Objective of this workshop (Training of the Trainers)*
  - Provide hands-on training on shared memory parallel programming using OpenMP
  - Provide ideas and resources on how to integrate parallel and distributed computing (PDC) concepts in undergraduate CS and CE courses
  - Provide future support
    - Cohort group
    - Organizers and sponsors

# Big Thanks to Training Workshop Organizers!

- *Organizers:*
  - Ashish Kuvelkar, CDAC, India
  - Sheikh Ghafoor, Tennessee Tech
  - Preeti Malakar, IIT Kanpur, India
  - Sushil Prasad, UT San Antonio
- **Additional Instructors:**
  - Vamshi Krishna, Vardhaman Parakh
- **HiPC Organizers:** Chiranjib Sur, Vivek Yadav, Sanmukh Kuppannagari, Ananth Kalyanaraman, Krishna Nandivada
- EduHiPC Workshop
  - Tomorrow 9am-4:30pm
- Next Edu\* Workshop
  - EduPar'23, May 15, St. Petersburg, Florida
- Sponsors:
  - CDAC - Primary
  - IEEE TCPP
  - CDER Center
  - Tennessee Tech
  - UT San Antonio

# Agenda

- 9:00 – 9:30 Welcome & Introduction (Workshop goal, CDER activities, Curriculum) – Sushil Prasad
- 9:30 – 10:30 Pervasive PDC Concepts and unplugged activities for early CS classes - Sheikh Ghafoor
- 10:30 – 11:00 *Break*
- 11:00 – 11:30 Intro to cluster and system check and login to cluster – Ashish Kuvelkar
- 11:30 – 12:30 Core OpenMP - Sheikh Ghafoor
- 12:30-1:30 *Lunch*
- 1:30 – 4:30 Hands on exercise (*with a break at 3:30 PM*)  
- Sheikh, Vamshi, Vardhaman, Ashish, Sushil
- 4:30 – 5:00 Discussion on integration strategies in CS1 and CS2  
- Sheikh/Sushil/Ashish
- 5:00 -5:30 Beyond core OpenMP and other technologies, NSM, survey and feedback  
- Sheikh, Ashish, Sushil

# NSF/IEEE-TCPP Curriculum Initiative

What should every Computer Science and Engineering Student know about Parallel and Distributed Computing (PDC)?

<https://tcpp.cs.gsu.edu/curriculum/>

# Some Participants at the NSF Planning Workshop, Washington DC, Feb 5-6, 2010



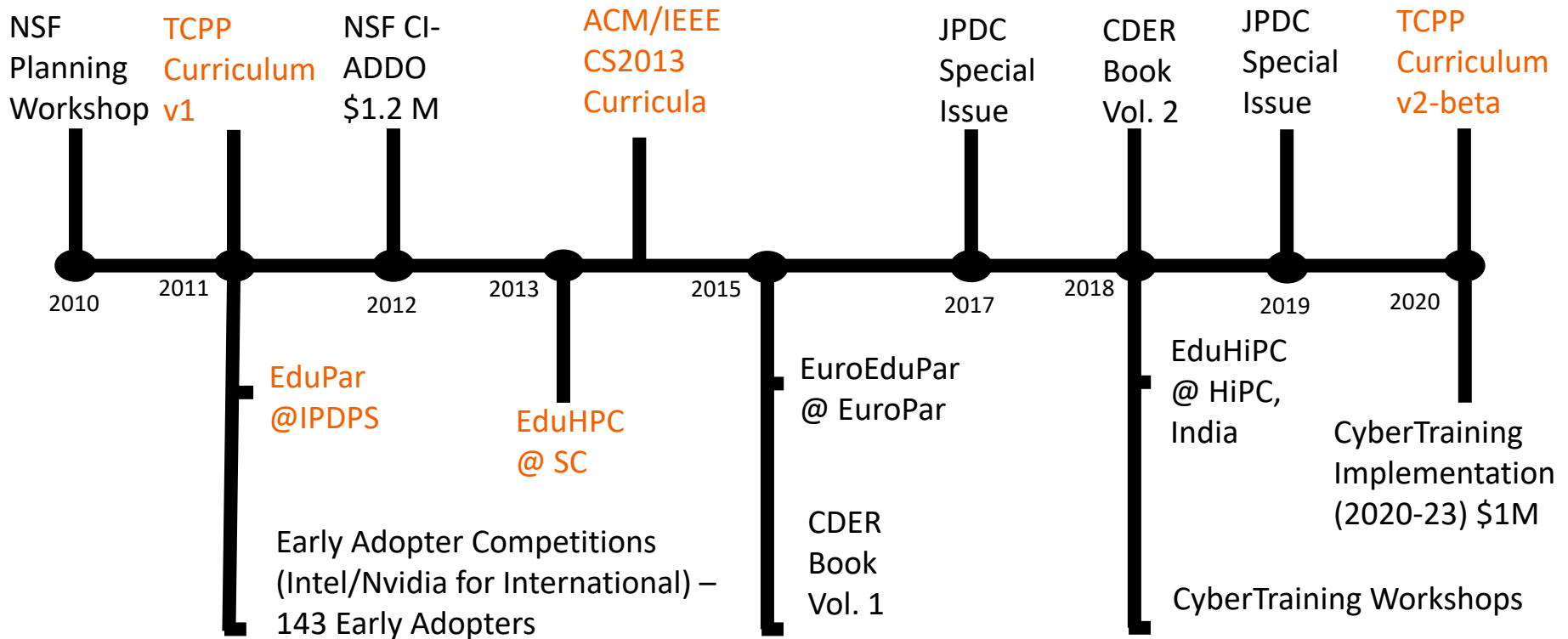
## Main Outcomes

- Priority:  
Core curriculum  
revision at  
undergraduate level

- Preliminary Core  
Curriculum Topics

- Sample Intro and  
Advanced Course  
Curriculums

# CDER Timeline



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

# TCCP Curriculum Example

Algorithms Topics		Bloom#	Course	Learning outcome and teaching notes
<b>Algorithmic problems</b>				Algorithmic problems section contains parallel algorithms for certain problems. The important thing here is to emphasize the parallel/distributed aspects of the topic
<i>Communication and Synchronization</i>				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	<b>C</b>	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)	<b>C</b>	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	<b>N</b>		
	Permutation	<b>N</b>		



# Early Adopter and Training Programs

- Over 140 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
- **NSF CyberTraining PDC** Workshops - Summer 2018-23
  - **LSU**; UMass/Maryland; Tennessee Tech
  - NSF/Intel funded stipend up to \$5K/proposal
  - *Instructor training + adoption plans*

# 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-15 @IPDPS, May, India; EduPar-16, Chicago, EduPar-17 in Orlando; EduPar-18 in Vancouver, EduPar-19 @ IPDPS, Brazil, EduPar'20, EduPar21 – online, EduPar22 - online – May 30, **EduPar23 – May 15, FL**
- **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, EduHPC-19 @ SC in Denver
  - EduHPC-20 @ SC - online, EduHPC-21 @ SC – hybrid, **EduHPC-22 @ SC**
- **Euro-EduPar Aug 2015**; Euro-EduPar-2016, EEP-2017, EEP-18, **EEP-22**
- **EduHiPC 2018 @ HiPC in Bangalore** – for India and the region
  - EduHiPC'19 @ HiPC in Hyderabad, EduHiPC'21 @ HiPC in Bangalore – online
  - **EduHiPC'22 in Bangalore**

# NSF/TCPP Curriculum Initiative – Additional Resources

- CDER Book series:
  - Vol 1: Topics in Parallel and Distributed Computing
    - Introducing Concurrency in Undergraduate Courses, *Morgan Kaufman*
  - Vol 2: Topics in Parallel and Distributed Computing
    - Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms, *Springer*
    - **Free Pre-Print Version** on CDER site (**49K downloads**)
    - **Plan for 3<sup>rd</sup> Volume** – Experience of Adopters
      - Exemplars + Resources on courses and topics
- CDER Heterogenous Cluster
  - Multi-core, GPU, Shared/Distributed Memory, **Hadoop/Spark**
  - **Ask for class accounts**
- **JPDC Special Issue** - Keeping up with Technology: Teaching Parallel, Distributed and High-Performance Computing (2017, 19, 21) – **upcoming CFP**

# 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

# Curriculum Version II Activities

	Areas	Architecture	Algorithms	Programming
<b>New Aspects</b>	<b>Area Lead/ Aspect Lead</b>	Chip Weems	Anshul Gupta	Alan Sussman
<b>Exemplars</b>	Sushil Prasad	Karen Karavanic, Eric Freudenthal	Erik Saule, Duane Merrill, David Bunde	David Brown, Eric Freudenthal
<b>Distributed</b>	Vaidyanathan Ramachandran	Vaidyanathan Ramachandran, Manish Parashar	Vaidyanathan Ramachandran, Costas Busch, Denis Trystram	Alan Sussman, Chi Shen
<b>Big Data</b>	Trilce Estrada	Craig Stunkel	Cynthia Phillips,	Debzani Deb
<b>Energy</b>	Krishna Kant, Craig Stunkel	Craig Stunkel, Karen Karavanic	Denis Trystram	John Dougherty
<b>Pervasive</b>	<b>Sheikh Ghafoor</b>	Craig Stunkel, Eric Freudenthal	Robert Robey, Martina Barnas	Sheikh Gafoor, Eric Freudenthal

- **Timeline:**
  - **Version-2-beta released @ EduHPC'20**
    - **Public Feedback: [sushil.prasad@utsa.edu](mailto:sushil.prasad@utsa.edu)**
  - **Companion Activities:**
    - CE-oriented TCPP Curriculum
    - ABET's exposure to PDC
      - **Minimal set of topics**
      - Interfacing with evaluators, CSAB
- NSF Institute Planning Grant => 5 planning workshops
  1. SC'19
  2. SIGCSE'20 - online
  3. July 27, 2020 – online
  4. Mar 26-27, 2021 - online
  5. **NSF Report Workshop – Oct'21**

# Architecture Topics

Chip Weems  
Univ of Massachusetts

# Overview

- Limited coverage in core courses
  - Only K or C Bloom levels
  - Most in Systems course (a few topics in CS1, CS2, such as floating point, atomicity as basis for threaded libraries, event handlers as threads in GUIs)
  - **Goal** is for all students to be aware of hardware aspects of PDC affecting and providing opportunities for algorithmic problem solving
- Most topics in table are for advanced courses
  - Many already normally covered there, at or beyond suggested bloom levels



# Topics Overview

- **Classes of Parallelism**: data (superscalar, SIMD, dataflow), pipelines (OoO, streams), control (multicore, multithread, heterogeneity), shared memory (snooping, directory), distributed memory (topology, latency)
- Underlying mechanisms (caching, atomicity, consistency, coherence, interrupts/events, handshaking, ID, virtualization)
- Floating point representation (in support of HPC)
- Performance metrics (IPC, benchmarks, network/memory bandwidth, peak performance, sustained performance)
- **Power** (power/energy, larger scale, embedded, density, static/dynamic, DVFS, heterogeneous cores)
- **Scaling** (**Big data**, HPC, fault tolerance, data bound computation, volume, velocity, scale out, cost of data movement)

# Pervasive and Emerging Topics

Sheikh Ghafoor, Tennessee Tech

R. Vaidyanathan, LSU

# Pervasive Concepts

- Broad themes that are conceptual underpinning of PDC
- Appears at different depth throughout the curriculum
- Transcend PDC and often computing
- Should be taught at the core
  - May be through unplugged activities
  - least at “C” level in terms of Bloom classification.
- These should be further reinforced in architecture, programming, and algorithms
- **Four topics have been identified**
  - **Asynchrony**
  - **Concurrency and Sequential Dependency**
  - **Locality**
  - **Performance**

# Emerging Topics

Topics	Core Bloom Level	Learning Outcome and Teaching Suggestion	Where Covered
Machine Learning	N	Understand that accelerators, parallel matrix decompositions, parallel matrix multiplication, and vectorized operations are fundamental aspects that enable modern machine learning	ParProg, Machine Learning(K)
Mobile Adhoc Network	N	Understand that mobility admits a dynamic topology and a different relationship between "sender" and "receiver." Explore the impact of these differences in the system.	Networking (c)
Quantum Computing	N	Know that quantum computing algorithms can solve some problems (e.g., optimization problems) exponentially faster than classical algorithms via parallelism, and that quantum processors to implement some of those algorithms are under construction, using new parallel programming models	Arch2, ParProg(k)

# Programming Topics

Alan Sussman

University of Maryland

# 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 advanced courses (or deferred to upper-level completely, so at Bloom level N for intro courses)

# Algorithms

A. Gupta, IBM Research

R. Vaidyanathan, LSU

# Algorithms Overview

- Algorithms topics are recommended for coverage along with their sequential counterparts to minimize additional instruction time
- Organized into three sub-areas:
  - ***Parallel/Distributed Models and Complexity:*** Foundational topics aimed at equipping the students with basic know how for designing and analyzing parallel algorithms
  - ***Algorithmic Techniques:*** Recurrent themes or constructs that are generally useful in designing a wide variety of parallel algorithms
  - ***Algorithmic Problems:*** Basic problems for which learning both the sequential and parallel algorithms would be considered valuable for almost all CS/CE students



# Q&A

**TCPP Curriculum Initiative:**

<https://tcpp.cs.gsu.edu/curriculum/>

Feedback & Participation: [sushil.prasad@utsa.edu](mailto:sushil.prasad@utsa.edu)