ABET Accreditation: A Way Forward for PDC

Sherif Aly, Haidar Harmanani, Rajendra K. Raj, and Sanaa Sharafeddine

EduPar-21
May 17, 2021

Why Parallel and Distributed Computing?

- It is the new computing nexus and is linked to new computing advances
  - Deep learning
  - Cloud computing
  - IoT
  - ...
- Computing education has been lagging behind in formalizing PDC in curricula
  - Various curricular moves to tackle this issue (CS2013, CC2020)
- Computing education programs need to incorporate PDC concepts

Accreditation to Move PDC Forward

- ABET's Computing Accreditation Commission (CAC) has been accrediting computer science programs since 1985-86 academic year
- At present, CAC accredits 398 undergraduate CS programs around the world using the ABET Criteria for Accrediting Computing Programs (Criteria)
  - It also accredits undergraduate programs in Cybersecurity (associate's too), Information Systems, Information Technology, and other general computing programs
- ABET's Engineering Accreditation Commission (EAC) accredits computer and software engineering programs
- The CS Criteria comprise the General Criteria that apply to all computing programs and the Computer Science Program Criteria that are specific to CS programs.

PDC and ABET CS Program Accreditation

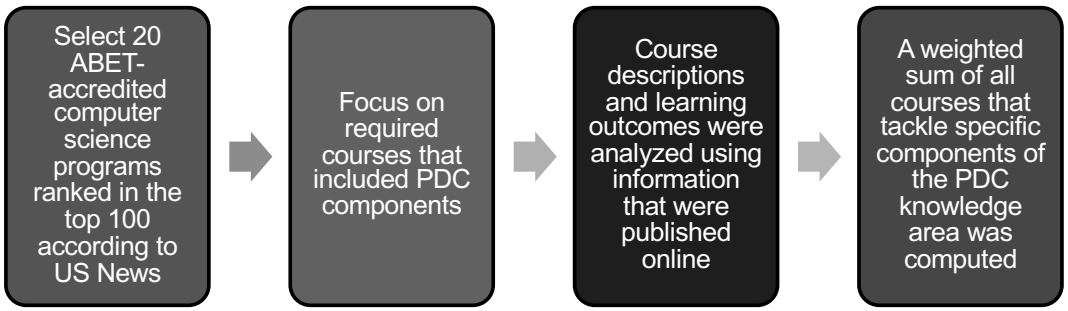
> . . .
> The curriculum requirements specify topics, but do not prescribe specific courses. For CS, the requirements are: at least 40 semester credit hours (or equivalent) that must include (among several other topics):
>
> - . . .
> - Exposure to computer architecture and organization, information management, networking and communication, operating systems, and **parallel and distributed computing**.
> - . . .

Question: What is the best way to enable PDC in modern computer science curricula?

Paper Approach

- Review PDC coverage in top accredited programs
- Use three different ABET-accredited computer science programs as case studies
- Review ACM and IEEE PDC curricular guidelines in CS, CE and SE programs
- Recommendations on possible approaches to cover PDC in modern CS curriuclua

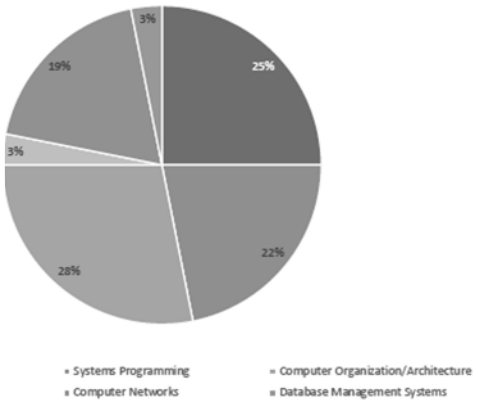
PDC In the Top 20 ABET-Accredited CS Programs



PDC Topics in CS Courses

	Systems Programming	Computer Organization/ Architecture	Operating Systems	Database Systems	Computer Networks
Programming with threads	×		×		×
Transactions processing				×	
Parallelism and concurrency	×	×	×	×	×
Shared-Memory programming	×		×		
Inter-Process Communication (IPC)	×		×		×
Atomicity	×		×		
Performance measurement, speed-up, and scalability		×			
Multicore processors		×			
Shared vs. distributed memory	×	×	×		
SIMD and vector processors		×			
Instruction Level Parallelism		×			
Flynn's taxonomy		×			
Client-server programming	×				×
Memory and caching	×	×	×		

PDC Topics in Courses



PDC CS Case Study 1: Lebanese American University

- Program supports PDC through multiple courses
  - Parallel programming, operating systems, computer organization, and database management systems
- ABET PDC requirement is met through a dedicated parallel programming course that has the following outcomes:
  - Understand the challenges of as well as the motivations for using parallel programming.
  - Demonstrate an ability to analyze the efficiency of a given parallel algorithm
  - Demonstrate an ability to design, analyze, and implement programming applications using multicore and manycore systems.

A Course on Parallel Programming

- Course delivery has three main components
  - Recent PDC driving forces: decrease of feature size, increase in the number of transistors per chip, clocking, performance scalability, and power consumption
  - Multicore programming using Pthreads and OpenMP
  - Manycores computing including SIMT execution model
- Course is supported by Intel and by NVIDIA's Deep Learning Institute (DLI)
  - Students can earn two NVIDIA certificates in this part, an OpenACC and a CUDA C
  - Labs are completed on the cloud using NVIDIA's accelerated computing environment

PDC CS Case Study 2: American University in Cairo

- Uses an early maturity pedagogical approach to ensure students learn key knowledge units as early as possible in the curriculum
- PDC is enabled through knowledge units that are satisfied across various courses
  - Ability to create and control threads, as well as a simple client server connectivity in fundamental course sequences
  - Computer organization and architecture: multiprocessing, thread level parallelism, pipelining, instruction level parallelism, superscalar architectures, VLIW architectures and architectures based on dynamic scheduling
  - Operating systems: multi-threading, speedup, multiprocessing, mutual exclusion, synchronization, deadline and starvation, and scheduling on single/multiprocessor systems
  - Further exposure to PDC concepts in concepts of programming languages and software engineering

PDC CS Case Study 3: Rochester Institute of Technology

- The CS program at RIT realized ~2010 the importance of PDC in CS education
- Offers a coherent course called *Concepts of Parallel and Distributed Systems*
- Course objectives include:
  - Exposure to how parallel and distributed systems function AND how to write programs for these computing systems
  - Exposure to synergies between various topics such as multithreaded programming and network programming
- Also, enables PDC concepts starting in the freshman year using Java threads and synchronization, as well as in the required Concepts of Computer Systems course that covers pipelining and other related PDC concepts

PDC in Engineering Programs

- No explicit mention of PDC in ABET's engineering accreditation criteria
  - Requires that the program "provide adequate content for each area, consistent with the student outcomes and program educational objectives, to ensure that students are prepared to enter the practice of engineering"
- ACM/IEEE CS Curricular Guidelines
  - CE2016 delineates 12 broad knowledge areas that are decomposed into associated knowledge units with a list of learning outcomes each
  - SE14 comprises 10 knowledge areas with a set of associated knowledge units, some of which are designated as essential in relevance to the core knowledge

PDC in Computer and Software Engineering Programs

- Parallel and distributed computing concepts are explicitly addressed in CE2016

Knowledge Area	PDC-related Core Knowledge Units
Computing Algorithms	Parallel algorithms/threading
Architecture and Organization	Multi/Many-core architectures Distributed system architectures
Systems Resource Management	Concurrent processing support
Software Design	Event-driven and concurrent programming

- Computing essentials area is a main knowledge area of SE2014 - SEEK and emphasizes two PDC-related topics

Knowledge Area	PDC-related Core Topics
Computing Essentials	Concurrency primitives (e.g., semaphores and monitors) Construction methods for distributed software (e.g., cloud and mobile computing)

PDC in Engineering Programs: Some Thoughts

- Computer engineering and software engineering programs *inherently* address PDC concepts with varying levels of attainment
- ABET criteria for computer engineering tackles, among others, the analysis and design of complex software
  - Incorporating PDC concepts in software design applies to this criterion
- Recommended cognitive skill level for concurrency and construction methods in SE2014 strengthens compliance with ABET program criterion for SE on the development of complex software systems

Final Remarks

- Accreditation and curricular guidelines can help to incorporate PDC into programs
- ABET's CAC criteria for CS has required PDC exposure (since 2018)
- Systematically increased PDC content in accredited CS programs using:
  - A predominant approach that infuses multiple topics across various CS courses, OR
  - A dedicated PDC course, OR
  - Both
- EAC Criteria for computer engineering and software engineering do not state PDC
  - However, a modern engineering curriculum inherently requires PDC coverage
- Community push must be on making PDC more prevalent in curricular guidelines
  - ABET is guided by such curricular guidelines