NC STATE UNIVERSITY

Course Overview

Lecture 1 (Chapter 1) http://go.ncsu.edu/ece506

http://go.ncsu.edu/ece506

Learning Objectives

- 1. Understand the problem of race conditions in concurrent systems,
- 2. Learn how to decompose a program for parallel execution,
- 3. Be able to write simple parallel programs in the important programming models,
- 4. Understand the operation of common cachecoherence protocols, both bus-based and networkbased, and
- 5. Learn about common memory-consistency models, and appreciate the advantages and disadvantages of each.

Textbook



NC STATE UNIVERSITY

CSC/ECE 506: Architecture of Parallel Computers

TAs





Abhishek Bajaj

Tripti Samal

NC STATE UNIVERSITY

CSC/ECE 506: Architecture of Parallel Computers

Instructor





Where do you find hope, truth, and life?

We encourage you to apply yourself and engage your mind fully in the pursuit of knowledge and academic training in your field of study. College can be one of the best experiences of your life; lasting friendships are developed and your future path is chosen. While here you will no doubt be exposed to various philosophies. Jesus said, *"I am the Way, the Truth, and the Life. No one comes to the Father except through me."* Jesus never forced or coerced anyone to follow him or believe in him. Neither do we. We simply invite you to explore his life and perhaps find new life. We are a group of faculty and staff who are united in our discovery and experience that Jesus Christ provides intellectually and spiritually satisfying answers to life's most important questions. Interested? Have questions? Talk with us or go to EveryStudent.com or MeetTheProf.com. **Contact us at cfsn-ncsu.org or info@cfsn-ncsu.org – Sponsored and paid for by the Christian Faculty Staff Network at NCSU**

Everette Gray Allen – OIT IT Specialist Dr. Chris Austin – CSAPC Dr. Steve H. Barr – Management, Innovation & Entrepreneurship Valerie Basham – NC State Veterinary Hospital Carrie Baum-Lane – Applied Ecology Dr. Mark Beasley – Department of Accounting Donise Benton – Communications Dr. Emity Zechman Berglund – Civil, Const. & Environ. Engineering Dr. Roy Borden – Prot. Emeritus Civil Engineering Dr. Michael Boyette – Biological & Agricultural Engineering Dr. Michael Boyette – Biological & Agricultural Engineering Dr. Michael Boyette – Biological & Agricultural Engineering Dr. Michael Boyette – Department of Computer Science Dr. Rick L. Brandenburg – Entomology & Plant Pathology Dr. Joseph Brazel – Department of Accounting Dr. Steve Broome – Crop & Soil Science Dr. Andy Hale — Biological & Agricultural Engineering Dr. Steven Hall — Biological & Agricultural Engineering Alicia Harris — College of Humanities & Social Science Felicia Harris — Office of Global Engagement (IEP) Dr. Gabriel Keith Harris — Food, Bioprocessing, & Nutrition Sciences Nikki Harris — College of Natural Resources Dr. Robert B. Hayes — Nuclear Engineering Dr. Gary Hodge — Forestry & Environmental Resources Gail Hill — GTI Dr. Daniel Israel — Crop & Soll Sciences Lee Ivy — Horticultural Science Dr. Jeffrey Johnson— Biological & Agricultural Engineering Dr. Chad Jordan — Plant & Microbial Biology Dr. David L Jordan — Crop Science Dr. Jason Painter – The Science House Dr. Michael Parker – Horticultural Science Lacy Parrish – Biological & Agricultural Engineering Dr. Bob Patterson – Crop & Soll Sciences Jill Phipps – OIT-Business Services Dr. Carrie Pickworth – Animal Sciences Wayne Pollard – Heatth & Exercise Studies Dr. Samuel B. Pond III – Industrial Organizational Psychology Dr. Daniel H. Poole – Animal Sciences Terry K. Price – Poole College of Management Cynthia P. Pullen – Friday Institute Dr. Bobby Puryear – Department of Economics Dr. Scott Ragan – The Science House Dr. Gary Roberson – Biological & Agricultural Engineering James E. Robinson III – Friday Institute

6

NC STATE UNIVERSITY

"Attendance" requirement

- You are required to "attend" 20 of the 26 classes.
 - **16** of these must be in the classroom.
 - "Attend" \rightarrow Respond intelligently to $\geq \frac{1}{2}$ of Google forms
 - Each one not passed $\Rightarrow -0.5\%$ on semester average.
- You are required to pass 24 of 25 daily quizzes, plus the Syllabus Quiz.
 First one due Wednesday!
 - "Passed" \Rightarrow score of $\ge 80\%$
 - Each one not passed $\Rightarrow -0.5\%$ on semester average.
- You are required to team with 3 students on homework.
 - Each teammate you are lacking $\Rightarrow -0.5\%$ on semester average



Playposit quizzes

- Three lectures (9, 15, 23) will be videos to watch.
- They have embedded quizzes.
 Do the quizzes to get attendance credit.



NC STATE UNIVERSITY

Zoom session

Launch Meeting - Zoom	× +		~	-		×
$\leftarrow \rightarrow$ C \triangle (S http://g	o.ncsu.edu/506zoom	۰ 🔶	Y K	Q	* 👰	÷
Hub Spring 2022	Personal hub - Goo	» 📙 Ot	her bookma	arks 🚹	Readin	g list
zoom				Support	English •	
						П
	Your meeting has been la	unched				
	Don't see your Zoom me	eting?				
By clicking "Launch M	leeting", you agree to our Terms o	f Service and Priv	acy Statem	ient		
	Launch Meeting					
	Don't have Zoom Client installed? Dov	vnload Now				I
На	iving issues with Zoom Client? Join fron	n Your Browser			0	
l						

http://go.ncsu.edu/506zoom

If you join the Zoom session from the classroom, be sure to let me know.

CSC/ECE 506: Architecture of Parallel Computers

Grading

	4 programs: 24%		
Homework 50%	3 problem sets: 18%		
	1 peer-reviewed exercise: 8%		
	Test 1: 10%		
Tests 50%	Test 2: 15%		
	Final exam: 25%		

10

NC STATE UNIVERSITY

Homework

- 4 programs
- 3 problem sets*
- 1 peer-reviewed madeup problem



*One of the problem sets will be dual submission.

CSC/ECE 506: Architecture of Parallel Computers

Tests

- Two 120-minute midterm tests (10%, 15% of grade)
- 150-minute final (24% of grade)
- Open notes
- No books, computers or communication devices



Extra Credit

- All activities for which extra credit is given must *help other students* to learn the course material.
- Examples
 - Making outstanding contributions to answering other students' questions on <u>Piazza</u>
 - Contributing useful practice problems via <u>Peerwise</u>
 - Doing extra peer reviews of madeup problems submitted to Expertiza
 - Suggesting Web or print resources that will help other write useful madeup problems



Outline for Lecture 1

- Architectural trends
- Types of parallelism
- Flynn taxonomy
- Scope of CSC/ECE 506

Key Points

- More and more components can be integrated on a single chip
- Speed of integration tracks Moore's law, doubling every 18– 24 months.
- *Exercise:* Look up how the number of transistors per chip has changed, esp. since 2006. Submit <u>here</u>.
- Until recently, performance tracked speed of integration
- At the architectural level, two techniques facilitated this:
 - Cache memory
 - Instruction-level parallelism
- Performance gain from uniprocessor system was high enough that multiprocessor systems were not viable for most uses.

Illustration

- 100-processor system with perfect speedup
- Compared to a single processor system
 - Year 1: 100x faster
 - Year 2: 62.5x faster
 - Year 3: 39x faster
 - ...
 - Year 10: 0.9x faster
- Single-processor performance catches up in just a few years!
- Even worse
 - It takes longer to develop a multiprocessor system
 - Low volume means prices must be very high
 - High prices delay adoption
 - Perfect speedup is unattainable

How did uniprocessor performance grow so fast?

- ≈ half from circuit improvement (smaller transistors, faster clock, etc.)
- ≈ half from architecture/organization:
- Instruction-level parallelism (ILP)
 - Pipelining: RISC, CISC with RISC back-end
 - Superscalar
 - Out-of-order execution
- Memory hierarchy (caches)
 - Exploit spatial and temporal locality
 - Multiple cache levels

But uniprocessor perf. growth has stalled

- Source of performance growth had been ILP
 - Parallel execution of independent instructions from a single thread
- But ILP improvement has slowed abruptly
 - Memory wall: Processor speed grows at 55%/year, memory speed grows at 7% per year
 - ILP wall: achieving higher ILP requires quadratically increasing complexity (and power)
- Power efficiency
- Thermal packaging limit vs. cost

Types of parallelism

- Instruction level (cf. ECE 563)
 - Pipelining

A (a load)	IF	ID	EX	МЕМ	WB			
В		IF	ID	EX	MEM	WB		
С			IF	ID	EX	МЕМ	WB	

Types of parallelism, cont.

- Superscalar/VLIW
- Original:

LD	F0,	34 (R2)
ADDD	F4,	F0, F2
LD	F7,	45 (R3)
ADDD	F8,	F7, F6

• Schedule as:

LD	F0,	34 (R2)	LD	F7,	45 (I	R3)
ADDD	F4,	F0, F2	ADDD	F8,	F0,	F6

+ Moderate degree of parallelism- Requires fast communication (register level)

Why ILP is slowing

- Number of pipeline stages is already deep (≈ 20–30 stages)
 - But critical dependence loops do not change
 - Memory latency requires more clock cycles to satisfy
- Branch-prediction accuracy is already > 90%
 - Hard to improve it even more
- Cache size
 - Effective, but also shows diminishing returns
 - In general, size must be doubled to reduce miss rate by half.

Current trends: multicore and manycore

Aspect	Intel Clovertown	AMD Barcelona	IBM Cell
# cores	4	4	8+1
Clock frequency	2.66 GHz	2.3 GHz	3.2 GHz
Core type	000 Superscalar	000 Superscalar	2-issue SIMD
Caches	2x4MB L2	512KB L2 (private), 2MB L3 (sh'd)	256KB local store
Chip power	120 watts	95 watts	100 watts

Exercise: Browse the Web (or the textbook ③) for information on more recent processors, and for each processor, fill out <u>this form</u>. (You can <u>view</u> the submissions.)

NC STATE UNIVERSITY

Scope of CSC/ECE 506

- Parallelism
 - Loop-level and task-level parallelism
- Flynn taxonomy
 - SIMD (vector architecture)
 - MIMD
 - Shared memory machines (SMP and DSM)
 - Clusters
- Programming Model
 - Shared memory
 - Message-passing
 - Hybrid
 - Data parallel

Loop-level parallelism

• Sometimes each iteration can be performed independently.

```
for (i=0; i<8; i++)
a[i] = b[i] + c[i];
```

 Sometimes iterations cannot be performed independently ⇒ no loop-level parallelism.

for (i=0; i<8; i++)
 a[i] = b[i] + a[i-1];</pre>

- + Very high parallelism > 1K
- + Often easy to achieve load balance
- Some loops are not parallel
- Some apps do not have many loops

Task-level parallelism

- Arbitrary code segments in a single program
- Across loops:

```
...
for (i=0; i<n; i++)
   sum = sum + a[i];
for (i=0; i<n; i++)
   prod = prod * a[i];
...</pre>
```

• Subroutines:

Cost = getCost(); A = computeSum(); B = A + Cost;

- Threads: e.g., editor: GUI, printing, parsing
- + Larger granularity \Rightarrow low overheads, communication
- Low degree of parallelism
- Hard to balance

Program-level parallelism

- Various independent programs execute together
- gmake:
 - gcc -c code1.c
 - gcc -c code2.c
 - gcc -c main.c

// assign to proc1

- // assign to proc2
- // assign to proc3
- gcc main.o code1.o code2.o
- + No communication
- Hard to balance
- Few opportunities

Scope of CSC/ECE 506

- Parallelism
 - Loop-level and task-level parallelism
- Flynn taxonomy
 - SIMD (vector architecture)
 - MIMD
 - Shared-memory machines (SMP and DSM)
 - Clusters
- Programming Model
 - Shared memory
 - Message-passing
 - Hybrid

NC STATE UNIVERSITY

Data parallel

Taxonomy of parallel computers

The Flynn taxonomy

- Single or multiple instruction streams.
- Single or multiple data streams.
- 1. SISD machine
 - Only one instruction fetch stream
 - Some not-too-ancient laptops or desktops



SIMD

- Examples: Vector processors, SIMD extensions (MMX), GPUs
- A single instruction operates on multiple data items.



CSC/ECE 506: Architecture of Parallel Computers

NC STATE UNIVERSITY

MISD

- Example: CMU Warp
- Systolic arrays



CSC/ECE 506: Architecture of Parallel Computers

NC STATE UNIVERSITY

Systolic arrays (contd.)

Example: Systolic array for 1-D convolution

y(i) = w1 x(i) + w2 x(i + 1) + w3 x(i + 2) + w4 x(i + 3)



- Practical realizations (e.g. iWARP) use quite general processors
 - Enable variety of algorithms on same hardware
- But dedicated interconnect channels
 - Data transfer directly from register to register across channel
- Specialized, and same problems as SIMD
 - General-purpose systems work well for same algorithms (locality etc.)

NC STATE UNIVERSITY

MIMD

- Independent processors connected together to form a *multiprocessor* system.
- Physical organization
 - Determines which memory hierarchy level is shared
- Programming abstraction
 - Shared Memory:
 - on a chip: Chip Multiprocessor (CMP)
 - Interconnected by a bus: Symmetric multiprocessors (SMP)
 - Point-to-point interconnection: Distributed Shared Memory (DSM)
 - Distributed Memory:
 - Clusters, Grid

MIMD Physical Organization



NC STATE UNIVERSITY

Shared-cache architecture:

- CMP (or Simultaneous Multi-Threading)
- e.g.: Pentium 4 chip, IBM Power4 chip, Sun Niagara, Pentium D, etc.
- Implies shared-memory hardware

UMA (Uniform Memory Access) Shared Memory :

- Pentium Pro Quad, Sun Enterprise, etc.
- What interconnection network?
 - Bus
 - Multistage
 - Crossbar
 - etc.
- Implies shared-memory hardware

MIMD Physical Organization (2)



NUMA (Non-Uniform Memory Access) Shared Memory :

- SGI Origin, Altix, IBM p690, AMD Hammer-based system
- What interconnection network?
 - Crossbar
 - Mesh
 - Hypercube
 - etc.

NC STATE UNIVERSITY

Scope of CSC/ECE 506

Parallelism

- Loop-level and task-level parallelism

- Flynn taxonomy
 - MIMD
 - Shared memory machines (SMP and DSM)
- Programming Model
 - Shared memory
 - Message-passing
 - Hybrid
 - Data parallel

Programming models: shared memory

- Shared Memory / Shared Address Space:
 - Each processor can see the entire memory



 Programming model = thread programming in uniprocessor systems

Programming models: message-passing

- Distributed Memory / Message Passing / Multiple Address Space:
 - A processor can directly access only its local memory.
 - All communication happens by explicit messages.



Programming models: data parallel

- Programming model
 - Operations performed in parallel on each element of data structure
 - Logically single thread of control, performs sequential or parallel steps
 - Conceptually, a processor associated with each data element



NC STATE UNIVERSITY

Data parallel (cont.)

- Architectural model
 - Array of many simple, cheap processing elements (PEs) each with little memory
 - Processing elements don't sequence through instructions
 - PEs are attached to a control processor that issues instructions
 - Specialized and general communication, cheap global synchronization
- Original motivation
 - Matches simple differential equation solvers
 - Centralize high cost of instruction fetch/sequencing

Top 500 supercomputers

- <u>http://www.top500.org</u>
- Let's look at the Earth Simulator, #1 in 2004
- Hardware:
 - 5,120 (640 8-way nodes) 500 MHz NEC CPUs
 - 8 GFLOPS per CPU (41 TFLOPS total)
 - 30s TFLOPS sustained performance!
 - 10 TB total memory
- Now (Nov. 2023)
 - Frontier, at Oak Ridge National Laboratory, is #1
 - 8.7 million cores
 - 1194 PFLOP/s max performance (Rmax)
 - 1680 PFLOP/s peak performance (Rpeak)

Exploring the Top 500 list ...

- Lists > Top500 > November 2023 > The list
 See a list of the top systems
- Statistics > List Statistics > Vendors
 - Lenovo is top vendor, more than double HPE
- Statistics > List Statistics > Architecture
 - Clusters are overwhelmingly dominant
- Statistics > Developm't over Time > Countries
 - China comes from nowhere to lead in # of systems
 - But US still leads in performance share

Exercise

- Go to <u>http://www.top500.org</u> and look at the Lists and Statistics menus in the top menu bar.
- From the Statistics dropdown,
 - choose either List Statistics or Development over time,
 - then select one of the statistics, e.g., Vendors, Processor Architecture, and
 - examine what kind of systems are prevalent. Then do the same for earlier lists, and report on the trend.
- You can go all the way back to the first list from 1993.
- Submit your results here.