Trying to do it all in a single course: a surprisingly good idea

Tia Newhall

Swarthmore College

EduHPC-20: Workshop on Education for High Performance Computing

in conjunction with SC20

November 13, 2020

PDC at the undergraduate level

Important to expose students to breadth of PDC:

- 2 huge fields, lots of approaches/solutions/problems
- PDC increasingly touches all of CS
- Prepare students for graduate study and careers

There must be depth:

- Time and multiple opportunities to develop understanding
- problem solving/thinking/analysis/application

PDC all in one Course

Seems like a bad idea:

- Two huge fields!
- Lose developing deep/real understanding of PDC
- Lose developing PD thinking and problem solving

Actually: 1 course can provide large breadth of exposure to PDC while developing deep PDC thinking and problem solving

Why single course?

Constraints of a Small Institution

- Fewer faculty, Smaller number of courses
- Liberal Arts College: most courses outside of major

  fewer total courses in major: 8 course CS major

Constraints led to “Do it all in 1 Course” approach

Curricular Goals

Expose to Breadth of PDC

- broad range of PDC topics, concepts, thinking, in different contexts (systems, algorithms, language, tools, applications)

Develop Depth in PDC

- Understanding, thinking, analysis, problem solving

Additional department/institution goals:

- Writing-intensive, written & oral presentations
- Research paper reading & discussion
- Independent research project

Student's Background

Only assume students have had our intro sequence

- CS1, CS2, Intro. to Computer Systems
- Recommend students take 1 UL before this course

Aided by Intro. to Computer Systems pre-req:

- First introduces parallelism, multicore, pthreads
- All enter with some parallel thinking/analysis
- Frees up time to add more advanced PDC topics

Course Structure

Two Parts:

½ Lecture based: lecture and in-class problems

½ Seminar-style: research paper discussion

Also Lab Component:

First half: programming assn. diff lang, tools

- Pthreads, TCP client/server, Cuda, MPI, OpenMP
- designing experiments & analysis

Second Half: focus on course project

Lecture

- Broad range of PDC topics:
  - architecture, algorithms, analysis, systems, languages
- Overarching unifying theme of Scalability
  - Ties disparate topics together
  - Depth: apply in different contexts
- In-lecture group problem solving activities

  (Systems Eval, Parallel Soln, Alg. Analysis, Consensus, …)
  - Students more engaged with problem and solutions
  - Students more comfortable participating in general

Paper Reading & Discussion

- Weekly in-class discussion of papers
  - Papers need to be accessible to undergrads
    - Earlier papers in a field, survey papers
- Assigned Paper Reading Groups
  - Discussion & Writing to prepare for in-class
- Professor leads in-class discussion
  - Goal: understanding of paper details, comparisons

Course Project

- Assigned part way through course
- Topic is Very Open
  - Must have main focus on PDC
  - Structured around research question
- Large written and oral presentation parts
- With more work beyond semester, many have led to publications for students

Student Assessment of Learning Goals

How has the Course affected your ability to:

1. Analyze and critically read CS papers?
2. Formulate research question, design experiments to answer?
3. Write clear & complete research paper?

Universally: students noted large improvements in their abilities of all of these:

"tremendously"

"dramatically"

"significantly" …

What We See

- Huge improvements in student's discussion & analyses of papers, in-class problem solving
- Impressive projects, written reports, oral presentations
  - Getting to pick the topic keeps them engaged, excited, invested in their project
- Sparks interest (and confidence) in PDC
  - Graduate school in PDC areas, seek jobs in field
  - Hear from alumni using what learn in this class
- Extremely satisfying & rewarding to teach!

Lessons Learned

1. Can do lot of breadth and keep in depth,
   and both are important!
   But can't do it all: have to cut important & loved
2. Designing around an overarching theme helps
   tie topics together & provides common depth
3. Need multiple opportunities for depth
   Develop PDC thinking, analysis, problem solving
4. Give Practice in broad range of PDC lang/tools

Conclusions

Our course that covers huge breadth of PDC was born out of constraints of our institution

We think this is the right design for a PDC course at the undergraduate level regardless of institution type/constraints:

- Exposure to breadth of fields & depth
- Sparks interest in PDC
- Prepares students for graduate work & work in the field, regardless of it directly PDC-based or not…ubiquity of PDC!

Thank you

course webpage(s):

`www.cs.swarthmore.edu/~newhall/cs87`