

Advanced Computer Architecture II: Multiprocessor Design

Introduction to Multiprocessor Design

Professor Russ Joseph
Department of Electrical Engineering and Computer Science
Northwestern University

January 3, 2007

ECE453

Welcome to Multiprocessor Design

- Introduction
- Course Mechanics
- Project Structure
- Why Focus on Parallelism
- Project Ideas
- Summary

Introduction

This course covers multiprocessor systems. What exactly does this mean?

Computer systems with more than one processor...still very general.

Focus on traditional MP systems:

- multiple general purpose CPUs
- some stand-alone memory
- some interconnection network

ECE453

2

Multiprocessor Landscape Is Huge

This restricted definition still covers a lot of ground:

- small scale multiprocessors (SMP)
- commercial multiprocessor systems (e.g. IBM SP2, SGI Origin)
- large supercomputer systems (e.g. IBM BlueGene)

Buckle your safety-belt, we are doing it all...

How This Course Relates To Others

ECE358 - Introduction to Parallel Computing

Focus: parallel programming techniques and practices

Overlap: In ECE453, we need to understand how parallel programs work.

Difference: We build and evaluate systems, not programs.

ECE452 - Advanced Computer Architecture I

Focus: advanced uniprocessor architecture, techniques for exploiting Instruction Level Parallelism (ILP)

Overlap: In ECE453, the processors become components.

Difference: We are primarily interested in multiprocessor systems and thread-level parallelism.

Topics Covered: A Little Less Usual

These topics will be covered via readings and group discussions:

- SMT and CMP architectures
- high performance I/O
- thread-level speculation
- power-aware design

Topics Covered: Usual Suspects

These topics will be covered in your standard lecture-like format:

- metrics and workloads
- parallel programming fundamentals
- snoop-based MP systems
- scalable MP systems
- synchronization
- interconnection networks

More Mechanics

Class Meetings

Days: Tuesdays and Thursdays

Time: 3:30PM-5:00PM

Location: Tech LG72

Class Webpage

<http://www.ece.northwestern.edu/~rjoseph/ece453>

Lectures, Discussions, and Readings

Lectures

- Use textbook as supplemental reading
- Covers basic information that people expect you to know

Discussions

- Cover “researchy” topics
- Rely on assigned reading of research papers
- Depend on good participation

Contact Info

Instructor: Professor Russ Joseph
Office: Tech L467
Email: rjoseph@ece.northwestern.edu
Phone: 1-3061
Office Hours: Friday 2:00PM-4:00PM

Grading

Breakdown	
Project	40% (10% + 30%)
Exams (2 in-class)	50%
Participation	10%
Homework	0%

Project: Multiprocessor Research

Phase 1 - Introduction to MP Architecture:

- out-of-the-box assignment
- straight-forward experiments
- mostly plug and chug
- short executive summary (1-2 pgs)

Phase 2 - Original MP Research:

- creative experiments
- paper (think workshop paper 6-8 pgs)
- presentation

Project: Keeping It Real

The time and resource constraints are significant.
Expectations will be scaled back.
Negative results are acceptable.
Quarter-long project for groups of 1-2 persons.
Regular checkpoints along the way.

Project: Ideas

We'll talk about projects during the next few lectures

Project: Simulation

We will use the M5 Simulator (from UMich).
This is a full system simulator (with support for networking).
It is publicly available, builds on x86/Linux without incident.
Check <http://m5.eecs.umich.edu/>

Why Build/Program Parallel Systems?

Parallel programs can be harder to implement and debug.
Multiprocessor systems can be more challenging to build and analyze.
If so, then why kill ourselves?

Answer: Performance, Performance, Performance

Why do we do anything in computer architecture?

Go Faster, man!!!

Example:

Problem of size n takes time t on uniprocessor.

- In ideal world, p -way multiprocessor solves problem in time t/p .
- In real world, probably not, but you already knew that.
- For many important problems the non-ideal speedup is worth the effort.

Parallel Speedup

The speedup of a parallel implementation over a serial one can be represented as:

$$S_p = \frac{T}{T_p} \quad (1)$$

- T - time for serial implementation
- T_p - time for parallel implementation

The efficiency of the implementation is given by:

$$E_p = \frac{S_p}{p} = \frac{T}{pT_p} \quad (2)$$

- If efficiency is 100%, then the speedup is linear.
- In most practical cases, the efficiency never reaches 100%.

Amdahl's Law

Amdahl's Law expresses the limits on performance improvement:

$$T_p = T\left(\alpha + \frac{1-\alpha}{p}\right) \quad (3)$$

- α - the fraction of the implementation that cannot be parallelized (serial portion)
- Speedup of a parallel implementation is limited by the fraction of time that parallelism cannot be exploited.

Answer: More On Performance

Parallel architectures offer real hope for solving large problems.

- gigabytes of data
- gigaflops of computation

It's not just about finishing the job quickly:

- more variables allow better *quality* results
- need more computing power to handle *larger* problem size

Answer: The Price Is Right

Parallel processing has become relatively “affordable”:

- 1 CRAY T90 Supercomputer - 2 GFlops (\$2,500,000)
- 8 Node IBM SP2 - 2 GFlops (\$400,000)
- 8 Node SGI Origin 2000 - 10 GFlops (\$200,000)
- There are even cheaper Linux-cluster based approaches (e.g. Beowulf) if they meet your performance criteria

(Performance/Price quotes courtesy of Prof. Gokhan Memik)

Answer: Many Other Possibilities

Energy-Efficiency: Use extra resources to make computation efficient

Availability: Use extra resources to provide failure tolerance

Security: Use extra resources for protection

Any others?

Answer: Better Integration

Components “play well with others”.

Virtually all high-performance processors have builtin SMP support

Next wave of innovation: chip multiprocessors (CMPs)

Other Kinds of Parallelism

The focus in this class is on thread-level parallelism exploited by multiprocessors.

But there are many other forms of parallelism (e.g. bit-level, instruction-level, etc.)

We will not talk at length about some of the other alternative architectural approaches for exploiting parallelism (e.g. vector architectures, SIMD ISA extensions, etc.)

On-Chip Parallelism: The Way of The Future

Sure, large scale supercomputers are not going away anytime soon.

But on-chip (thread) parallelism will be an increasingly important player.

In fact, large scale parallel computers will be built from high-performance microprocessors which will have lots of thread-level parallelism.

(Fine. This wasn't really a good segway, but we need to talk at least briefly about on-chip parallelism and this seemed as good a place as any...)

Two Major Ideas: CMP and SMT

Perhaps the two most prominent ways to increase thread level parallelism are simultaneous multithreading (SMT) and chip multiprocessing (CMP).

All the major industry players have multicore and/or multithreaded processors already (Intel, AMD, IBM, SUN, ARM???)

What are the similarities/differences?

Why This Is Inevitable

Moore's Law will continue to give us an abundance of transistors (see ITRS).

What can we do with them?

Could build bigger conventional processors, but there is diminishing return and exponential design complexity, and power doesn't play nice.

Could add more cache, but this doesn't help computation bound applications. Also, may have to deal with hit rate vs access latency tradeoffs.

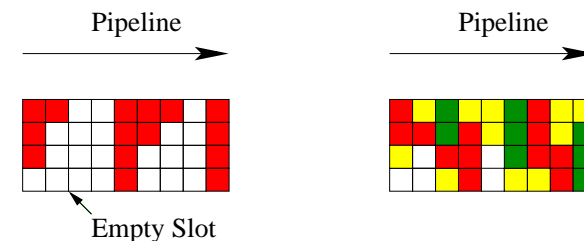
Could put more processors on a chip (e.g. CMPs)

Thread-Level parallelism sounds like the way to go...

SMT in A Nutshell

Threads have their own registers and PC, but share the rest of the pipeline (caches, execution resources, issue slots, etc.)

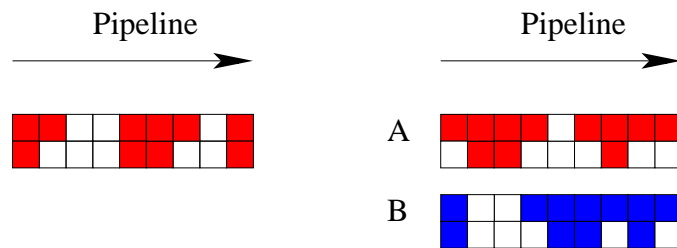
The processor takes turns fetching/issuing instructions from different threads.



CMP in A Nutshell

Threads run on separate pipelines and hence have private execution resources, but may share some less frequently accessed resources (e.g. L2 or L3 cache)

Each processor takes fetches/issues instructions from a different thread.



Project Ideas

Projects should draw on your own background and research interests.

Don't be afraid to take risks:

- negative results are OK
- opportunity to think "outside-the-box"

Just make sure that you can make tangible progress in a few months

Have Your Cake and Eat It, Too

There's no reason why you can't have SMT and CMP at the same time.

IBM Power5 is a dual core multithreaded processor.

You get the best of both worlds.

This will become the norm for desktop, workstation, and server environments.

Summary

Presented a quick sketch of the course.

Multiprocessor research is a rich and exciting area.

Next Time: Fundamentals of Multiprocessor Architecture

Skim: Chap 1, Read: 1.2, 1.3

Read: 4.1, 4.2, 4.4