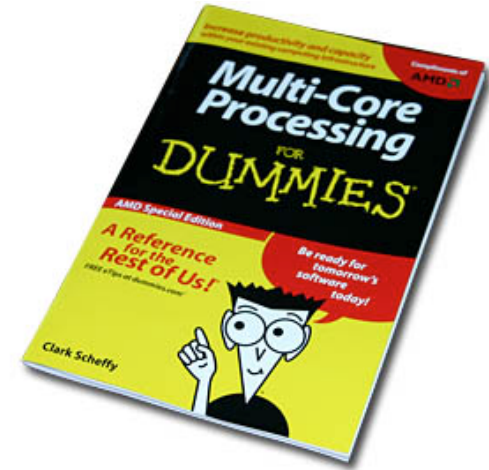




Lecture 1 The Multicore Revolution



John Cavazos

Dept of Computer & Information Sciences

University of Delaware

www.cis.udel.edu/~cavazos/cisc879

CISC 879 : Software Support for Multicore Architectures



- John Cavazos <cavazos@cis>
- Assistant Professor, CIS
- Research
 - Applying machine learning to hard systems problems
 - Compiling for advanced architectures
 - Interplay between compiler and architecture
 - Nitty-gritty things that happen in compiler back-ends
 - Rethinking fundamental structure of optimizing compilers

Interested in starting new research involving Multicores!

CISC 879 : Software Support for Multicore Architectures



Lecture 1: Overview

- Structure of Course
- Motivation
- A Brief Overview of the Landscape
- The Rest of the Course
- Administrivia



Structure of the Course

- First 14 lectures by myself and my research group
- Next N lectures are done by you!
 - Student presentations (20 mins.)
 - Check papers online and sign up soon!



Structure of the Course

- Projects
- Two projects (next slide)
 - Team projects (2 or 3 per team)
 - Project reports due for Project 1 and 2
 - Amount of work proportional to size of team
 - Presentation due for Project 2



Project 1: Empirical Study

- Choose two programs
 - Convert 1 Sequential Kernel to OpenCL or OpenACC
 - Profile 1 large Application
 - Compile, Run, Profile
- Goal : Familiarization with OpenCL or OpenACC and profiling
- Deliverable: Project Report
 - ~2 pages per team member
 - Template available online (font size, margins, etc.)
- Handout available soon



Project 2: Parallelization

- Choose two real-work applications
 - Bioinformatics, Data mining, Financial applications, SPEC benchmarks
- Parallelize some *important* loops
- Deliverable: Report (~5 pgs per team member)
 - Conference paper format
 - Project resentation (~30 mins)
- Handout available in a couple weeks



Multicores Available

4 nVidia GT200 (Tesla C1060) GPU cards

Access Control via SLURM (Simple Linux Utility for Resource Management)

```
srun -N1 --gres=gpu:G <command>
```

provide exclusive access to GPU

To compile a simple example:

```
g++ -L/software/cuda-sdk/shared/lib/linux/  
-lOpenCL -I/software/cuda/include/  
VectorAddition.cpp
```

```
srun -N1 --gres=gpu:1 a.out
```




Basis for Grading

- Paper presentation (30%)
- Projects (70%)
 - Project 1 (30%)
 - Checkpoint
 - Project report
 - Project 2 (40%)
 - Checkpoint
 - presentation
 - Project report

No Midterm or Final!

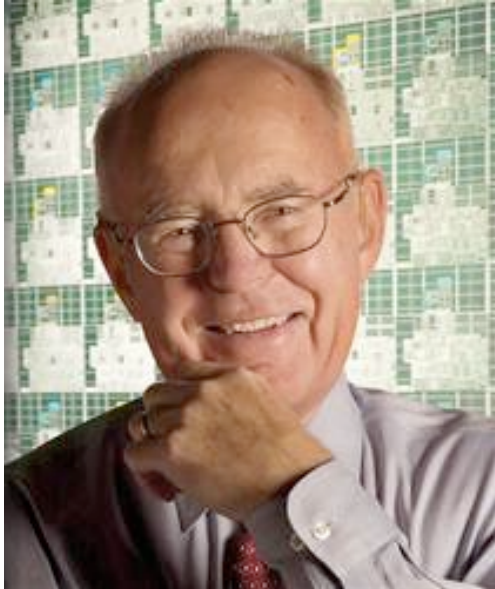


Lecture 1: Overview

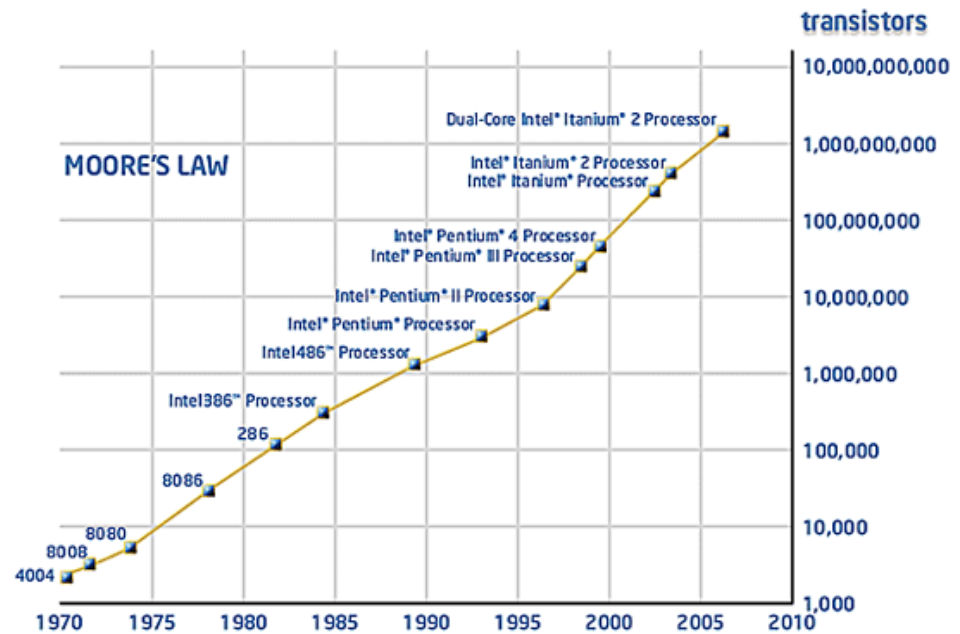
- Structure of Course
- **Motivation**
- A Brief Overview of the Landscape
- The Rest of the Course
- Administtrivia



Technology Trends



Gordon Moore (Intel co-founder) predicted in 1965 that transistor density of semiconductor chips would double every 18 months.

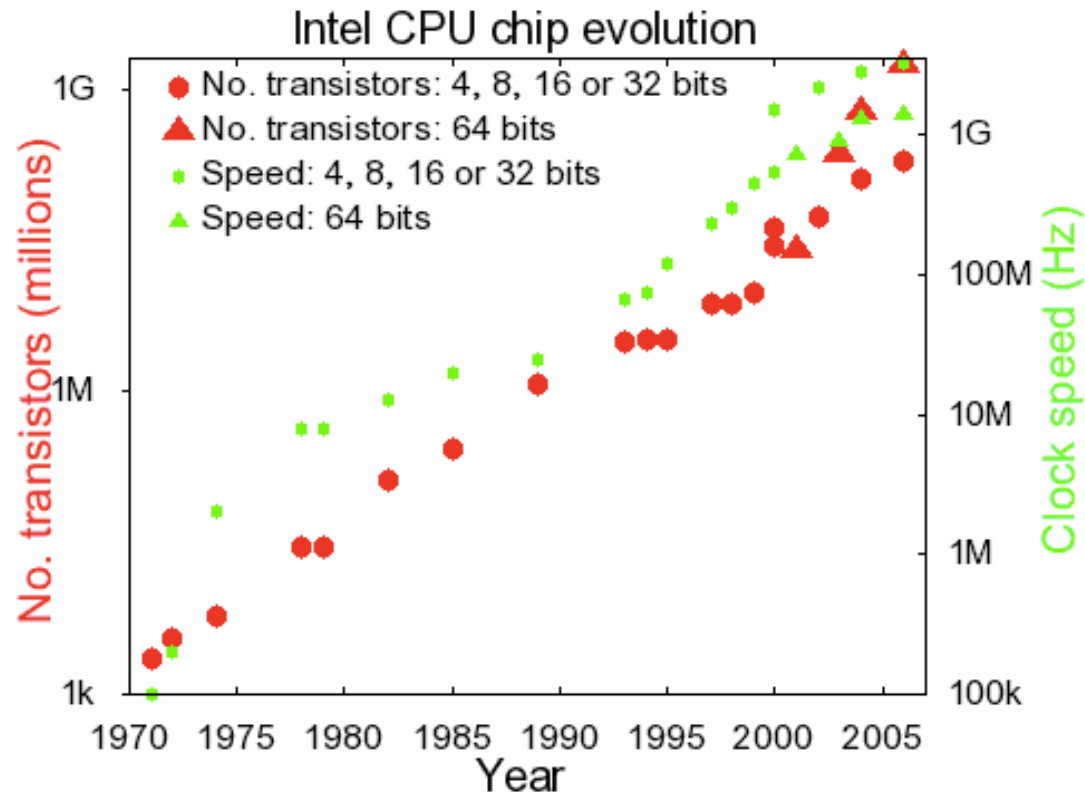


Moore's Law

Slide Source: Jack Dongarra



Transistors and Clock Rate

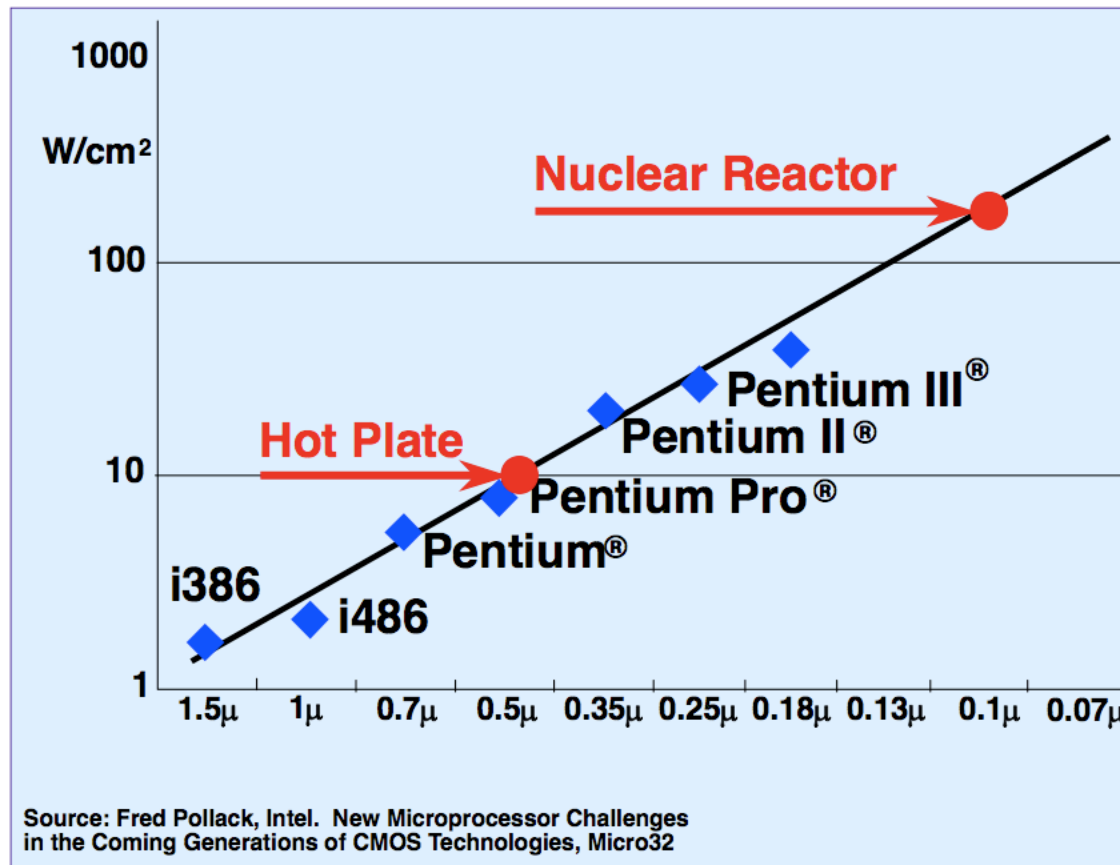


So, processors got faster every 18 months (roughly 50% to 100%!)
Why bother with parallel programming? Just wait a year or two...

Slide Source: <http://audilab.bmed.mcgill.ca/~funnell/Bacon/HW/trends.html>



The “Power Wall”



- High Power Consumption and Heat Dissipation



Multicores Save Power

- Multicores with simple cores decreases frequency and power
- Example : Processor with power budget N
 - Increase frequency by 20%
 - Substantially increases power, by more than 50%
 - But, only increase performance by 13%
 - Decrease frequency by 20% (e.g., simplifying core)
 - Decreases power by 50%
 - Simplifying core means we can now add another simple core
 - Power budget stays at N with increased performance!



Architects Passed the Ball

- “We are dedicating **all of our future product development to multicore designs**. ... This is a sea change in computing”
- Paul Otellini, President, Intel (2005)
- “**Multicore**: This is the one which will have the biggest impact on us. **We have never had a problem to solve like this**. A breakthrough is needed in how applications are done on multicore devices.”
- Bill Gates, Microsoft
- “When we start talking about parallelism and ease of use of truly parallel computers, we're talking about a problem that's **as hard as any that computer science has faced**. ... I would be panicked if I were in industry.”
- John Hennessy, President of Stanford
ACM Queue Magazine, 1/07



Looming “Multicore Crisis”

Old Conventional Wisdom	New Conventional Wisdom
Increasing clock frequency is primary method of performance improvement	Processors Parallelism is primary method of performance improvement
Don't bother parallelizing app, just wait and run on much faster sequential computer	No one is building 1 processor per chip End of La-Z-Boy Programming Era
Less than linear scaling for a multiprocessor is failure	Given the switch to parallel hardware, even sub-linear speedups are beneficial

Slide Source: Berkeley View of Landscape

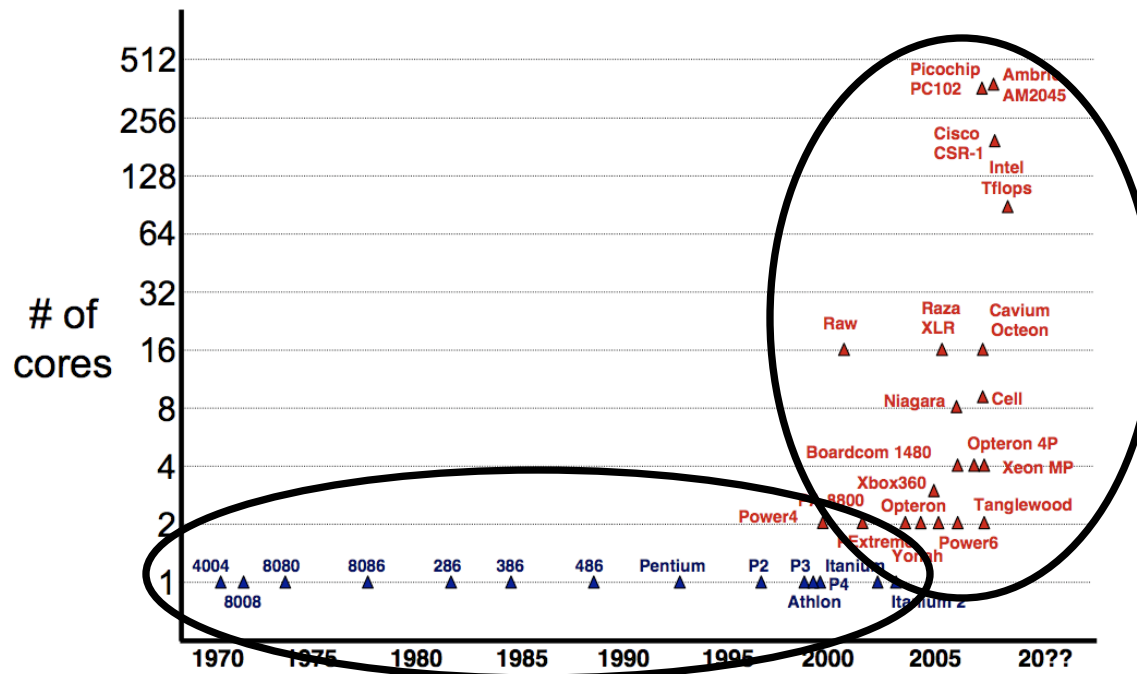


Lecture 1: Overview

- Structure of Course
- Motivation
- **A Brief Overview of the Landscape**
- The Rest of the Course
- Administrivia



The Rise of Multicores



Around the same time:

- Cell B.E. released with 8 cores
- Nvidia Graphics Processing Unit (GPU) has 128 cores
- Intel demonstrates an 80-core research chip

Slide Source: Amarasinghe, 6189 IAP 2007

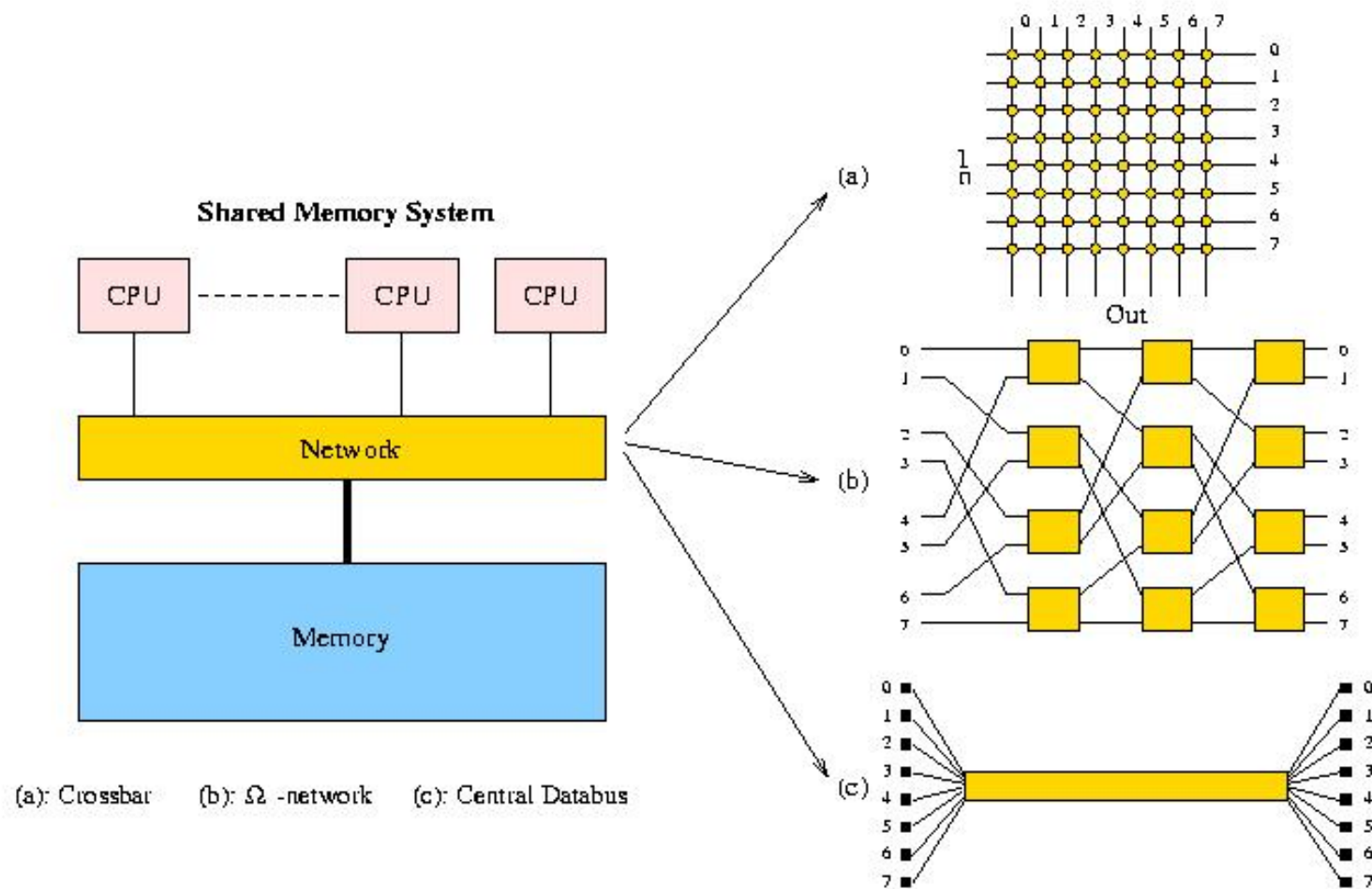


Classes of Multicores

- Shared-Memory
 - Intel “Clovertown”
 - AMD “Barcelona”
 - Sun Niagara
- Distributed-Memory
 - MIT Raw
 - Cell
- Minicores
 - Intel Teraflops
 - GPUs



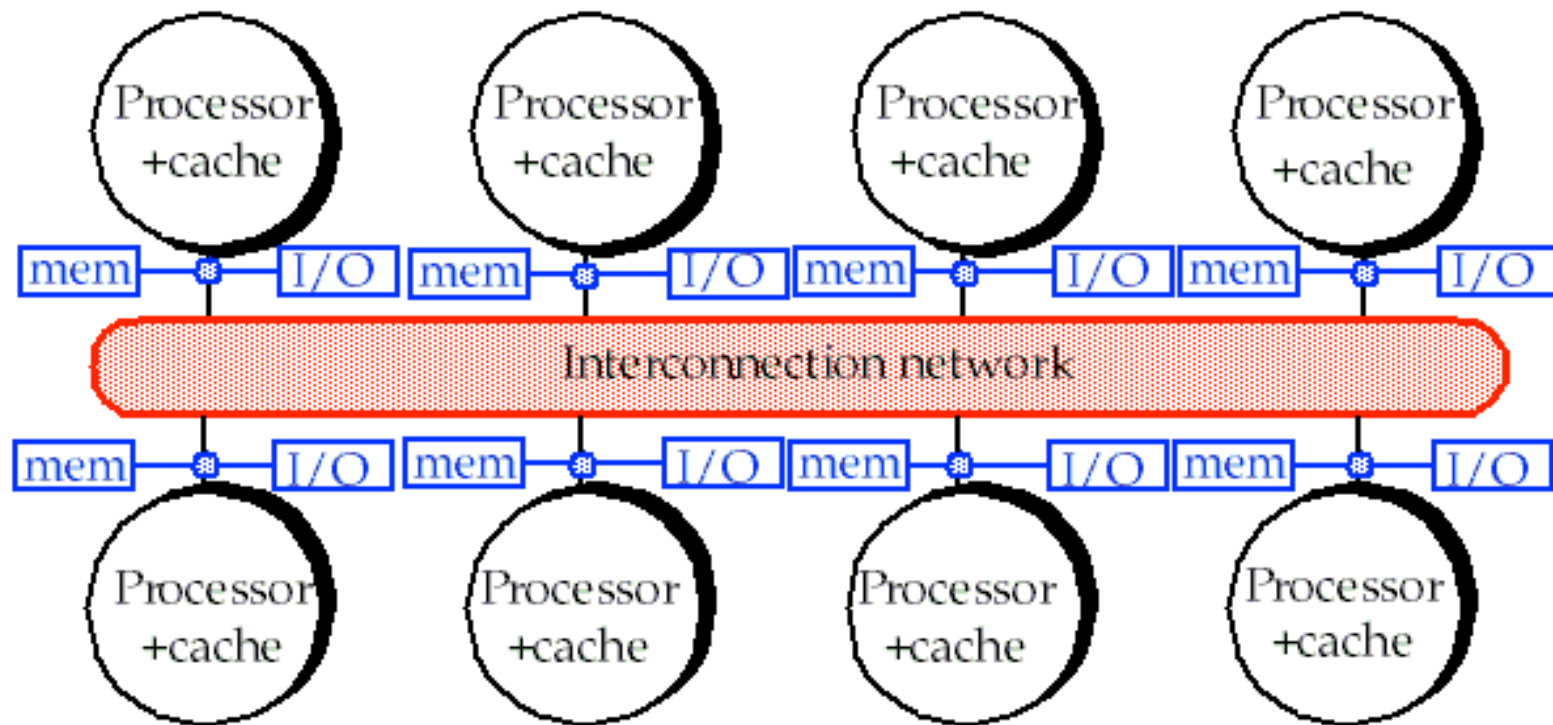
Shared-Memory Multicores



Slide Source: <http://www.cosc.brocku.ca/Offerings/3P93/KimFrith/3P99.html>



Distributed-Memory Multicores



Slide Source: <http://www.csee.umbc.edu/~plusquel/611/index.html>



Lecture 1: Overview

- Structure of Course
- Motivation
- A Brief Overview of the Landscape
- The Rest of the Course
- **Administrivia**



Background/References

- Parallel programming not required
- Computer architecture not required
- Must be familiar with the C programming language
- No textbook required
 - There are several references, see course website



Project Guidelines

- Papers should be
 - Well-written and formatted correctly
 - Properly referenced
 - Results should be presented with graphs
 - Intellectual merit most important factor
- Negative result is fine
 - However, must demonstrate something interesting

Think of this as a conference submission!



Expectations

- Class participation
- Ask questions
- Challenge all speakers.
- NOT a lecture class or a passive experience.
ACTIVE learning.
- Most common project problem: Not getting started
- *Ask for help if you need it!*
 - I will hold office hours Saxby's on Amstel Ave.
 - Email *first* me whenever you want an appointment.
 - Require checkpoints to show me status!