Break:

Survey; Projects in 533 after class

Classes to take:

Kahan's Class: 279 Fall

Parallel Processing sequence

268 (spring 91), Hands on, here and now

257 (Fall 91), Advanced languages and architectures

283 (fall 90) Foundations of parallel processing

Performance Analysis

26x

Review:

Review course and the future, in the end pound my pulpit about the future

Chapter 1:

Resposibilities of a computer architecture: functional requrements, balancing HW & SW, designing to last through trends

Quatitative Principles:

Common case fast, Amdahl's Law, Locality of reference

Architect => build computers as well as instruction sets

•••10 years from now at a conference => AMdahls Law

Chapter 2:

Accurate performance evaluation; everything but execution time can be misleading (if not intuitive)

•••Basic formula: **IC*CPI*CCT**

Cost as half of the equation of cost performance

Semiconductor cost model; Price vs cost

Chapters 3 & 4:

Princples of ISA & 3 classic examples; leading examples of mainframe, minicomputer, microcomputer + machine of future

Frequency of occurrence and role of ISA with compilers

All future work **must** be in conjunction with highly optimizing compilers

••• Rough idea of instruction mixes in RISC machines

25%-30% loads,

Ch 5

Second act of a play; difficult to be execiting, set up for final act

why CPI of microprogram is 3 to 6

difficulty of interrupts

Ch 6

•••Magic of pipelining, hazards that prevent the magic from occuring

•••Be able to understand stutural hazards, control hazards, RAW, WAR, and WAW hazards

HW & SW techniques to get more performance from pipelining

(look up tomasulo, scorebaord, trace sceduling, SW pipelining)

Ch 7

Why vector have such brittle performance characteristics

•••What is ISA and what is reason=> bandwidth of FPU

Understand why peak can be so misleading for these machines

Ch 8

••• Memory systems may be real heros of computer architecuture, princple of locality

Inherent simlatiry of levels of the memory hierarchy

All characterized by 4 questios

Where can a block be placed in upper level? (placement)

How is a block found in upper level? (identifiation)

Which block is replaced on a miss? (replacement)

What happens on a write? (write strategy)

••• Single most important pitfall: too small an address space

Ch 9

The orphan of computer architecture, I/O

Evaluating an I/O system

BW vs Latency, econmomic arguement for response time

••• Remember that computer without I/O is not interesting, and is surely the bottleneck as processors increase in performance

1. *Single instruction stream, single data stream* (SISD, the uniprocessor)
2. *Single instruction stream, multiple data streams* (SIMD)
3. *Multiple instruction streams, single data stream* (MISD)
4. *Multiple instruction streams, multiple data streams* (MIMD)

SIMD vs. Vector?

ISA =

N paralell streams & adders vs pipeline

(Array processor)

Single PC (a “uniprocessor”)

Best case: data parallelism, if not lots of data, nothing to do

Worst case: case statement, only singe data item

MISD

Special Purpose Architectures => SW+HW in single chip

100X improvement in cost performance if enough volume

(Speech likely custom chip vs CM2)

MIMD: why so attractive? what is stumbling block?

Naming confusion
Multiprocessor vs multicomputer
Scalalbe? Programmable?
MIMD great for timesharing
Suppose cannot do timesharing; what applications?

Why hard? Communication as number of people goes up
Efficiency and need of understanding
Everything speedup to achieve great speedup

Example page 576 shows % sequential & get speedup
=> new algortihms

Figure 10.1 p 577: which road to El Dorado?
Low road: Programmers, from left to right
High road: hardware designers replication of processor & switch

False dichotomy: shared vs distributed memory

shared => single address space
distributed => physically separate modules
Figure 10.2 on page 578

Which is best?
Communication:
Explicit vs. Implicit
Explicit: programming harder
vs efficiency by multiple ports
Overal communication and computation, sends vs fetches
Pointers vs data

The bet!
The second Bell prediction concerns the number of data streams in supercomputers shipped in 1995. Danny Hillis believes that while supercomputers with a small number of data streams may be best sellers, the biggest machines will be machines with many data steams, and these will perform the bulk of the computations. Bell bet Hillis that in the last quarter of calendar year 1995 more

sustained MFLOPS will be shipped in machines using few data streams (<100) rather than many data streams (>1000). This bet concerns only supercomputers, defined as machines costing more than $1,000,000 and used for scientific applications. Sustained MFLOPS is defined for this bet as the number of floating-point operations per **month**, so availability of machines affects their rating. The loser must write and publish an article explaining why his prediction failed; your authors will act as judge and jury.

Real questions:
(1) Do problems exist with sufficient parallelism?
(2) Can people/compilers be trained to exploit parallelism?

Future Directions in Compilers
Data Layout to reduce memory hierarchy latencies and communication in MP
Faster cycle time increases importance
Larger fraction of time in memory latency
For 10 years compilers were built for machines with registers that used them only for temps, no allocation
Can't be done today; similarily of data layout in future

Exploitation of parallelism:
Express calaculation as Array, up to compiler to layout data so as to minimize communication and write the loops
Figure 10.4 page 582 multiplicative combination

Future Directions in Architecture
Figure 10.5 page 587

Advice:

Not talking about academic subjects; talking about your future!

Prediction of 64Mb DRAM in 1995; 64Mb per chip => 8MB/Chip cost
First year students before receive your PhD
by 1997 will be $25=> 64M transistors
Before complete first postdoc
Reasonable processor in 150,000 transistors => 0.22%
Cost? $1 to have a 30 MIPS processor in every DRAM?

April 31, 1990 front page, lead story of NYTimes

Article that Japan's focus after 5G; SW for massive parallelism, not hardware for massive parallelism

Difficult time for the field: Academics filling up, Industry shrinking
Hitorically introspective: software means building better compilers and operating systems; hardware means building better CPUs
Time to reach out! CS

AAA! Applications, Algorithms, and Architecture=> Achieving 1000X improvement in performance givenr 1000X peak potential increase

Who will be heros and heroines of the 1990s?
It will be those who can deliver that achieve 1000X performace in programs given 1000X imrprovement in performance

What do you need to know?

- Applications
- Algortithms
- Architecture

You already have one A; get the others a lead the way to the next century in computer science