CSE 260 – Introduction to Parallel Computation

Larry Carter
carter@cs.ucsd.edu

Office Hours: AP&M 4101
MW 10:00-11 or by appointment

Topics

	Instances	Principles	Theory
Hardware	specific machines	parallelism, pipelining, ...	limits to performance
Languages	...		
Applications			
Algorithms			
Systems			

Emphasis of course

- Scientific computation
  - I'll mostly ignore commercial computing
    (even though it's important)
- Supercomputers and supercomputing
- Applications
- Focus on topics of practical significance

This is a graduate course ...

- If you want to shift emphasis, let's talk!
  - E.g. you might want to do a different project
- Surprise me!
  - Do something extra occasionally (write mini-studies)
- Question authority!
  - Questions and discussions are encouraged
  - I'm opinionated and sometimes wrong
- Listen to your classmates!
  - They have insights from different perspectives

Syllabus

- Weeks 1-4: Whirlwind overview (20%)
  - Learn vocabulary used in field
  - Build mental "filing cabinet" for organizing later topics
  - Three quizzlets
- Weeks 5-10: Selected topics in depth (25%)
  - Important and/or interesting papers
  - Give presentation, or (perhaps) write critique
- Project: One application in various languages (35%)
  - All on same computer (SUN Ultra at SDSC)
- Mini-projects (20%)
  - 5-minute report supplementing class material

Vocabulary (1st three weeks)

- Know terms that are underlined*
  - Should be in your passive vocabulary (i.e. if someone uses the term, you should have a reasonably good idea what it means)
  - Not necessarily in your active vocabulary (i.e., I don't expect you to be able to list all the terms).

Quizzlets* will be multiple choice or fill-in-the-blanks, not essays.

* unfortunately, PowerPoint may underline misspellings.

Any Administrative Questions?

Class 1: Parallel Architectures

Interesting reading:

- Chapter 9 of Patterson & Hennessy's undergrad text (second edition). [Or Chapter 10 of Hennessy & Patterson's graduate text].

Parallel computer:

- Almasi + Gottlieb: "a large collection of processing elements that can communicate and cooperate to solve large problems fast."
- Many "processing elements" cooperating on a single problem.
- Multiprocessor server? Not large enough. Networks of workstations and internet-connected computers when working together? Sure .

Supercomputer:

A computer costing $3,000,000 or more.

Why bother?

Gene Amdahl - 1967(!): "For over a decade, prophets have voiced the contention that the organization of a single computer has reached its limit."

- he went on to argue that the single processor approach to large-scale computing was still viable.

Parallel computing is expensive

- Higher cost per cycle
- Greater programming effort
- Less convenient access

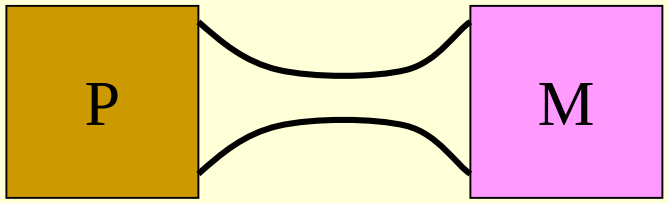
Possible answers ...

- Answers today are more valuable than answers tomorrow
  - weather prediction
  - conference submissions
  - product design (airplanes, drugs, ...)
- Some problem requires huge memories
  - Once you have a huge memory, it's more economical to have multiple processors.

notice the underline

Von Neumann Bottleneck

(term introduced by John Backus in 1978, referring to design described by John Von Neumann in 1945)



"The instruction stream is inherently sequential – there is one processing site and all instructions, operands and results must flow through a bottleneck between processors and memory."

The goal of parallel computers is to overcome the Von Neumann Bottleneck.

Flynn's Taxonomy

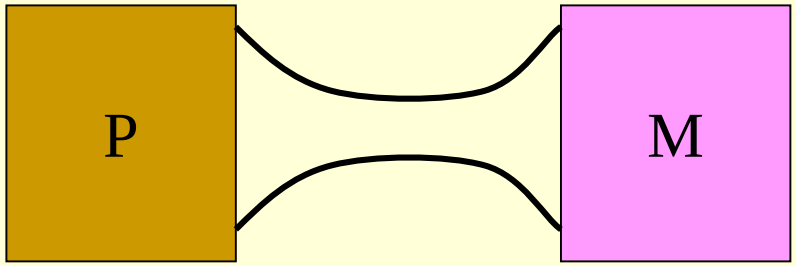
- Flynn (1966) classified machines by data and control streams

notice the underline

Single Instruction Single Data (SISD)	Single Instruction Multiple Data SIMD
Multiple Instruction Single Data (MISD)	Multiple Instruction Multiple Data MIMD

SISD

- SISD
  - Model of serial von Neumann machine
  - Logically, single control processor
  - Includes some supercomputers, such as the 1963 CDC 6600 (perhaps the first supercomputer)



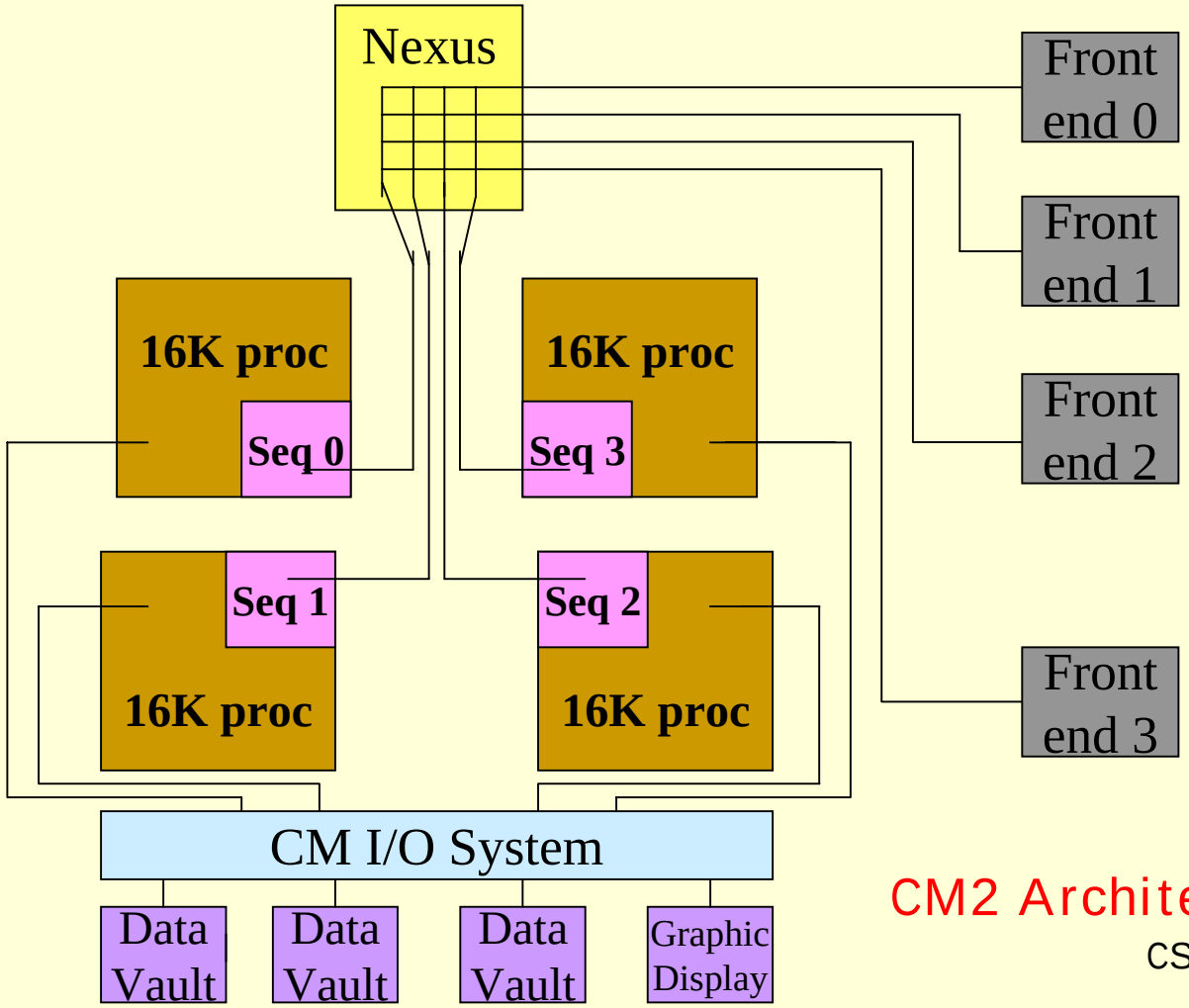
SIMD

- Multiple processors executing the same program in *lockstep*
- Data that each processor sees may be different
- Single control processor
- Individual processors can be turned on/off at each cycle ("masking")
- Examples: Illiac IV, Thinking Machines' CM-2, Maspar, DAP, Goodyear MPP,

The ill-fated Illiac IV

- Project started in 1965, predicted to cost $8M and provide 1000 MFLOP/S.
- Delivered to NASA Ames in 1972, cost $31M, ran first application in 1976, performed 15 MFLOP/S.
- 64 processors, 13-MHz clock, 1MB memory
  - Meanwhile, the CDC 7600 (basically a superscalar uniprocessor), was 36 MHz, 36 MFLOP/S, 3.75 MB memory, $5.1M, and running in 1969.

- CM2 (1990, built by Thinking Machines Corp) had 8,192 to 65,536 one-bit processors, plus one floating-point unit per 64(?) procs.
- Data Vault provides peripheral mass storage
- Single program - all unmasked operations happened in parallel.



CM2 Architecture

Vector Computers

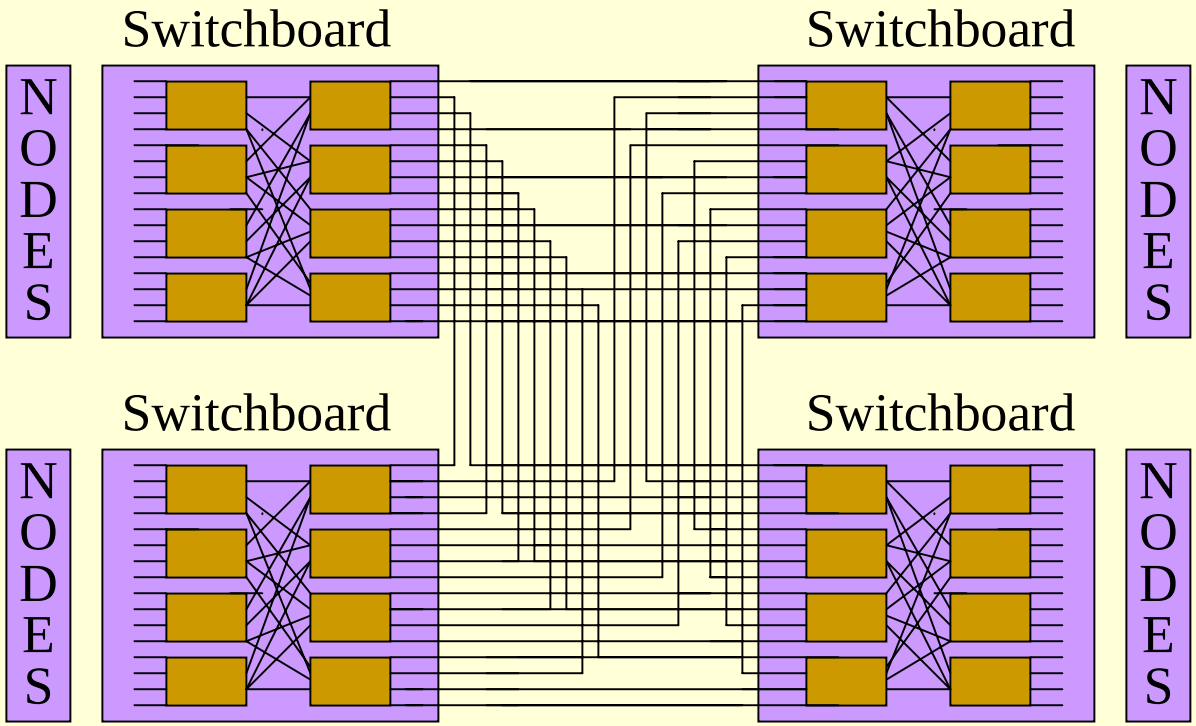
- Hybrid SISD/SIMD – has ordinary "scalar" operations and "vector operations", which operate on up to (say) 256 independent sets of operands (utilizing "vector registers") in fast pipeline mode.
- Examples: Cray supercomputers (X-MP, Y-MP, C90, T90, SV1, ...), Fujitsu (VPPxxx), NEC, Hitachi.
  - many of these have multiple vector processors, but typically the separate processors are used for separate jobs.
- 4 or 8-way SIMD also used in graphics & multimedia accelerators, video game machines.

MIMD

- All processors execute their own set of instructions
- Processors operate on separate data streams
- May have separate clocks
- IBM SP's, TMC's CM-5, Cray T3D & T3E, SGI Origin, Tera MTA, Clusters, etc.

SP2

- High Performance Switch of 64 node SP2
- Multiple paths between any two nodes
- Network scales with added nodes



Some more MIMD computers

- Cluster: computers connected over high-bandwidth local area network (usually Ethernet or Myrinet), used as a parallel computer.
- NOW (Network Of Workstations): homogeneous cluster (all computers on network are same model).
- "The Grid": computers connected over wide area network

Larry's conjecture

- SIMD is used on early machines in a given generation; it gives way to MIMD.
  - When space is scarce, you can save by having only one control unit
  - As components shrink and memory becomes cheaper, the flexibility of MIMD prevails

(Conceivable mini-project: find evidence for or against Larry's conjecture)

What about MISD?

- Multiple Instruction Single Data
- The term isn't used (except when discussing the Flynn taxonomy) .
- Perhaps applies to pipelined computation, e.g. sonar data passing through sequence of special-purpose signal processors.