

NSF/IEEE-TCPP Curriculum on Parallel and Distributed Computing for Undergraduates

- Version IIβ

Big Data, Energy, and Distributed Computing

Sushil K Prasad, UT San Antonio
Sheikh Ghafoor, Tennessee Tech
Krishna Kant, Temple University

Alan Sussman, University of Maryland
R. Vaidyanathan, LSU
Charles Weems, UMass Amherst

SIGCSE'23, March 16, 2023, Room 716

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



Public Feedback on TCPP Curriculum & Contact

sushil.prasad@gmail.com



Agenda

- Introduction
- Pervasive Concepts
- Architecture Area
 - *Online Feedback Survey*
- Programming Area
 - *Online Feedback Survey*
- Algorithms Area
 - *Online Feedback Survey*
- Emerging Aspects
 - Big Data
 - Energy
 - Distributed Computing
 - *Online Feedback Survey*
- Q&A

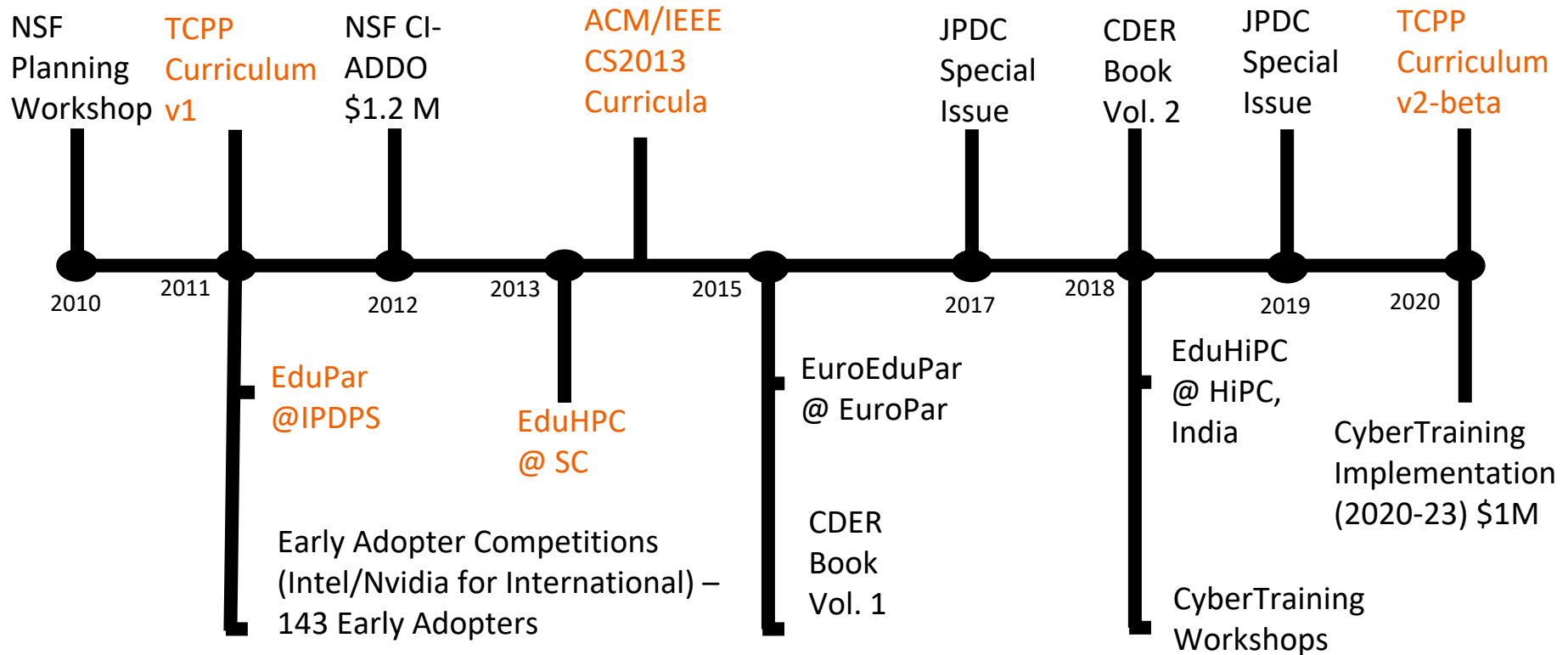
Some Participants at the NSF Planning Workshop Washington DC, Feb 5-6, 2010



Main Outcomes

- Priority:
Core curriculum
revision at
undergraduate level
- Preliminary Core
Curriculum Topics
- Sample Intro and
Advanced Course
Curriculums

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



3 Curriculum Areas + Cross-Cutting

Architecture, Programming,
Algorithms

TCCP 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
<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	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: 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 2018-22
 - **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
- **EduHiPC 2018 @ HiPC in Bangalore** – for India and the region
 - EduHiPC'19 in Hyderabad, EduHiPC'21 in Bangalore – online
 - **EduHiPC'22 in Bangalore**

Additional CDER 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 (**50K+ downloads**)
 - **Plan for 3rd 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
New Aspects	Area Lead/ Aspect Lead	Chip Weems	Anshul Gupta	Alan Sussman
Exemplars	Sushil Prasad	Karen Karavanic, Eric Freudenthal	Erik Saule, Duane Merrill, David Bunde	David Brown, Eric Freudenthal
Distributed	Vaidyanathan Ramachandran	Vaidyanathan Ramachandran, Manish Parashar	Vaidyanathan Ramachandran, Costas Busch, Denis Trystram	Alan Sussman, Chi Shen
Big Data	Trilce Estrada	Craig Stunkel	Cynthia Phillips	Debzani Deb
Energy	Krishna Kant, Craig Stunkel	Craig Stunkel, Karen Karavanic	Denis Trystram	John Dougherty
Pervasive	Sheikh Ghafoor	Craig Stunkel, Eric Freudenthal	Robert Robey, Martina Barnas	Sheikh Ghafoor, Eric Freudenthal

- **Timeline:**
 - **Version-2-beta released @ EduHPC'20**
 - Public Feedback: sushil.prasad@gmail.com
 - **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**

CDER Events at SIGCSE'23

Special Session

NSF/IEEE-TCPP Curriculum on Parallel and Distributed
Computing for Undergraduates - Version II β
Big Data, Energy, and Distributed Computing

Thu 16 Mar 2023 13:45 - 15:00 at Room 716

Curriculum Feedback Surveys

Architecture
Area



Programming
Area



Algorithms
Area



Aspects of Big Data,
Energy and Distributed



Training Workshop

Integrating Parallel
and Distributed
Computing in Early
Computing Classes

**Fri 17 Mar'23 19:00 - 22:00
at Room 715**

CDER Booth #217

Thu 16- Sat 18 Mar'23

Pervasive Concepts

Sheikh Ghafoor, Tennessee Tech

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

Pervasive Concepts

Topics	Learning Outcome and Teaching Suggestion
Asynchrony	1) Understand cause and effect of Asynchrony and how to ensure that computational correctness. 2) Understand asynchrony is the characteristics of modern system and understand asynchrony in PDC context. 3) Can utilize a standard coordination strategy to prevent incorrect operation due to uncontrolled concurrency (race conditions).
Concurrency & Dependency	1) Understand concurrency is an algorithmic property and difference between concurrency and parallelism. 2) Understand sequential dependency limit degree of concurrency and hence parallelism. 3) Identify sequential dependency in algorithms.

Architecture Topics

Chip Weems
Univ of Massachusetts

Overview

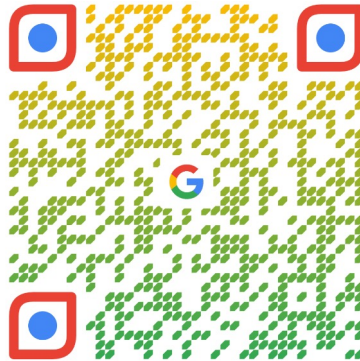
- Limited coverage in core courses
 - Only K or C Bloom levels
 - Most in Systems course (a few topics in CS1, CS2, such as floating point, atomicity as basis for threaded libraries, event handlers as threads in GUIs)
 - Goal is for all students to be aware of hardware aspects of PDC affecting and providing 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)
- Survey only covers a top-level subset

Architecture Area Survey

URL: <https://forms.gle/TnMy39dcX9C1XmFX7>



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 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)
- Revised learning outcomes, Bloom levels for some topics (mainly between C and K)
- Added learning outcomes and Bloom levels for advanced courses

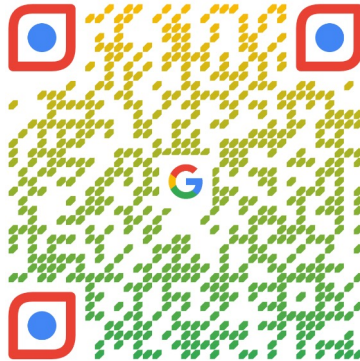
Programming Topics

– Input from Reviewers

- Several suggestions to do a better job on pervasive ideas
 - Across programming areas, algorithms, architecture – e.g., scalability
- Better definitions of terms and acronyms
- More limited energy/power topics – keep them high level
- Some suggestions for updating topics, new topics, eliminating topics no longer relevant
 - Informed discussion for revisions

Programming Area Survey

<https://forms.gle/vcaFmVBSG4k4HBKF7>



Algorithms

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 knowhow 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

Algorithms: Major update 1

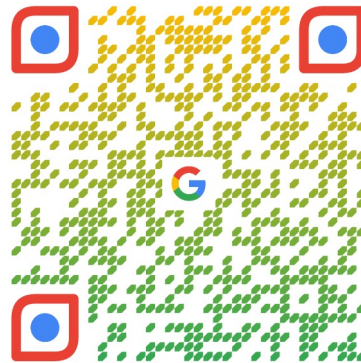
- Model(s) of Choice:
 - Understand concurrency basics without the trappings of real systems (routing, data alignment etc.).
 - Recognize the PRAM as embodying the simplest forms of parallel computation: Embarrassingly parallel problems can be sped up easily just by employing many processors.
 - Recognize how a completely connected network abstracts away from routing details. Recognize the difference between the model(s) and real systems.
 - Such a Model of Choice (MoC) is assumed to be chosen and adopted by the instructor on which PDC concepts would be discussed.

Algorithms: Major update 2

- Deeper Algorithmic Experience:
 - Experience through class instruction and assignment/project the (1) design, (2) analysis, and (3) implementation aspects of at least one parallel or distributed algorithm of choice in detail.
 - Master PDC algorithmic concepts through a detailed exploration, including recognizing how algorithm design reflects the structure of the computational problem(s) and the PDC model/environment.
 - Possible computational problems (to be chosen by instructor(s)) include, but are not limited to, matrix product, map reduce, sorting, search, convolution, a graph algorithm of your choice, etc.
 - We recommend that all CS/CE undergrads have this experience at some point before graduating.

Algorithms Area Survey

<https://forms.gle/isxtjEKqa2Y3TUNq9>



Emerging Aspects

- Big Data
- Energy
- Distributed Computing

Big Data

Alan Sussman

University of Maryland

Big Data Topics - Overview

- Focus is on the increasing need for PDC in data intensive topics
 - Recognize the need for skilled workforce in data science and machine learning
 - Address the needs of academic programs in data science, data analytics, and machine learning
 - All within a PDC context
- Relevant areas include:
 - hardware/software support for data collection, storage, organization, processing
 - constraints from I/O and memory hierarchies
 - performance bottlenecks from data movement
 - parallel algorithmic approaches for massive data analyses

Big Data Topics Coverage

- Architecture
 - Hardware limitations for data bound computation
 - Pressures imposed by data volume and velocity
- Programming
 - Big data technology stack (layered architecture)
 - NoSQL databases for scalable data manipulation
 - Key-value storage systems
 - Replicated distributed file systems (e.g., HDFS)
- Algorithms
 - MapReduce as a problem-solving technique in distributed systems
 - minimizing data movement
 - Time vs. space tradeoffs when not all intermediate data can be stored

Energy Topics

Krishna Kant

Temple University

Energy Topics – Overview

- Newly added in 2020 curriculum update due to increasing importance
 - Usually not a part of traditional UG CS/CE UG curricula.
 - Power consumption is key driver of increased parallelism and heterogeneity.
- The energy topics can be addressed in the core Systems class.
- Energy is a cross-cutting topic
 - Pervasive (Performance & energy tied intimately)
 - Architecture (HW design, power mgmt)
 - Algorithms
 - Programming (data layout and use)
- Energy consumption crucial across the entire computing spectrum
 - Embedded, wearable, mobile, data center/HPC

Coverage of Some Energy Topics

- Architecture
 - Leakage current, idle power, dynamic voltage/frequency control
 - Power advantages of simpler designs
 - Heterogeneous architecture saves energy
- Programming
 - Data layout and usage affect energy
 - Energy cost of data loading into memory
 - Operations to facilitate use of active/idle power control
- Algorithms
 - Data movement vs. computing tradeoff
 - Energy aware process scheduling
 - Decomposition into CPU/GPU computation

Distributed Computing

R. Vaidyanathan

Louisiana State University

Distributed Computing Topics – Overview

- More centrally addressed in 2020 curriculum update
 - Usually addressed indirectly through (operating systems, computer networks, etc.)
 - Important to understand fundamentally. Many more “visible” distributed computing applications now
- Distributed Computing concepts span various areas
 - Asynchrony, Locality, “partial knowledge”
 - These concepts cut across almost all “computing” areas
- Modern applications
 - Google docs, Zoom, Cloud Services, Cyber-Physical Systems, the Internet

Distributed Computing Topics - Coverage

- Architecture

- Hazards: expressions of time and local knowledge
- Buses and shared resources: symmetry breaking
- Interrupts, coherence, atomicity

- Programming

- Distributed memory and file systems: communication, collective operations, locality
- Distributed file systems: Consistency models, replication
- Deadlock, Livelock, Starvation: Local knowledge

- Algorithms

- Model of Choice
- Performance Complexity: communication and “locality” cost
- Broadcast and Reduction: Tree based distributed computations
- Mutual exclusion and conflict resolution

- Pervasive

- Asynchrony and Locality

Emerging Aspects Survey

<https://forms.gle/njwACGohHdMwPQgD9>

Aspects of Big Data,
Energy and Distributed



Q&A

TCPP Curriculum Initiative
<https://tcpp.cs.gsu.edu/curriculum/>



Feedback & Participation: sushil.prasad@gmail.com

Curriculum Feedback Surveys

Architecture
Area



Programming
Area



Algorithms
Area



Aspects of Big Data,
Energy and Distributed

