Introduction to Scientific Computing

… or “The Computer Science I wish I’d learned in school.”

Tim Mattson

Human Learning Group

Introduction to scientific computing

In the course of working on your degree, you will naturally be exposed to the computational problems most important to your field. A survey of such problems would not be very useful.   What we will focus on instead is those topics of computer science that everyone doing scientific computing must know. It's all the computer science I never learned in school that I wish I had.

The emphasis is on parallel computing used in scientific computing but we will cover computer arithmetic, pseudo-random numbers, the C programming language, and other topics in computer science everyone working in scientific computing must know.

The class will be interactive, noisy, and 2/3 hands-on. I assume you have some programming experience in a high-level language such as Python. We'll learn the C we need for the course together. Bring your laptops to the lectures and we will mix exercises with lectures so we can work on them together.

As for Exams, I don't like them. They have a place, but not in a course such as this one. Instead, you will do a simple and fun project of your choice where you choose a problem you are personally excited about and give a 5-to-10-minute presentation about it to the class. The key is for you to show me creative engagement with the material in any relevant way you choose.   We will work together on the problem … but my hope is that it's fun and won't take an unreasonable amount of work outside class. I know you all have work for other courses you need to do and I don't want this class to be a burden.

Text Book?

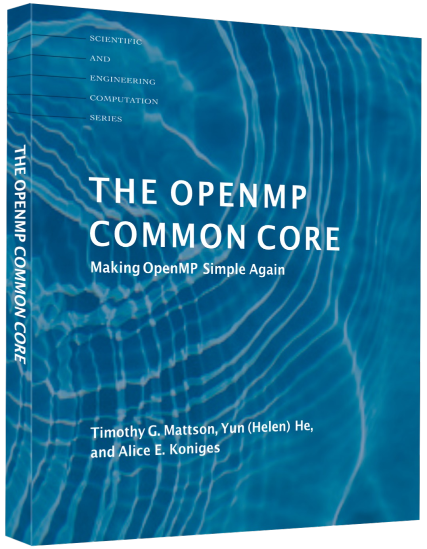
- There is no text book for the course.

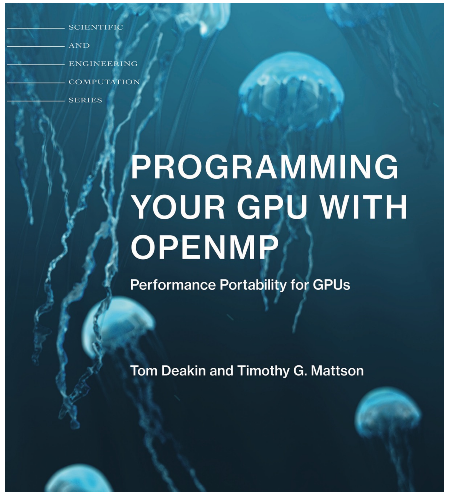
- My slides contain lots of words so they can serve as briefing documents for study after the lecture. That should be enough.

- If additional study material is desired, I highly recommend the online text books at the following URL:

  https://theartofhpc.com/

- My OpenMP books provide good reference material as well but you do not need to buy them.




SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES
PROGRAMMING YOUR GPU WITH OPENMP
Performance Portability for GPUs
Tom Deakin and Timothy G. Mattson

The Plan for this course

- This is a student driven course. The plan will evolver around your interests.

- I see the following components to the course.
  - An introduction and overview of the field of scientific computing
  - Computer Science part 1: Computer architecture, supercomputing, and programming languages
  - Preparing for our hands-on work: Linux, C, git and more
  - An introduction to Parallel computing with OpenMP
  - Programming your GPU with OpenMP
  - Cluster computing and MPI
  - Computer Science part 2: Floating Point numbers aren't real and your random numbers aren't random
  - Computer Science part 3: A quick introduction to data management and the future of software
  - What about Python? Can I do HPC with Python? ← a topic we will cover if time allows

Course materials will be managed through git hub in the repository:
https://github.com/tgmattso/SciCompHPC.git

Grading, Tests, and all that icky stuff

- This is a course where I want you to feel free to take risks. Hence, there will be NO high pressure tests and homework is evaluated on effort, not results.

- Grading:
  - 40% of the grade is on participation. In particular, is it clear to me you really worked hard on understanding the exercises?
  - 60% of the grade is on a course project.
    - The course project is required to be fun for you.
    - You must pick something that fascinates you
    - Pick some topic from the course and show how it interacts with the something that fascinates you.
    - Creatively share your passion for the topic with the class to educate them on it.
      - This can be a regular PowerPoint lecture, a musical composition, art work, dance … I don’t care as long as it shows a deep understanding of some aspect of the course.
      - Presentations will be 5 to 10 minutes. You are free to work in groups, but you need to show me that everyone in the group contributed.

Schedule

Mon	Tue	Wed	Thu	Fri	Sat
28/10 18:15-20:15					
4/11 18:15-20:15				8/11 8:15-10:15	9/11 9:00-11:00
11/11 18:15-20:15					
Super Computing 2024 in Atlanta Georgia					
25/11 18:15-20:15					30/11 9:00-11:00
2/12 18:15-20:15				6/12 8:15-10:15	
9/12 18:15-20:15				13/12 8:15 - 10:15	
16/12 18:15-20:15					

Topics

OpenMP is the most commonly used parallel programing model today.

It is also the easiest to use … we can cover most topics in parallel programing theory using OpenMP

In hands-on sessions we will be writing code together in class.

Bring well-charged laptops.

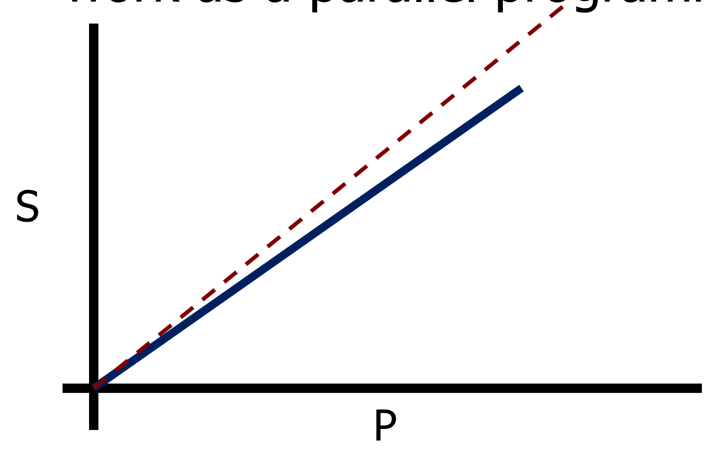
Mon	Tue	Wed	Thu	Fri	Sat
Intro to Sci Comp					
Computer Architecture				Using HPC systems + start OpenMP	OpenMP
OpenMP					
Super Computing 2024 in Atlanta Georgia					
OpenMP					GPU programming
GPU Programming				Cluster computing with MPI	
The joys of computer arithmetic				How databases work .. Student presentations	
The Future of computing ... Student presentation					

An Introduction to me

I'm just a simple kayak instructor

Photo © by Greg Clopton, 2014

To support my kayaking habit, I work as a parallel programmer



Which means I know how to turn math into lines on a speedup plot

My 40 year Career

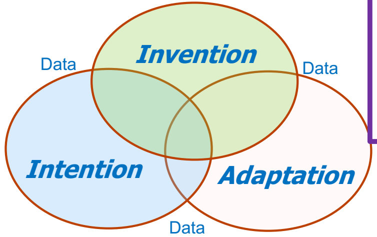
[tile]DB

2015
Sparse Array storage engine

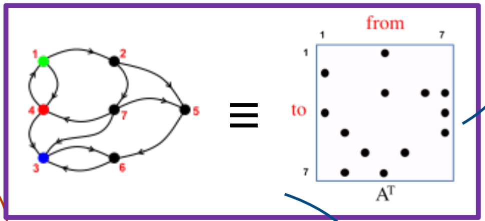
2015
BigDAWG Polystore system

2018-Aug'2023

oneAPI
Future of heterogenous computing



Machine Programming



Unified Theory of data and computing

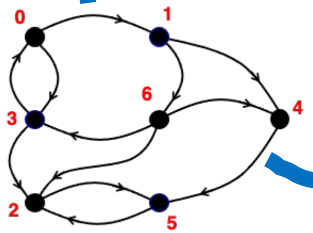
2023-2038
Retirement

Tim Mattson RIP

1958-2038

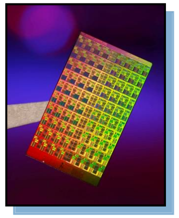
Here lies the Christmas bat

Write, teach, kayak, and think



2013
GraphBLAS

OpenCL
2008
Portable GPU programming

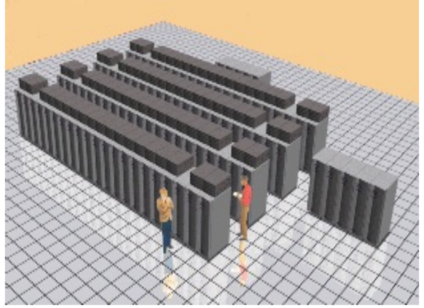


2007
World's First TFLOP chip

2002-2005
Director of life sciences

OpenMP

1997
Portable multithreading



1996
ASCI Red:
World's first TFLOP

PhD chemistry
1979-1985

PostDoc 1986
Caltech Concurrent Computation Project

Jobs with startups …
numerical analysis, signal processing,
scientific computing, and parallel computing
1987-1990

Yale University
1990-1993
Linda and pre-MPI message passing

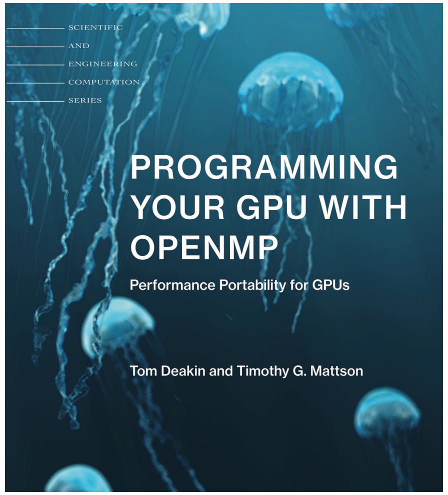
intel
1993
Supercomputing Systems Division

MPI
1994
Message Passing Interface

FDIV inside
1994
FDIV Bug

I love to write … and I love to teach

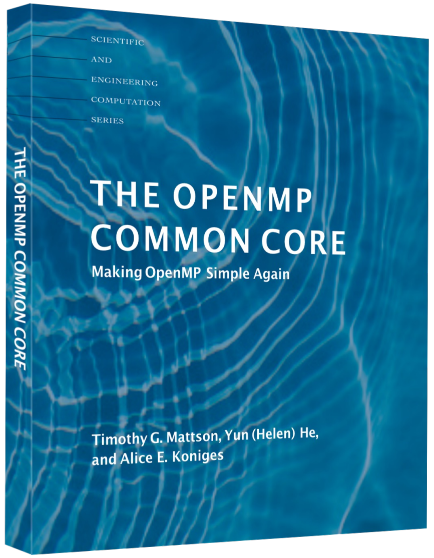
- I’ve written 6 books on different topics in parallel programming.


SCIENTIFIC AND ENGINEERING COMPUTATION SERIES

PROGRAMMING YOUR GPU WITH OPENMP

Performance Portability for GPUs

Tom Deakin and Timothy G. Mattson

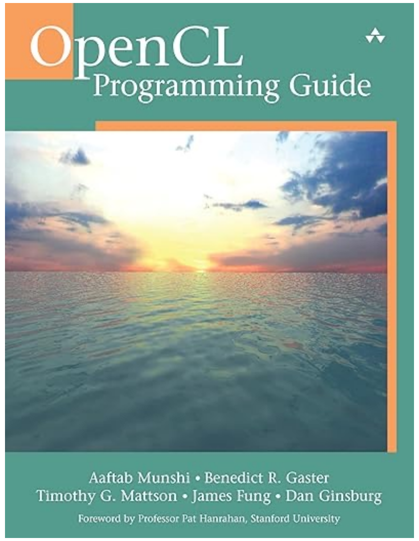

SCIENTIFIC AND ENGINEERING COMPUTATION SERIES

THE OPENMP COMMON CORE

Making OpenMP Simple Again

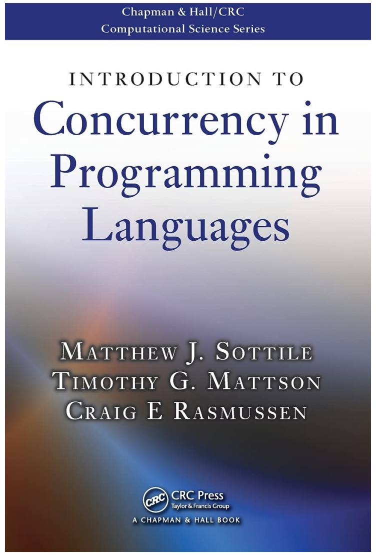
Timothy G. Mattson, Yun (Helen) He, and Alice E. Koniges

THE OPENMP COMMON CORE


OpenCL Programming Guide

Aaftab Munshi • Benedict R. Gaster
Timothy G. Mattson • James Fung • Dan Ginsburg

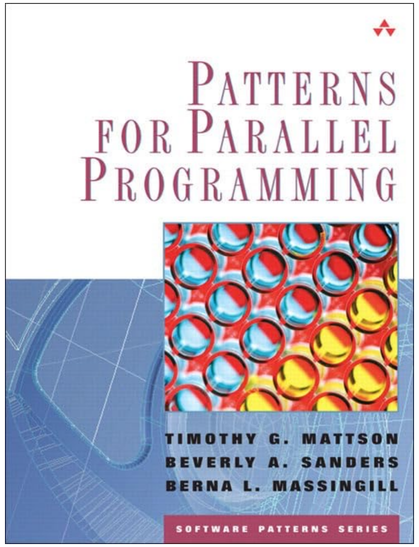
Foreword by Professor Pat Hanrahan, Stanford University


Chapman & Hall/CRC
Computational Science Series

INTRODUCTION TO
Concurrency in Programming Languages

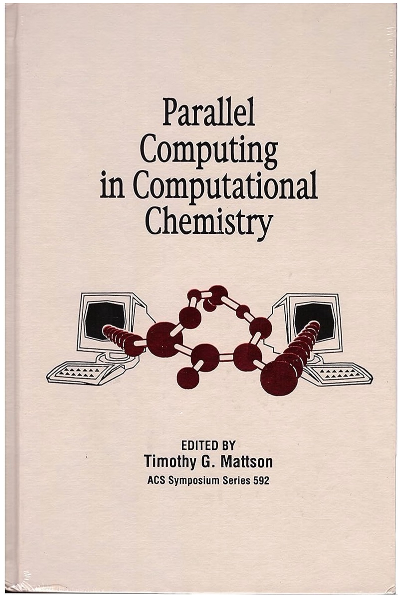
Matthew J. Sottile
Timothy G. Mattson
Craig E Rasmussen

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK


Patterns for Parallel Programming

TIMOTHY G. MATTSON
BEVERLY A. SANDERS
BERNA L. MASSINGILL

SOFTWARE PATTERNS SERIES


Parallel Computing in Computational Chemistry

EDITED BY
Timothy G. Mattson
ACS Symposium Series 592

- I’ve taught hundred’s of thousands of people how to write parallel software over the last 25 years.