

## CS 258 Parallel Computer Architecture

CS 258, Spring 99  
David E. Culler  
Computer Science Division  
U.C. Berkeley

### Today's Goal:

- Introduce you to Parallel Computer Architecture
- Answer your questions about CS 258
- Provide you a sense of the trends that shape the field

1/19/99

CS258 S99

2

### What will you get out of CS258?

- In-depth understanding of the design and engineering of modern parallel computers
  - technology forces
  - fundamental architectural issues
    - » naming, replication, communication, synchronization
  - basic design techniques
    - » cache coherence, protocols, networks, pipelining, ...
  - methods of evaluation
  - underlying engineering trade-offs
- from moderate to very large scale
- across the hardware/software boundary

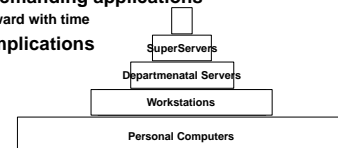
1/19/99

CS258 S99

3

### Will it be worthwhile?

- Absolutely!
  - even though few of you will become PP designers
- The fundamental issues and solutions translate across a wide spectrum of systems.
  - Crisp solutions in the context of parallel machines.
- Pioneered at the thin-end of the platform pyramid on the most-demanding applications
  - migrate downward with time
- Understand implications for software



1/19/99

CS258 S99

4

### Am I going to read my book to you?

- **NO!**
- Book provides a framework and complete background, so lectures can be more interactive.
  - You do the reading
  - We'll discuss it
- Projects will go "beyond"



1/19/99

CS258 S99

5

### 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?
  - 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?

1/19/99

CS258 S99

6

## 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*.

Parallelism:

- Provides alternative to faster clock for performance
- Applies at all levels of system design
- Is a fascinating perspective from which to view architecture
- Is increasingly central in information processing

1/19/99

CS258 S99

7

## 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**
- **Need to understand fundamental principles and design tradeoffs, not just taxonomies**
  - Naming, Ordering, Replication, Communication performance

1/19/99

CS258 S99

8

## Is Parallel Computing Inevitable?

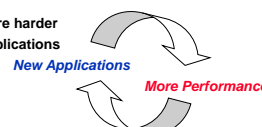
- **Application demands: Our insatiable need for computing cycles**
- **Technology Trends**
- **Architecture Trends**
- **Economics**
- **Current trends:**
  - Today's microprocessors have multiprocessor support
  - Servers and workstations becoming MP: Sun, SGI, DEC, COMPAQ!...
  - Tomorrow's microprocessors are multiprocessors

1/19/99

CS258 S99

9

## Application Trends

- **Application demand for performance fuels advances in hardware, which enables new appl'ns, which...**
    - Cycle drives exponential increase in microprocessor performance
    - Drives parallel architecture harder
      - » most demanding applications
- 
- **Range of performance demands**
    - Need range of system performance with progressively increasing cost

1/19/99

CS258 S99

10

## Speedup

- **Speedup (p processors) =**  $\frac{\text{Performance (p processors)}}{\text{Performance (1 processor)}}$
- **For a fixed problem size (input data set), performance = 1/time**
- **Speedup fixed problem (p processors) =**  $\frac{\text{Time (1 processor)}}{\text{Time (p processors)}}$

1/19/99

CS258 S99

11

## Commercial Computing

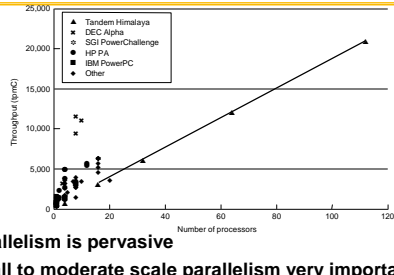
- **Relies on parallelism for high end**
  - Computational power determines scale of business that can be handled
- **Databases, online-transaction processing, decision support, data mining, data warehousing ...**
- **TPC benchmarks (TPC-C order entry, TPC-D decision support)**
  - Explicit scaling criteria provided
  - Size of enterprise scales with size of system
  - Problem size not fixed as p increases.
  - **Throughput is performance measure** (transactions per minute or tpm)

1/19/99

CS258 S99

12

## TPC-C Results for March 1996



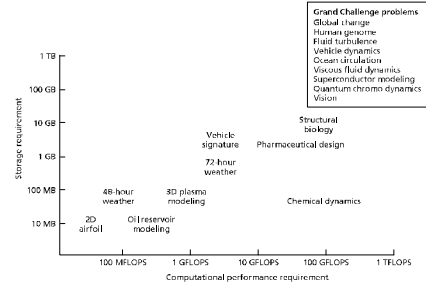
- Parallelism is pervasive
- Small to moderate scale parallelism very important
- Difficult to obtain snapshot to compare across vendor platforms

1/19/99

CS258 S99

13

## Scientific Computing Demand



1/19/99

CS258 S99

14

## Engineering Computing Demand

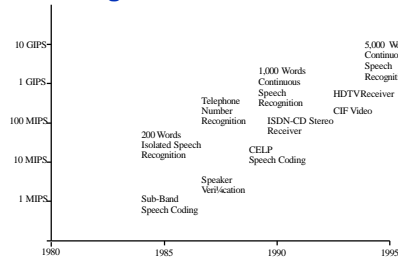
- Large parallel machines a mainstay in many industries
  - Petroleum (reservoir analysis)
  - Automotive (crash simulation, drag analysis, combustion efficiency)
  - Aeronautics (airflow analysis, engine efficiency, structural mechanics, electromagnetism)
  - Computer-aided design
  - Pharmaceuticals (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.

1/19/99

CS258 S99

15

## Applications: Speech and Image Processing



• Also CAD, Databases, . . .

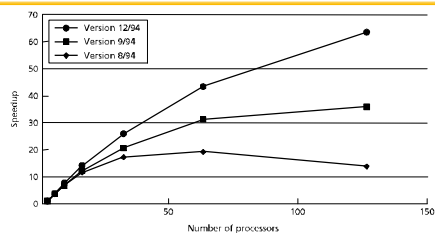
• 100 processors gets you 10 years, 1000 gets you 20 !

1/19/99

CS258 S99

16

## Is better parallel arch enough?



- 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

1/19/99

CS258 S99

17

## 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

1/19/99

CS258 S99

18

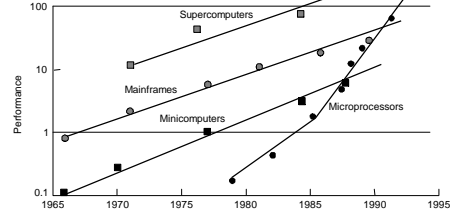
--- Little break ---

1/19/99

CS258 S99

19

### Technology Trends



• Today the natural building-block is also fastest!

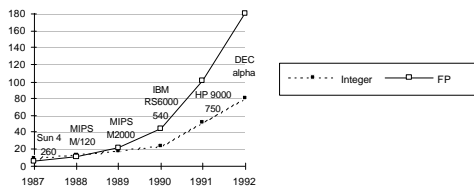
1/19/99

CS258 S99

20

### Can't we just wait for it to get faster?

- *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



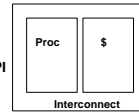
1/19/99

CS258 S99

21

### Technology: A Closer Look

- **Basic advance is decreasing feature size (l)**
  - Circuits become either faster or lower in power
- **Die size is growing too**
  - Clock rate improves roughly proportional to improvement in l
  - Number of transistors improves like l<sup>2</sup> (or faster)
- **Performance > 100x per decade**
  - clock rate < 10x, rest is 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**

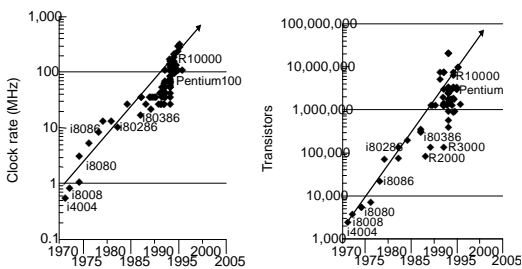


1/19/99

CS258 S99

22

### Growth Rates



• 30% per year

40% per year

1/19/99

CS258 S99

23

### Architectural Trends

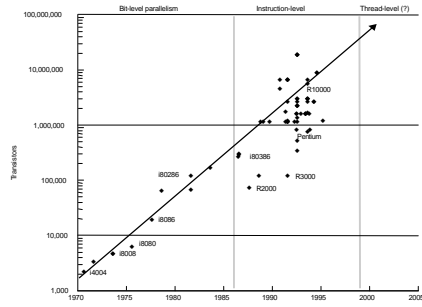
- **Architecture translates technology's gifts into performance and capability**
- **Resolves the tradeoff between parallelism and locality**
  - Current microprocessor: 1/3 compute, 1/3 cache, 1/3 off-chip connect
  - Tradeoffs may change with scale and technology advances
- **Understanding microprocessor architectural trends**
  - => Helps build intuition about design issues or parallel machines
  - => Shows fundamental role of parallelism even in "sequential" computers

1/19/99

CS258 S99

24

## Phases in "VLSI" Generation



1/19/99

CS258 S99

25

## Architectural Trends

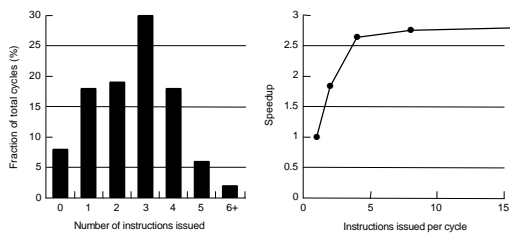
- **Greatest trend in VLSI generation is increase in parallelism**
  - Up to 1985: bit level parallelism: 4-bit -> 8 bit -> 16-bit
    - » slows after 32 bit
    - » adoption of 64-bit now under way, 128-bit far (not performance issue)
    - » great inflection point when 32-bit micro and cache fit on 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

1/19/99

CS258 S99

26

## How far will ILP go?



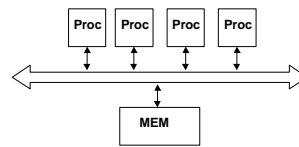
- **Infinite resources and fetch bandwidth, perfect branch prediction and renaming**
  - real caches and non-zero miss latencies

1/19/99

CS258 S99

27

## Threads Level Parallelism "on board"



- **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

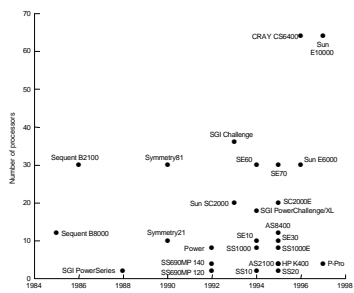
1/19/99

CS258 S99

28

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

## What about Multiprocessor Trends?

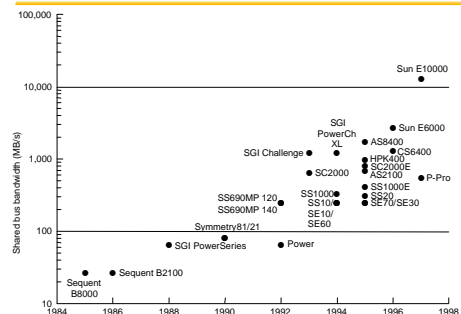


1/19/99

CS258 S99

29

## Bus Bandwidth



1/19/99

CS258 S99

30

## What about Storage Trends?

- Divergence between memory capacity and speed even 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

1/19/99

CS258 S99

31

## Economics

- Commodity microprocessors not only fast but CHEAP
  - Development costs tens of millions of dollars
  - BUT, many more are sold compared to supercomputers
  - Crucial to take advantage of the investment, and use the commodity building block
- Multiprocessors being pushed by software vendors (e.g. database) as well as hardware vendors
- Standardization makes small, bus-based SMPs commodity
- Desktop: few smaller processors versus one larger one?
- Multiprocessor on a chip?

1/19/99

CS258 S99

32

## Can we see some hard evidence?

1/19/99

CS258 S99

33

## Consider Scientific Supercomputing

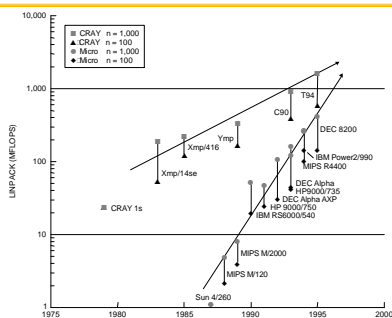
- Proving ground and driver for innovative architecture and techniques
  - Market smaller relative to commercial as MPs become mainstream
  - Dominated by vector machines starting in 70s
  - Microprocessors have made huge gains in floating-point performance
    - » high clock rates
    - » pipelined 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

1/19/99

CS258 S99

34

## Raw Uniprocessor Performance: LINPACK

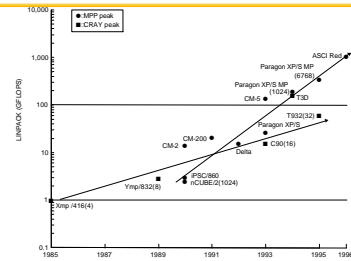


1/19/99

CS258 S99

35

## Raw Parallel Performance: LINPACK



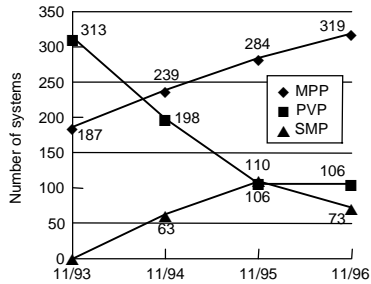
- Even vector Crays became parallel
  - X-MP (2-4) Y-MP (8), C-90 (16), T94 (32)
- Since 1993, Cray produces MPPs too (T3D, T3E)

1/19/99

CS258 S99

36

## 500 Fastest Computers



1/19/99

CS258 S99

37

## 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
- **Spectrum of parallel architectures make sense**
  - Different cost, performance and scalability

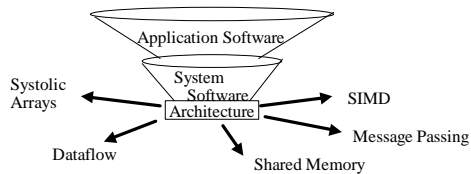
1/19/99

CS258 S99

38

## Where is Parallel Arch Going?

Old view: Divergent architectures, no predictable pattern of growth.



- Uncertainty of direction paralyzed parallel software development!

1/19/99

CS258 S99

39

## Today

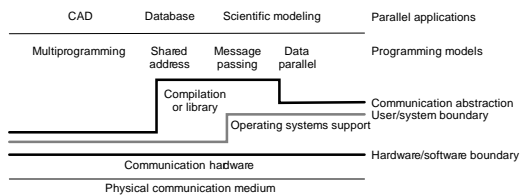
- **Extension of "computer architecture" to support communication and cooperation**
  - Instruction Set Architecture **plus Communication Architecture**
- **Defines**
  - Critical abstractions, boundaries, and primitives (interfaces)
  - Organizational structures that implement interfaces (hw or sw)
- **Compilers, libraries and OS are important bridges today**

1/19/99

CS258 S99

40

## Modern Layered Framework



1/19/99

CS258 S99

41

## How will we spend our time?

<http://www.cs.berkeley.edu/~culler/cs258-s99/schedule.html>

1/19/99

CS258 S99

42

### How will grading work?

---

- 30% homeworks (6)
- 30% exam
- 30% project (teams of 2)
- 10% participation

1/19/99

CS258 S99

43

### Any other questions?

---

1/19/99

CS258 S99

44