Introduction to Parallel Computing

Chris Kauffman Chris Kauffman

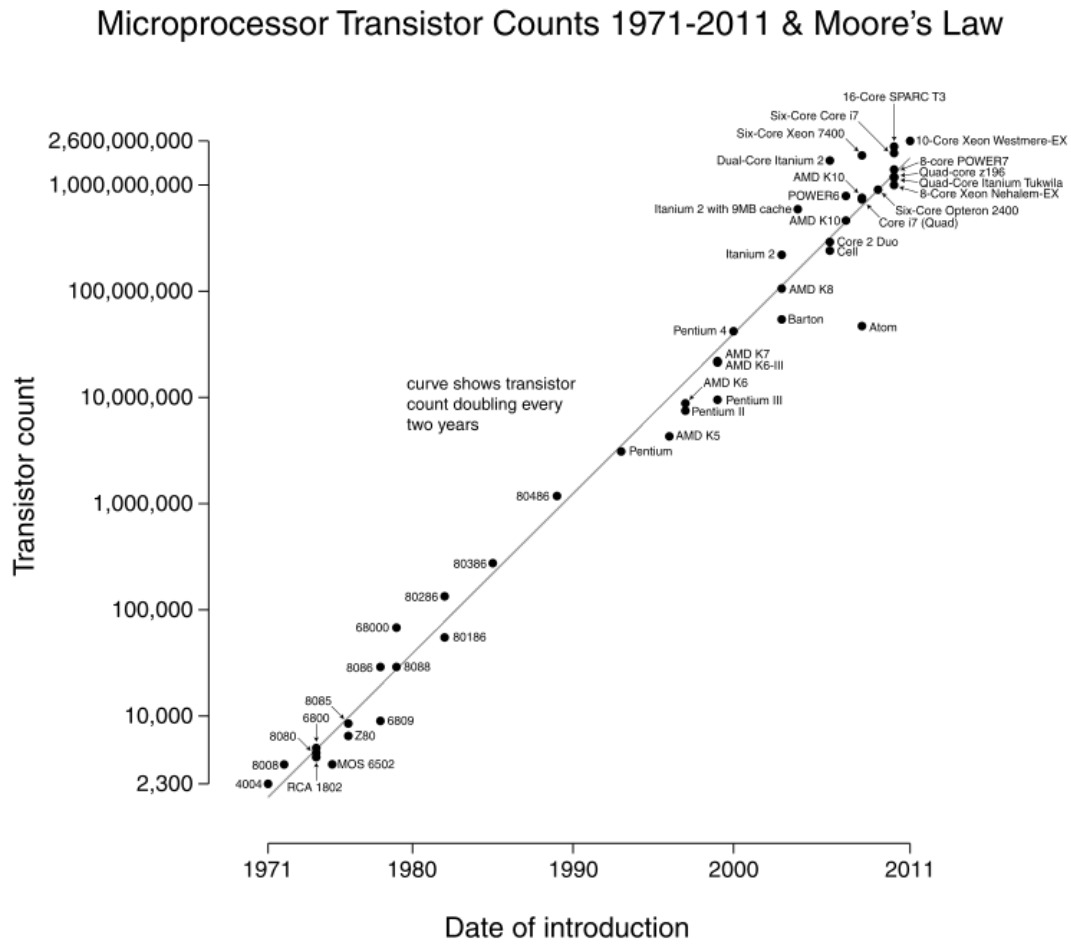
CS 499: Spring 2016 GMU

Goals

- Motivate: Parallel Programming
- Overview concepts a bit
- Discuss course mechanics

Moore's Law

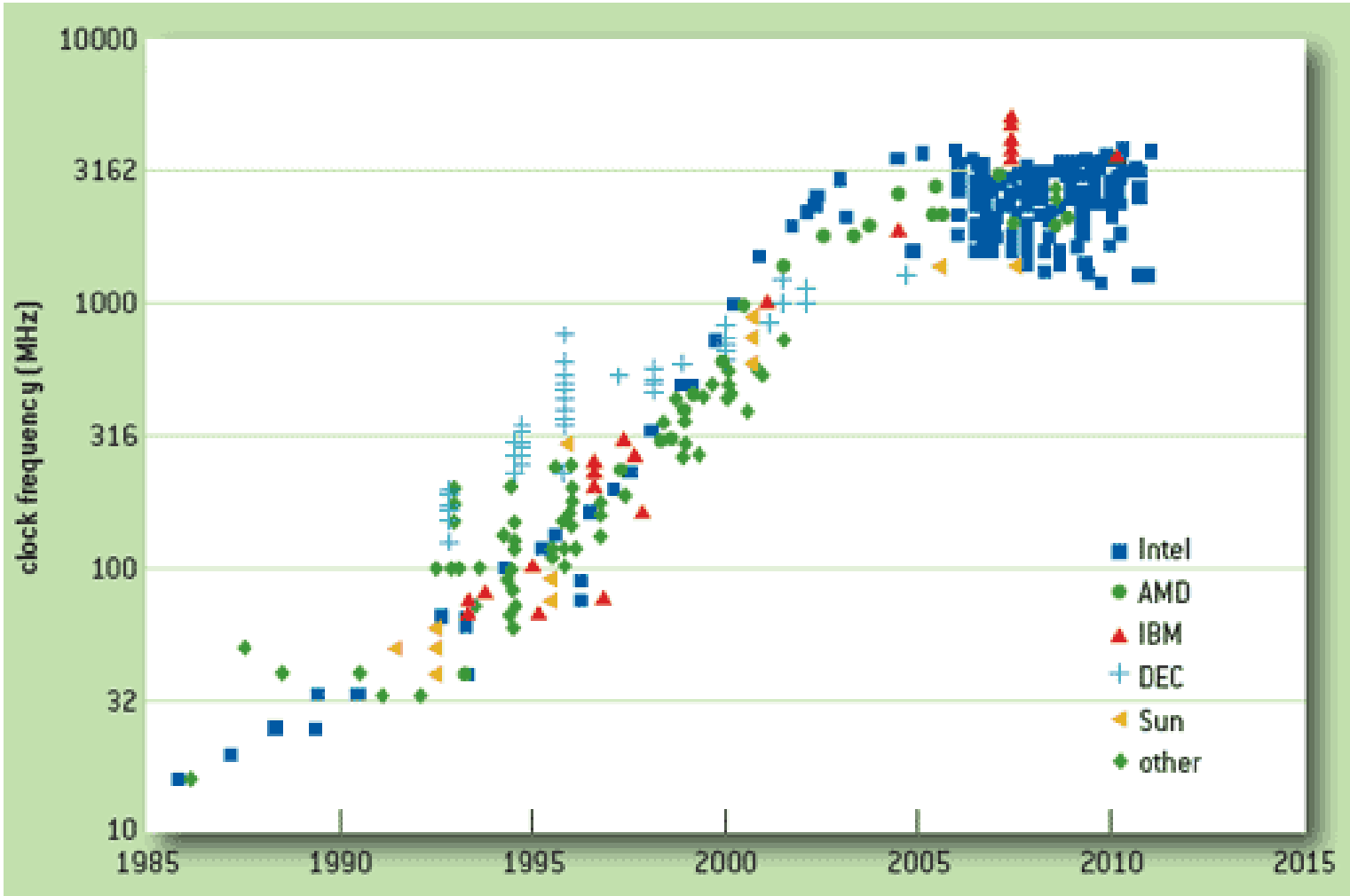
- Smaller transistors $\rightarrow$ closer together
- Smaller transistors can "flip" faster
- More faster transistors on a chip $\rightarrow$ more speed
- **Processor speed doubles every 18 months**



How Small are Transistors?

- Intel Core i7 uses a 32 nanometer process
- Distance between memory units in the processor is about 64 nanometers
- A hard sphere radius of a hydrogen Atom is about 0.11 nanometers
- About 591 **atoms** apart
- 22 nanometer processor is close

However. . .



Source: Danowitz et al

- CPU speed isn't getting faster these days
- Fastest Dell Speed I found was 4.0 GhZ (mine is 2.0 GhZ)

Today's Processors

Mini-Parallel Computer

- Multiple independent processors
- Can add 2 or 4 or 8 numbers independently
- Can actually run multiple programs simultaneously

Multitasking

multitask (verb)

To use the restroom and brush your teeth at the same time.
I was late for work today so I had to multitask!

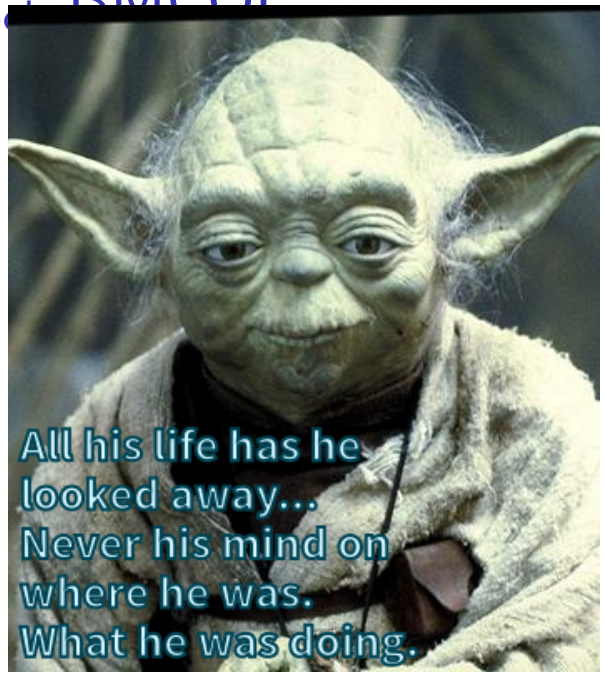
- Urban Dictionary Definition 2 by sasm

Questions

- Do humans multitask and if so how?
- What kinds of things can humans multitask?
- Are you a good multitasker?
- How do humans get a big job done faster?

Focus

:B$_{\text{blo}}$:BMCOL:



Study on Canadian Undergraduates

The students in the first experiment who were asked to multitask averaged 11 per cent lower on their quiz.
The students in the second experiment who were surrounded by laptops scored 17 per cent lower.
Laptop use lowers student grades, experiment shows, The Canadian Press, 8-14-2013
Original Paper by Sana et al

Computers and Multitasking

- How do computers multitask?
- Can a single CPU computer multitask?
- What are the advantages/disadvantages of this?
- What might drive one to write a parallel program?

Parallelism Shows Up a Lot in Computing

Low-level/hardware parallelism

- CPU Instructions,
- VLIW: Very Long Instruction Word
- Multi-media CPU instructions
- Graphics and GPU instructions
- CPU Pipelines
- Memory subsystem
- I/O controller

Many of these are like your heart-beat and breathing: not part of your explicit execution but as they run more efficiently your whole game gets better

Programmatic Parallelism

You write code meant to run in parallel

Motivation: Faster, Bigger

Google's Index of Web Pages

Item	Size/Count
Number of Pages indexed in 2014	30,000,000,000,000
Total number of Google servers	920,000
Total size of Googles index data	100,000,000 GB

Source: Statistic Brain

Questions

- How does Google rank web pages?
- What algorithm is used?
- Could you run it on your laptop?

Primary Motivation for Parallel Computing

Use multiple processors to...

- **Speedup**: solve the same size problem in less time
- **Sizeup**: solve a larger problem in the same time

Example: A *serial* program takes 100 seconds to perform a calculation on a 10MB input image file.

Speedup

- 2 processors should finish the 10MB calculation calculation in 50 seconds
- Right? .... right?
- Just like two people can bake a cake twice as fast as a single person...

Sizeup

- 2 processors should finish a 20MB file calculation in 100 calculation on a 20MB input image file?
- Right? .... right?
- Just like 2 speed boat captains can drive an aircraft carrier...

Concurrency and Parallelism

StackOverflow Questions

Concurrency vs Parallelism - What is the difference?

Accepted Answer (RichieHindle)

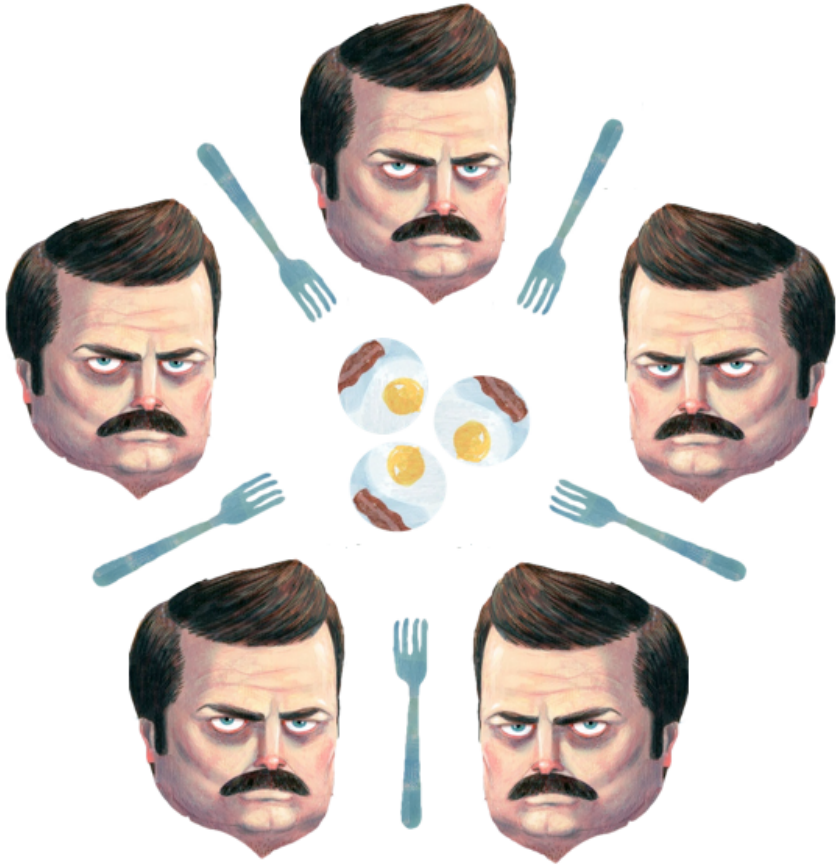
Concurrency is when two or more tasks can start, run, and complete in overlapping time periods. It doesn't necessarily mean they'll ever both be running at the same instant. Eg. multitasking on a single-core machine.
Parallelism is when tasks literally run at the same time, eg. on a multicore processor.

Implications

$\textit{Concurrent} \not\Longrightarrow \textit{Parallel}$: A concurrent program does not need to be run in parallel.
$\textit{Parallel} \Longrightarrow \textit{Concurrent}$: A parallel program had better deal with concurrency issues.

Coordination has a Cost: Dining "Philosophers"

- Each Swansons will only eat with two forks
- JJ's only has 5 forks, must share
- Each Swanson should eventually eat some of the eggs
- Algorithms that don't share forks will lead to injury



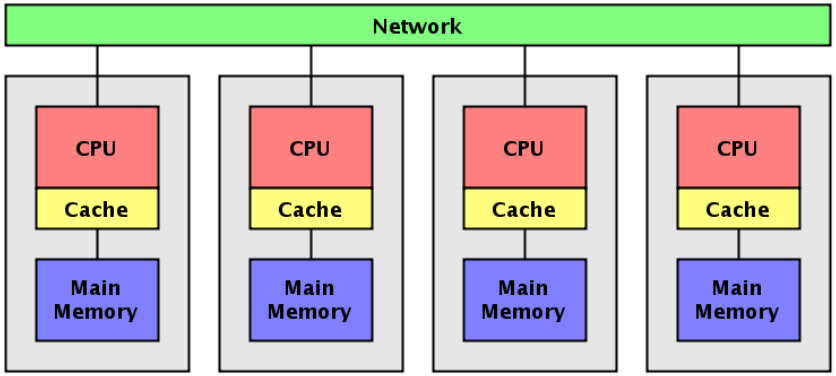
Source: Aditya Y. Bhargava,
Originally: Dustin DArnault

What We Will Discuss

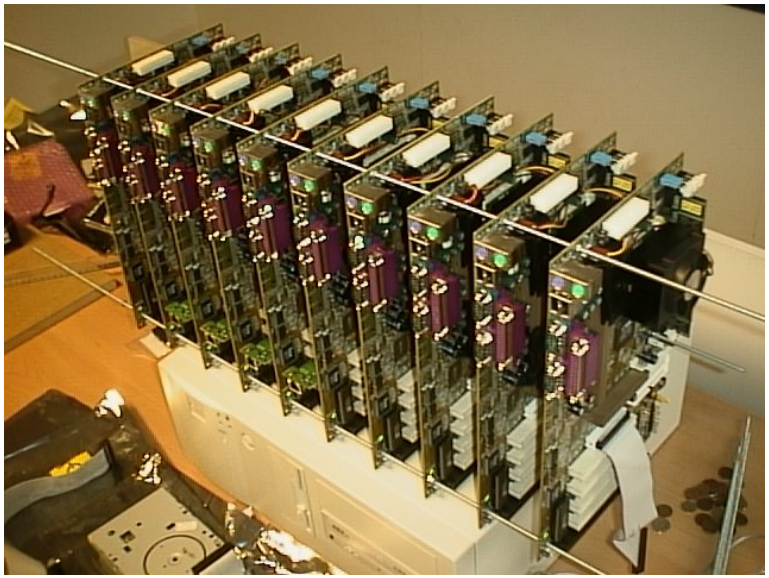
- Low latency parallel computers
- General hardware/network architectures of parallel computers
- Distributed memory parallel computers
  - Message Passing Interface (MPI)
- Shared Memory parallel computers
  - Interprocess Communication
  - OpenMP
  - POSIX Threads
- Co-processor Parallelism such as in GPUs
  - CUDA for NVIDIA GPU programming
- Analyzing Parallel Algorithms
- Parallelizing Classic Algorithms
  - Matrix Ops
  - Sorting
- *Scientific computing* angle

General Arrangement of Topics

Weeks 1-7 Distributed Memory

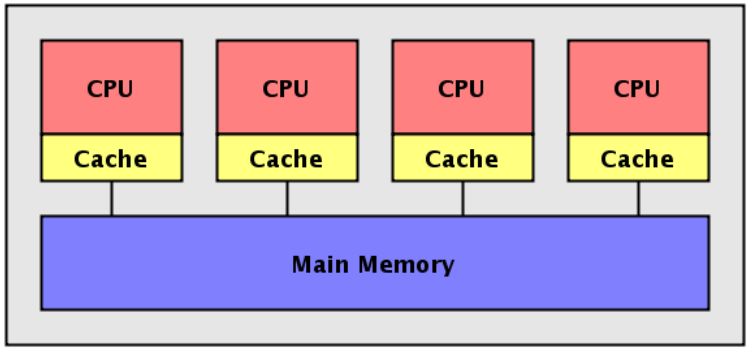


Source: Kaminsky/Parallel Java

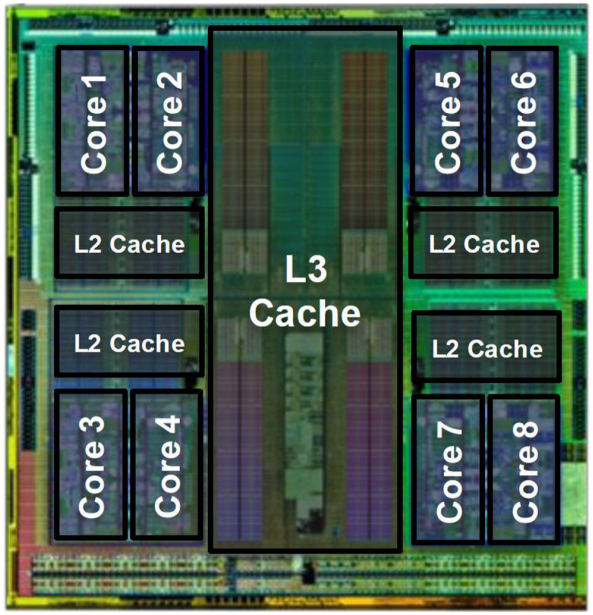


Source: ClusterComputer.com

Weeks 8-14 Shared Memory



Source: Kaminsky/Parallel Java



Source: Bit-Tech AMD FX-8350 review

What We Won't Discuss

- General Networking (CSCI 4211/5211)
- Server/Client Model for Web Programming (CSCI 4131)
- Deep Dive into Parallel Hardware (CSCI 5204)
- Graphics applications for GPUs (CSCI 4611 / 5607+8)
- Distributed/Grid Computing (CSCI 5105 / 5751)

Course Mechanics

Mull over the syllabus here:
https://cs.gmu.edu/syllabus/syllabi-spring16/CS499KauffmanC.html
Look for

- Contact info
- GTA
- Textbook
- Weights / Grading Scheme
- Hot Seats / Bonus Cards

Schedule of topics and lecture materials:
https://cs.gmu.edu/~kauffman/cs499/schedule.html

Homework

- 50% of your grade
- Combination of writing and programming
- Programming mostly in C in a unix environment
- One major programming HW on MPI + analysis
  - Getting a parallel cluster set up for you to use
  - Also probably use a simulator to analyze scale up
- One major programming HW on PThreads/OpenMP + analysis
  - Will use `zeus.vse.gmu.edu` (8 cores)
- Three other assignments, combination of some code and written work
- **Dust off your C/unix skills**
  `ssh`, `gcc`, `make`, `shell`: know them, love them

Exams Small and Large

Mini-Exams: 4 scheduled

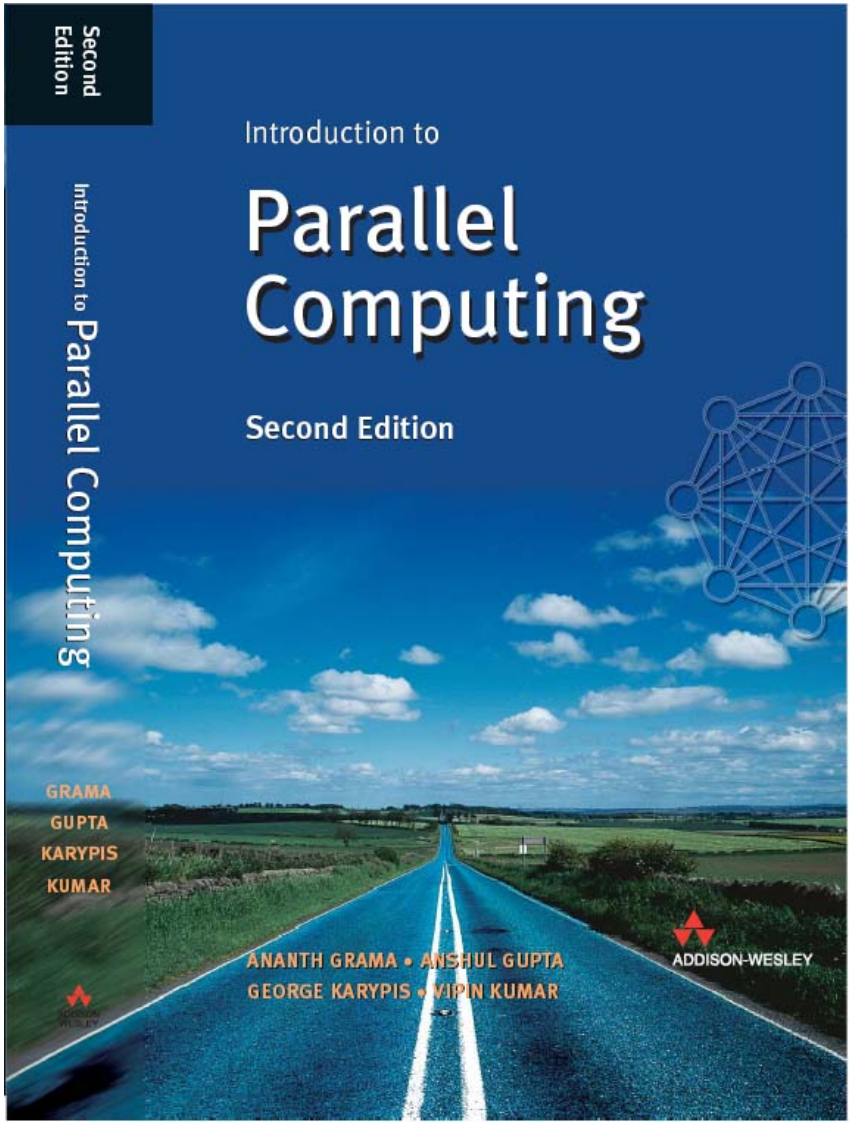
- Between a quiz and a midterm exam in length
- Last 30 minutes of class
- 1 page, front and back
- Open notes, book, slides
- Stuff from lecture, HW, readings
- Will work practice problems in class preceding them
- Total 25%

Final Exam: 2 hours 45 minutes, end of semester

Open Resource Exams

Unless otherwise specified, exams are open resource: use notes, compiler, code, slides, textbook. No googling, browsing, communication, cheating

Reading For Next Time



- Grama Ch 2: Parallel Programming Platforms
- Link to Library Copy of Textbook