

Lecture1: Introduction

1

Administrative info

Welcome to ECE669! Welcome off-campus students!

Csaba Andras Moritz, Associate Professor @ ECE/UMASS

Questions/discussions/email questions are welcome!

My Focus: Design of Parallel Computer Systems

- Chips to supercomputers
- Software/hardware tradeoffs
- Compilers and Programming Models

- Also: give an idea about state-of-the-art research in the area by discussing research paper

Info: www.ecs.umass.edu/ece/andras/courses/ECE669

2

Administrative info contd.

2(or 3) Homeworks (info will be available online):

- Cache simulators and network simulators (run on Sun machines)
- Experiments with analytical model based performance estimation

1 Research Project

- Speculative research on some aspect of parallel computing
- Two ideas/options are provided
 - Fine-grained synchronization support
 - Wireless connectivity on tiled systems on a chip
- Feel free to define your own project (out-of-the-box thinking appreciated!)
- Need to do research and write a paper, can be done in groups of max 2-- students depending on project. Send plans to me andras@ecs.umass.edu

Grading

- 40% midterm exam (will have questions about HWs)
- 40% project, 20% HWs

3

Administrative info contd

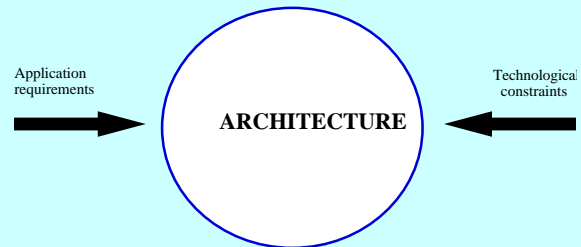
Need to have access to SUN/Solaris machine for the HWs

Acknowledgements

- Prof Anant Agarwal, MIT, Prof Rajeev Barua U Maryland, Walter Lee MIT, Michael Taylor MIT, Matt Frank MIT, .. and other members of the Raw project at MIT, as well as authors of the textbook that provided material leveraged in this class.

4

Major theme in architecture



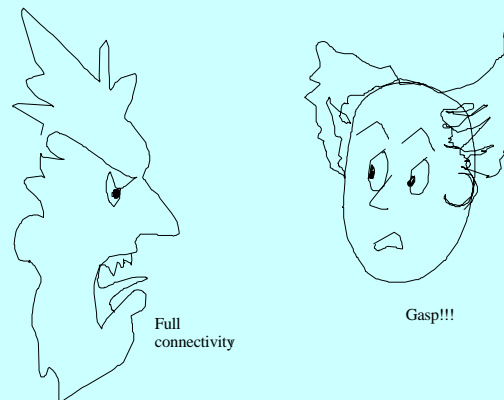
So,

- Look at typical applications
- Understand physical limitations

5

Unfortunately

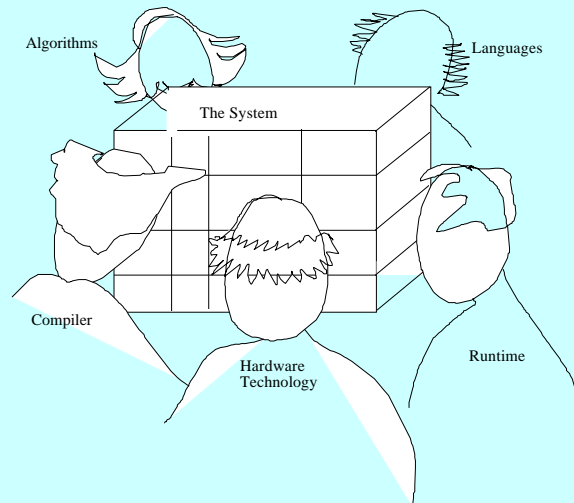
Requirements and constraints are often at odds with each other!



Architecture ---> making tradeoffs

6

Thought for the week



7

Introduction

What is Parallel Architecture?

Why Parallel Architecture?

- Application trends
- Technology trends
- Architecture trends

8

What is Parallel Architecture?

A parallel computer is a collection of processing elements that cooperate to solve large problems fast

Some broad issues:

- Resource Allocation:
 - how large a collection?
 - how powerful are the elements?
 - how much memory? How much communication bandwidth?
- Data access, Communication and Synchronization
 - how do the elements cooperate and communicate?
 - how are data transmitted between processors?
 - what are the abstractions and primitives for cooperation?
- Performance and Scalability
 - how does it all translate into performance?
 - how does it scale? (multiple processors...)

9

Why Study Parallel Architecture?

Role of a computer architect:

To design and engineer the various levels of a computer system to maximize *performance* and *programmability* within limits of *technology* and *cost*.

Need to understand hw/sw tradeoffs, fundamental concepts within each layer, ...

Why Parallelism?

- Provides alternative to faster clock for performance
- Applies at all levels of system design (hw, sw)
- Is a fascinating perspective from which to view architecture

10

Why Study it Today?

History: diverse and innovative organizational structures, often tied to novel programming models

Rapidly maturing under strong technological constraints

- The “killer micro” is ubiquitous
- Laptops and supercomputers are fundamentally similar!
- Technological trends cause diverse approaches to converge

Technological trends make parallel computing inevitable

- In the mainstream

Need to understand fundamental principles and design tradeoffs, not just taxonomies

- Naming, Ordering, Replication, Communication performance

11

Inevitability of Parallel Computing

Application demands: Our insatiable need for computing cycles

- *Scientific computing*: CFD, Biology, Chemistry, Physics, ...
- *General-purpose computing*: Video, Graphics, CAD, Databases, TP...

Technology Trends

- Number of transistors on chip growing rapidly
- Clock rates expected to go up only slowly

Architecture Trends

- Instruction-level parallelism valuable but limited
- Coarser-level parallelism, as in MPs, the most viable approach

Economics

Current trends:

- Today’s microprocessors have multiprocessor support
- Servers and workstations becoming MP: Sun, SGI, DEC, COMPAQ!...
- Tomorrow’s microprocessors are multiprocessors

12

Application Trends

Demand for cycles fuels advances in hardware, and vice-versa

- Cycle drives exponential increase in microprocessor performance
- Drives parallel architecture harder: most demanding applications

Range of performance demands

Goal of applications in using parallel machines: Speedup

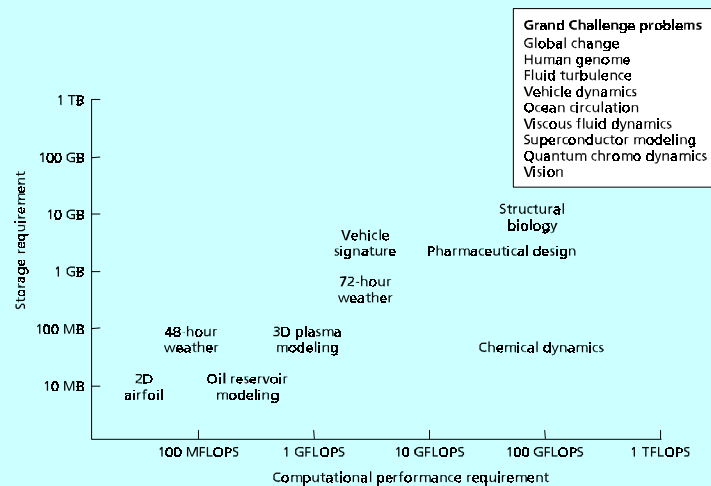
$$\text{Speedup } (p \text{ processors}) = \frac{\text{Performance } (p \text{ processors})}{\text{Performance } (1 \text{ processor})}$$

For a fixed problem size (input data set), performance = 1/time

$$\text{Speedup}_{\text{fixed problem}} (p \text{ processors}) = \frac{\text{Time } (1 \text{ processor})}{\text{Time } (p \text{ processors})}$$

13

Scientific Computing Demand



14

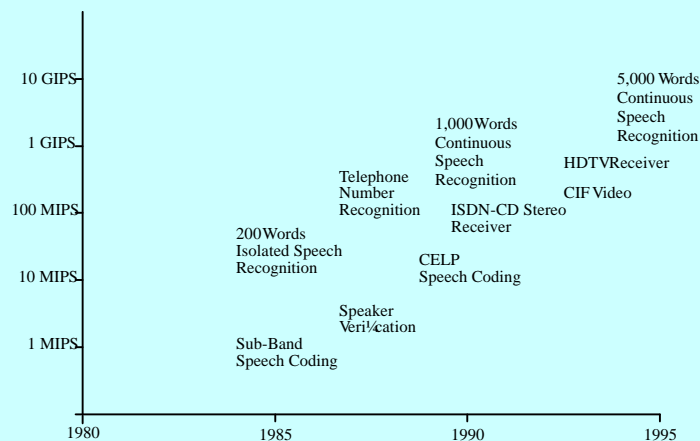
Engineering Computing Demand

Large parallel machines a mainstay in many industries

- Petroleum (reservoir analysis)
- Automotive (crash simulation, drag analysis, combustion efficiency),
- Aeronautics (airflow analysis, engine efficiency, structural mechanics, electromagnetism),
- Computer-aided design
- Pharmaceuticals (molecular modeling)
- Visualization
 - in all of the above
 - entertainment (films like Toy Story)
 - architecture (walk-throughs and rendering)
- Financial modeling (yield and derivative analysis)
- etc.

15

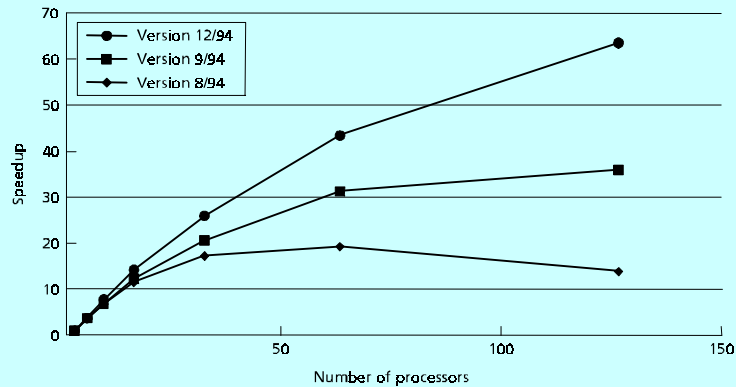
Applications: Speech and Image Processing



- Also CAD, Databases, . . .
- *100 processors gets you 10 years, 1000 gets you 20 !*

16

Learning Curve for Parallel Applications



- AMBER molecular dynamics simulation program
- Starting point was vector code for Cray-1
- 145 MFLOP on Cray90, 406 for final version on 128-processor Paragon, 891 on 128-processor Cray T3D

17

Commercial Computing

Also relies on parallelism for high end

- Scale not so large, but use much more wide-spread
- Computational power determines scale of business that can be handled

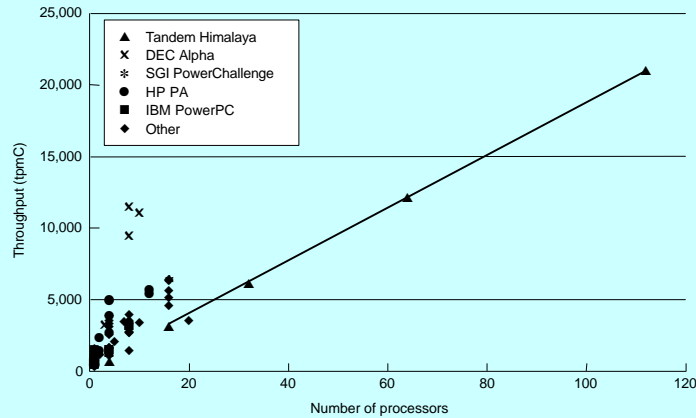
Databases, online-transaction processing, decision support, data mining, data warehousing ...

TPC benchmarks (TPC-C order entry, TPC-D decision support)

- Explicit scaling criteria provided
- Size of enterprise scales with size of system
- Problem size no longer fixed as p increases, so
- NOTE: throughput is used as a performance measure (transactions per minute or *tpm*)

18

TPC-C Results for March 1996



- Parallelism is pervasive
- Small to moderate scale parallelism very important
- Difficult to obtain snapshot to compare across vendor platforms

19

Summary of Application Trends

Transition to parallel computing has occurred for scientific and engineering computing

In rapid progress in commercial computing

- Database and transactions as well as financial
- Usually smaller-scale, but large-scale systems also used

Desktop also uses multithreaded programs, which are a lot like parallel programs

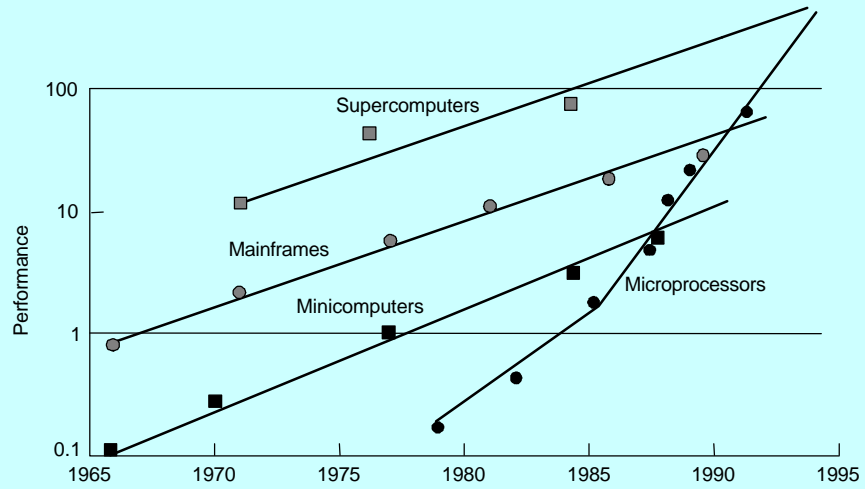
Demand for improving throughput on sequential workloads

- Greatest use of small-scale multiprocessors

Solid application demand exists and will increase

20

Technology Trends

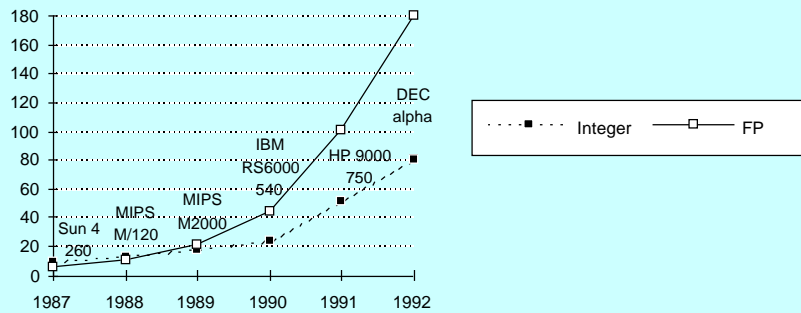


The natural building block for multiprocessors is now also about the fastest!

21

General Technology Trends

- *Microprocessor performance* increases 50% - 100% per year
- *Transistor count* doubles every 3 years
- *DRAM size* quadruples every 3 years
- Huge investment per generation is carried by huge commodity market



- Not that single-processor performance is plateauing, but that parallelism is a natural way to improve it.

22

Technology: A Closer Look

Basic advance is *decreasing feature size* (λ)

- Circuits become either faster or lower in power

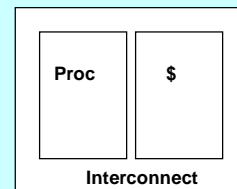
Die size is growing too

- Clock rate improves roughly proportional to improvement in λ
- Number of transistors improves like λ^2 (or faster)

Performance > 100x per decade; clock rate 10x, rest transistor count

How to use more transistors?

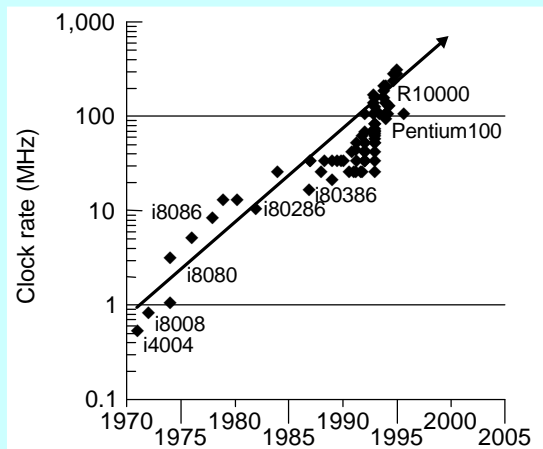
- Parallelism in processing
 - multiple operations per cycle reduces CPI
- Locality in data access
 - avoids latency and reduces CPI
 - also improves processor utilization
- Both need resources, so tradeoff



Fundamental issue is resource distribution, as in uniprocessors

23

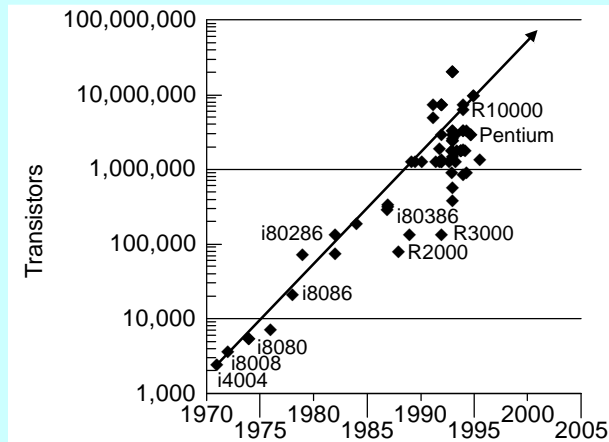
Clock Frequency Growth Rate



- 30% per year

24

Transistor Count Growth Rate



- 100 million transistors on chip by early 2000's A.D.
- Transistor count grows much faster than clock rate
 - 40% per year, order of magnitude more contribution in 2 decades

25

Similar Story for Storage

Divergence between memory capacity and speed more pronounced

- Capacity increased by 1000x from 1980-95, speed only 2x
- Gigabit DRAM by c. 2000, but gap with processor speed much greater

Larger memories are slower, while processors get faster

- Need to transfer more data in parallel
- Need deeper cache hierarchies
- How to organize caches?

Parallelism increases effective size of each level of hierarchy, without increasing access time

Parallelism and locality within memory systems too

- New designs fetch many bits within memory chip; follow with fast pipelined transfer across narrower interface
- Buffer caches most recently accessed data

Disks too: Parallel disks plus caching

26

Architectural Trends

Architecture translates technology's gifts to performance and capability

Resolves the tradeoff between parallelism and locality

- Current microprocessor: 1/3 compute, 1/3 cache, 1/3 off-chip connect
- Tradeoffs may change with scale and technology advances

Understanding microprocessor architectural trends

- Helps build intuition about design issues or parallel machines
- Shows fundamental role of parallelism even in "sequential" computers

Four generations of architectural history: tube, transistor, IC, VLSI

- Here focus only on VLSI generation

Greatest delineation in VLSI has been in type of parallelism exploited

27

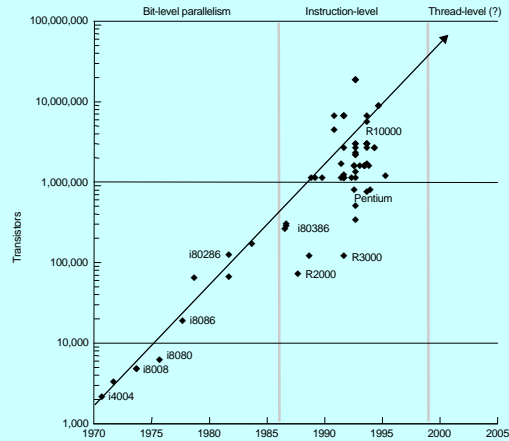
Architectural Trends

Greatest trend in VLSI generation is increase in parallelism

- Up to 1985: bit level parallelism: 4-bit -> 8 bit -> 16-bit
 - slows after 32 bit
 - adoption of 64-bit now under way, 128-bit far (not performance issue)
 - great inflection point when 32-bit micro and cache fit on a chip
- Mid 80s to mid 90s: instruction level parallelism
 - pipelining and simple instruction sets, + compiler advances (RISC)
 - on-chip caches and functional units => superscalar execution
 - greater sophistication: out of order execution, speculation, prediction
 - to deal with control transfer and latency problems
- Next step: thread level parallelism

28

Phases in VLSI Generation



- How good is instruction-level parallelism?
- Thread-level needed in microprocessors?

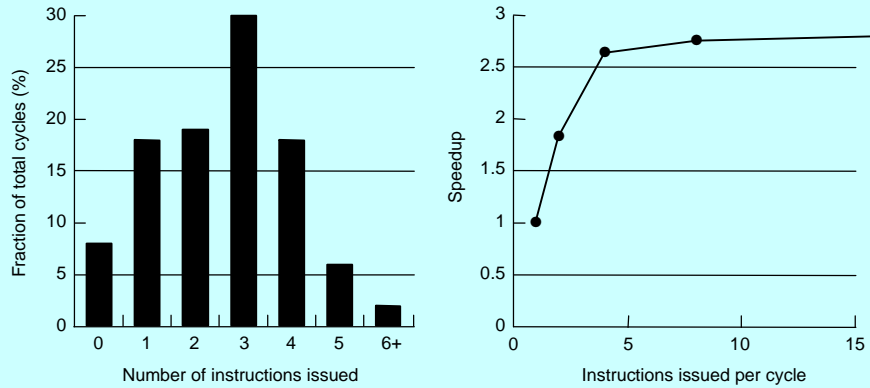
29

Architectural Trends: ILP

- Reported speedups for superscalar processors
 - Horst, Harris, and Jardine [1990] 1.37
 - Wang and Wu [1988] 1.70
 - Smith, Johnson, and Horowitz [1989] 2.30
 - Murakami et al. [1989] 2.55
 - Chang et al. [1991] 2.90
 - Jouppi and Wall [1989] 3.20
 - Lee, Kwok, and Briggs [1991] 3.50
 - Wall [1991] 5
 - Melvin and Patt [1991] 8
 - Butler et al. [1991] 17+
- Large variance due to difference in
 - application domain investigated (numerical versus non-numerical)
 - capabilities of processor modeled

30

ILP Ideal Potential

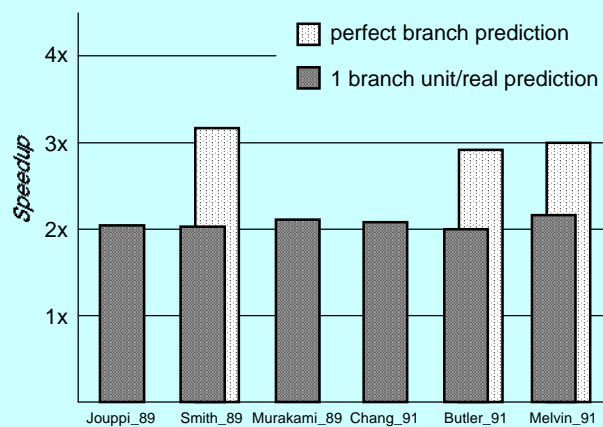


- Infinite resources and fetch bandwidth, perfect branch prediction and renaming
 - real caches and non-zero miss latencies

31

Results of ILP Studies

- Concentrate on parallelism for 4-issue machines

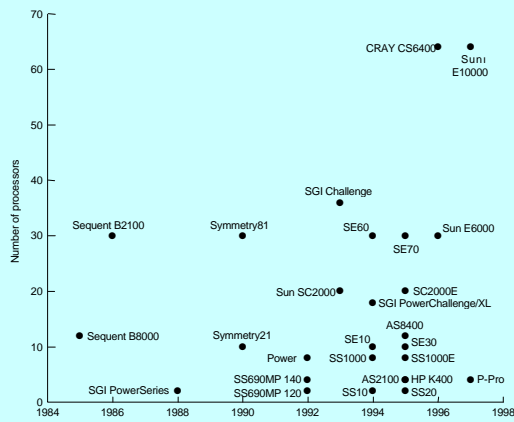


- Realistic studies show only 2-fold speedup
 - Recent studies show that more ILP needs to look across threads

32

Architectural Trends: Bus-based MPs

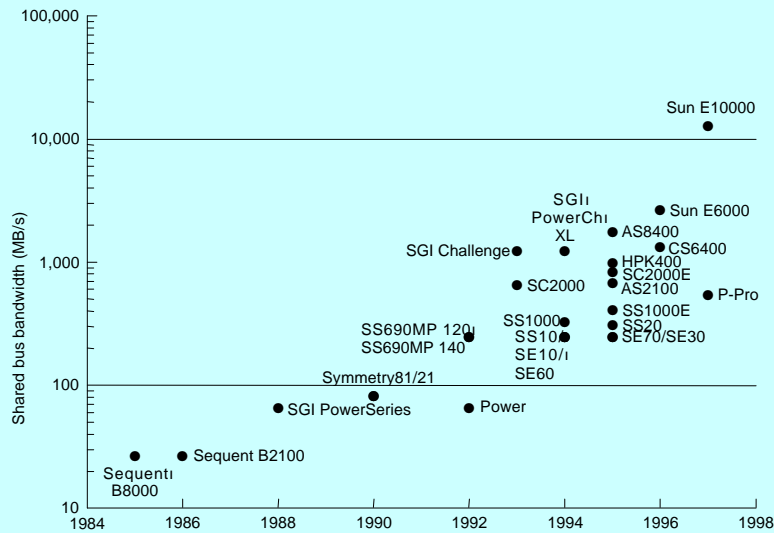
- Micro on a chip makes it natural to connect many to shared memory
 - dominates server and enterprise market, moving down to desktop
- Faster processors began to saturate bus, then bus technology advanced
 - today, range of sizes for bus-based systems, desktop to large servers



No. of processors in fully configured commercial shared-memory systems

33

Bus Bandwidth



34

Economics

Commodity microprocessors not only fast but CHEAP

- Development cost is tens of millions of dollars (5-100 typical)
- BUT, many more are sold compared to supercomputers
- Crucial to take advantage of the investment, and use the commodity building block
- Exotic parallel architectures no more than special-purpose

Multiprocessors being pushed by software vendors (e.g. database) as well as hardware vendors

Standardization by Intel makes small, bus-based SMPs commodity

Desktop: few smaller processors versus one larger one?

- Multiprocessor on a chip

35

Consider Scientific Supercomputing

Proving ground and driver for innovative architecture and techniques

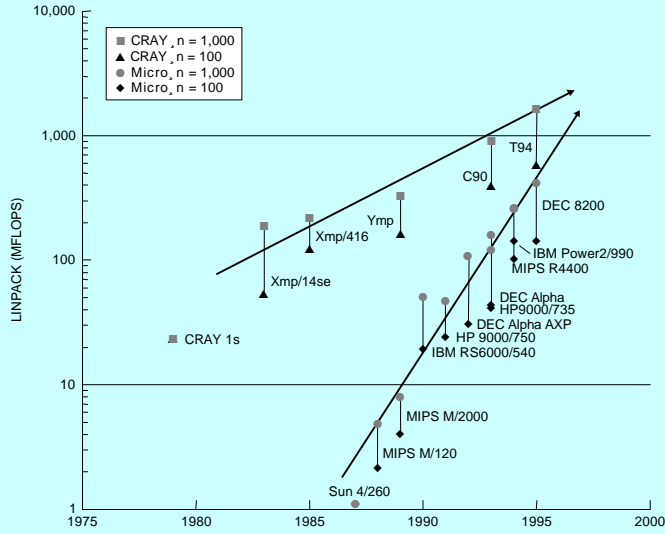
- Market smaller relative to commercial as MPs become mainstream
- Dominated by vector machines starting in 70s
- Microprocessors have made huge gains in floating-point performance
 - high clock rates
 - pipelined floating point units (e.g., multiply-add every cycle)
 - instruction-level parallelism
 - effective use of caches (e.g., automatic blocking)
- Plus economics

Large-scale multiprocessors replace vector supercomputers

- Well under way already

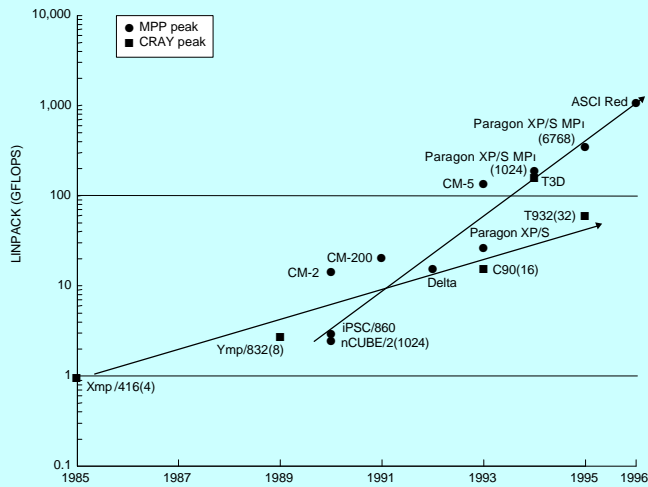
36

Raw Uniprocessor Performance: LINPACK



37

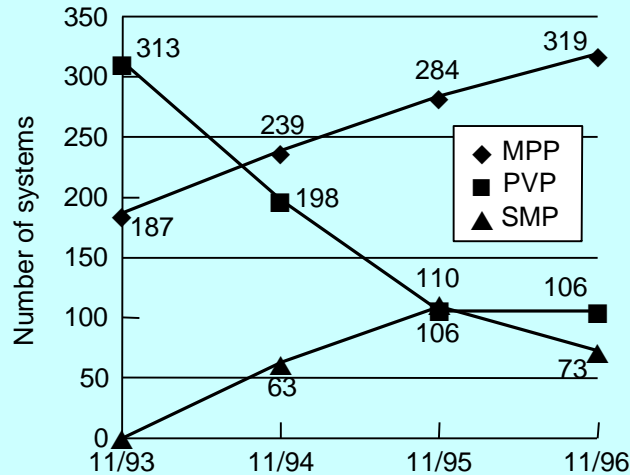
Raw Parallel Performance: LINPACK



- Even vector Crays became parallel: X-MP (2-4) Y-MP (8), C-90 (16), T94 (32)
- Since 1993, Cray produces MPPs too (T3D, T3E)

38

500 Fastest Computers



39

Summary: Why Parallel Architecture?

Increasingly attractive

- Economics, technology, architecture, application demand

Increasingly central and mainstream

Parallelism exploited at many levels

- Instruction-level parallelism
- Multiprocessor servers
- Large-scale multiprocessors ("MPPs")

Focus of this class: multiprocessor level of parallelism

Same story from memory system perspective

- Increase bandwidth, reduce average latency with many local memories

Wide range of parallel architectures make sense?

- Different cost, performance and scalability

40