
ISCA 25: Looking Backward, Looking Forward

**John Hennessy
Stanford University
June 1998**

Outline

- **Our Successes:**
 - obvious and not so obvious
- **Important Trends in the Past 25 Years**
 - What we work on
 - How we work
 - Who we work with
- **Where are we going and what does it mean?**
 - Trends in computing
 - Maintaining our impact

TWENTY FIVE YEARS OF BUILDING THE Discipline and Harvesting Success

- **Building a Scientific Foundation for Computer Architecture**
 - **Instruction Set Architecture**
 - **The Role of Time**
 - **New Frontiers**
- **Building the ILP Highway**
- **Tunneling Through the Memory Wall**
- **Bridging the Parallel Processing Swamp**

Building the Science of Computer Architecture

- **Instruction Set Architecture: making ISA design quantitative & doing retrospective evaluation**
 - ISCA-3: PDP-11 Retrospective: address space and architectural extensibility
 - ISCA-7: Retrospective on High-Level Language Computer Architecture: what matters is the optimal combination of HW and SW
 - ISCA-8: RISC-I: driving the architecture from measurements of compiled programs
- **It's About Time: Understanding how ISA affects performance**
 - ISCA-4: Instruction Timing Model of CPU Performance: instruction times vary widely
 - ISCA-11: Characterization of Processor Performance: Time = CPI x IC / CR

Building the ILP Highway

- **ISCA-3: Improving Pipeline Performance: pipelining=parallelism.**
- **ISCA-8: Branch Prediction: Branches are key to fast pipelining; take advantage of program behavior.**
- **ISCA-9: Decoupled Execute/Access: Blend statically and dynamically scheduled pipelines**
- **ISCA-10 VLIW: ILP as a concept.**
- **ISCA-12: Handling precise interrupts: enables out-of-order**
- **ISCA-13: HPSm: Data-flow/dynamic scheduling; checkpoint/restart**
- **ISCA-14: Instruction issue logic: reorder buffer for speculation**
- **ISCA-18: IMPACT: Compiler technology with the right HW support**
- **ISCA-19: Two-Level Branch Prediction: To get more ILP, branch prediction crucial**
- **ISCA-22: Multiscalar: blending ILP and multiprocessing**

Tunneling Through the Memory Wall

- **ISCA-3: Caches on the PDP-11: An early performance study.**
- **ISCA-8: Lock-up free caches: ahead of its time.**
- **ISCA-10: Snoopy caches: a key insight; more important than it looked!**
- **ISCA-11: The Illinois Protocol: one of several important advancements to coherency protocols.**
- **ISCA 13,17: Consistency papers: balancing correctness and performance.**
- **ISCA-15: Directories: compare well with snooping, but that's not the key.**
- **ISCA-15: Inclusion Properties: An important simplification for coherence.**
- **ISCA-17: Victim buffers and prefetch buffers: improving caches.**

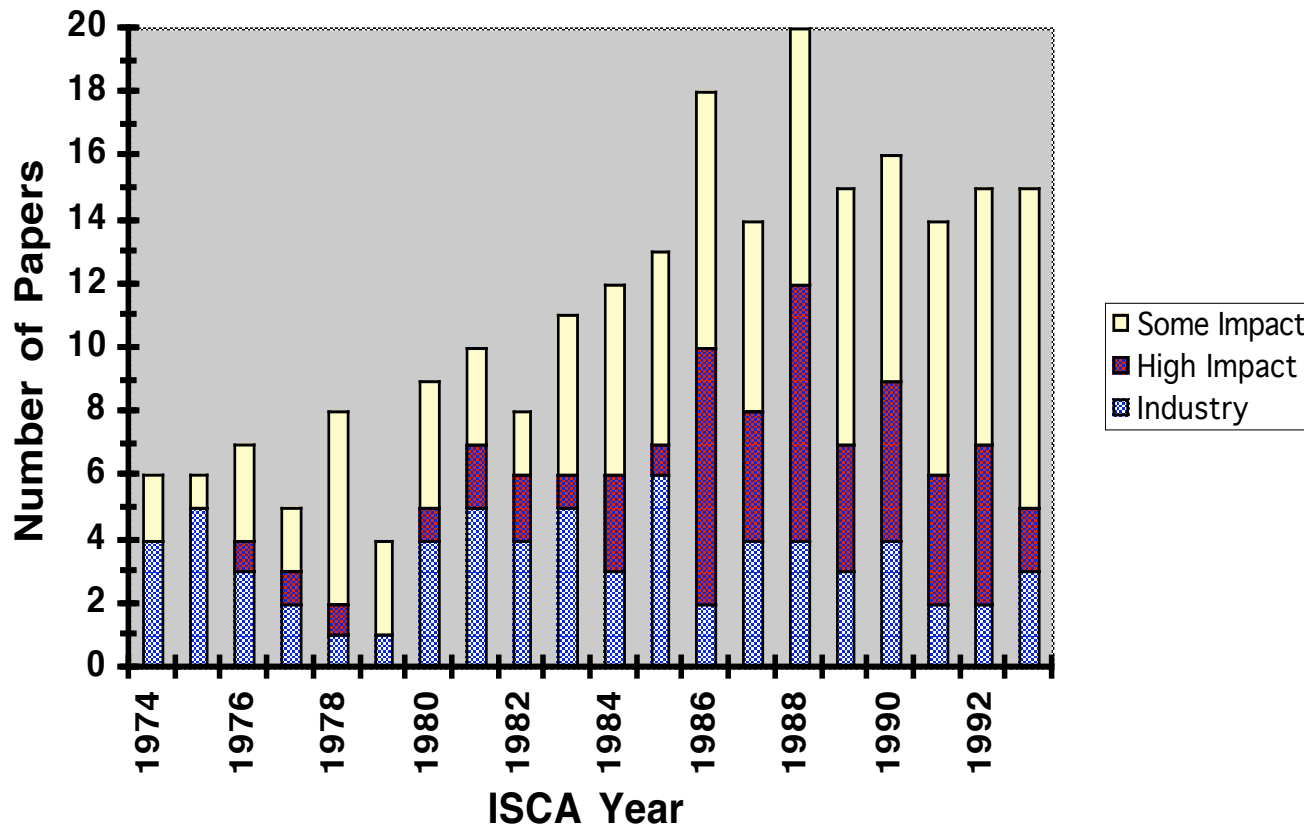
Bridging the Parallel Processing Swamp

- **ISCA-7: Massively Parallel: The advantages of SIMD.**
- **ISCA-9: Ultracomputer: challenges of scalable shared-memory**
- **ISCA-19: DASH: scalable cache-coherency can be efficient**
- **ISCA-19: Active Messages: understanding the SW/HW boundary in multiprocessor communications.**
- **ISCA-20: Cedar: mixing HW and SW strategies in a shared-memory MP**
- **ISCA-21: Shrimp: Building secure efficient, mechanisms for clusters**
- **ISCA-21: FLASH, Tempest and Typhoon making more flexible substrates for MPs**
- **ISCA-22: Alewife: using multithreading to tolerate latency; blending SW and HW support.**

Important Keys to the Success

- **Quantitative Research**
 - it's about performance and cost
- **Industrial Relevance**
 - there's someone at ISCA besides the academics
- **Experimentally Based**
 - propose and evaluate ideas
- **Prototyping**
 - build it and show me
- **Integration of SW Research**
 - good hardware requires good sare

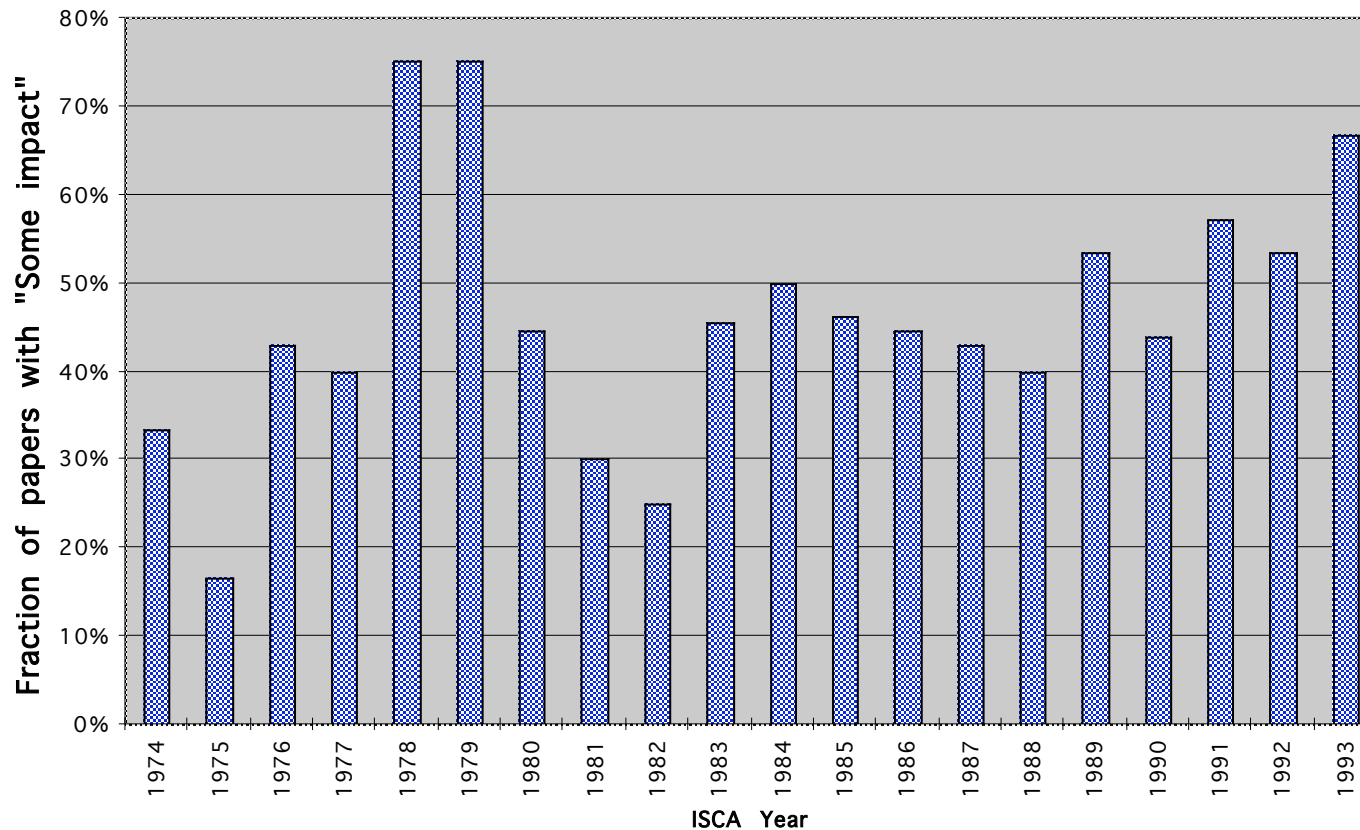
Industrial Relevance



- Twenty years of ISCA papers
- Relevance judged by ad hoc assessment
- Significant increase since early years (and with fewer papers)

Danger: Incrementality

- Danger of becoming too tied to short-term industrial goals.
- Particularly dangerous when much of the research agenda is extending and refining ideas in a small number of areas.



Experimentally Based

- **Clear trend in favor of experimentally based work.**
- **Increased emphasis on quantitative evaluation.**
- **Increased emphasis on complete system issues.**

Danger: Poor Experimental Methodology

Prototyping

Danger: Overwhelmed

Increased SW Expertise

- **Knowledge**
- **Balanced approach**

Danger: Breadth of Research and Team

Trends in Computing

- **Away from the desktop**
- **Consolidation of the desktop**
- **More systems focus**
 - networking
 - SW
- **Memory systems**
- **Parallelism**
- **Balancing Performance, Cost, and Programming Pain**
- **Grappling with the Challenge of Moore's Law**
- **Preparing for the end of Moore's Law**

SOME OPEN QUESTIONS THAT BUG Me

■ The Complexity of Tradeoffs

- ease of program
- compatability
- performance
- cost

■ Exploiting ILP

- HW intensive vs SW intensive approaches
- Accurately understanding what's going on

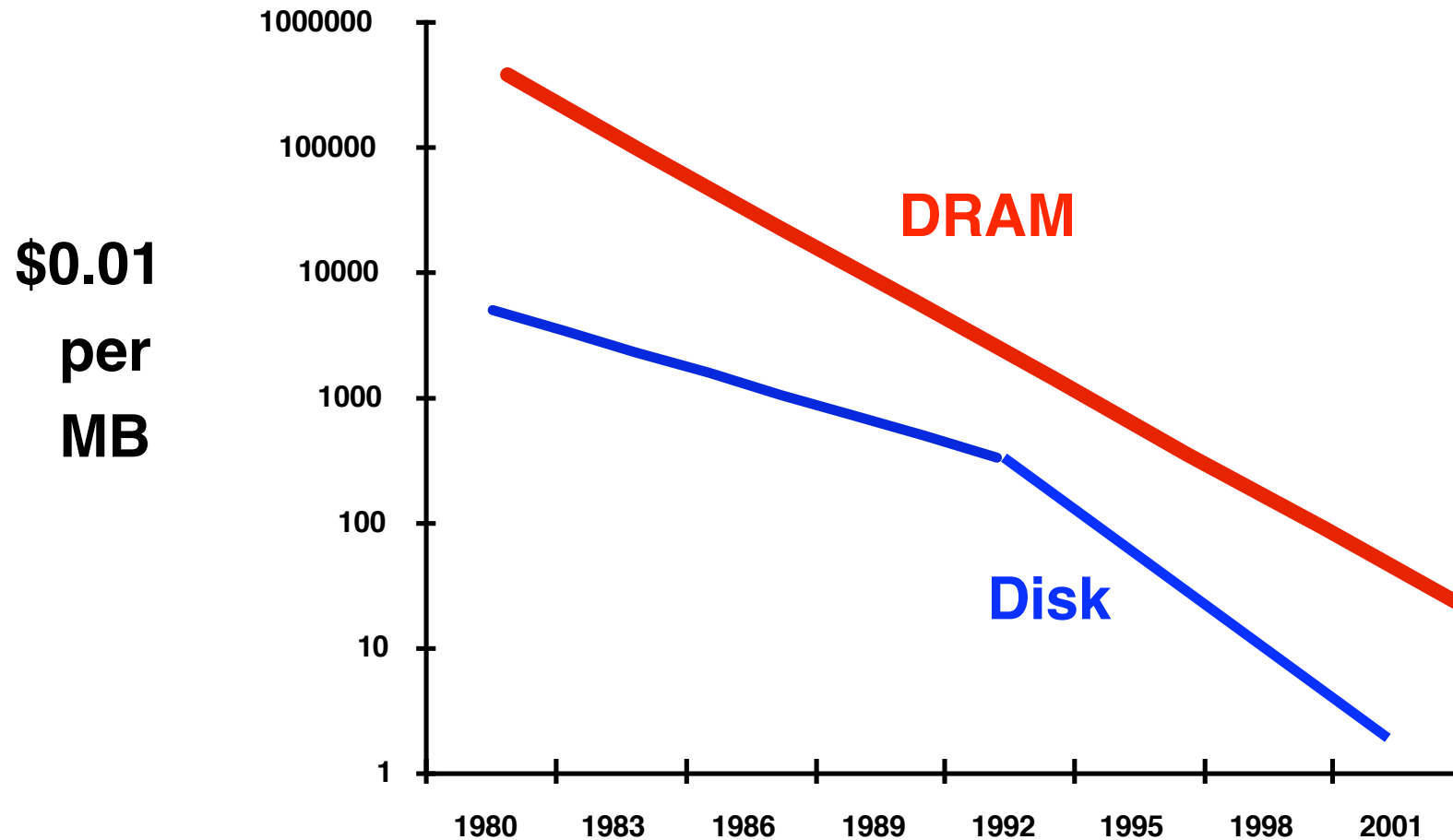
■ MPs

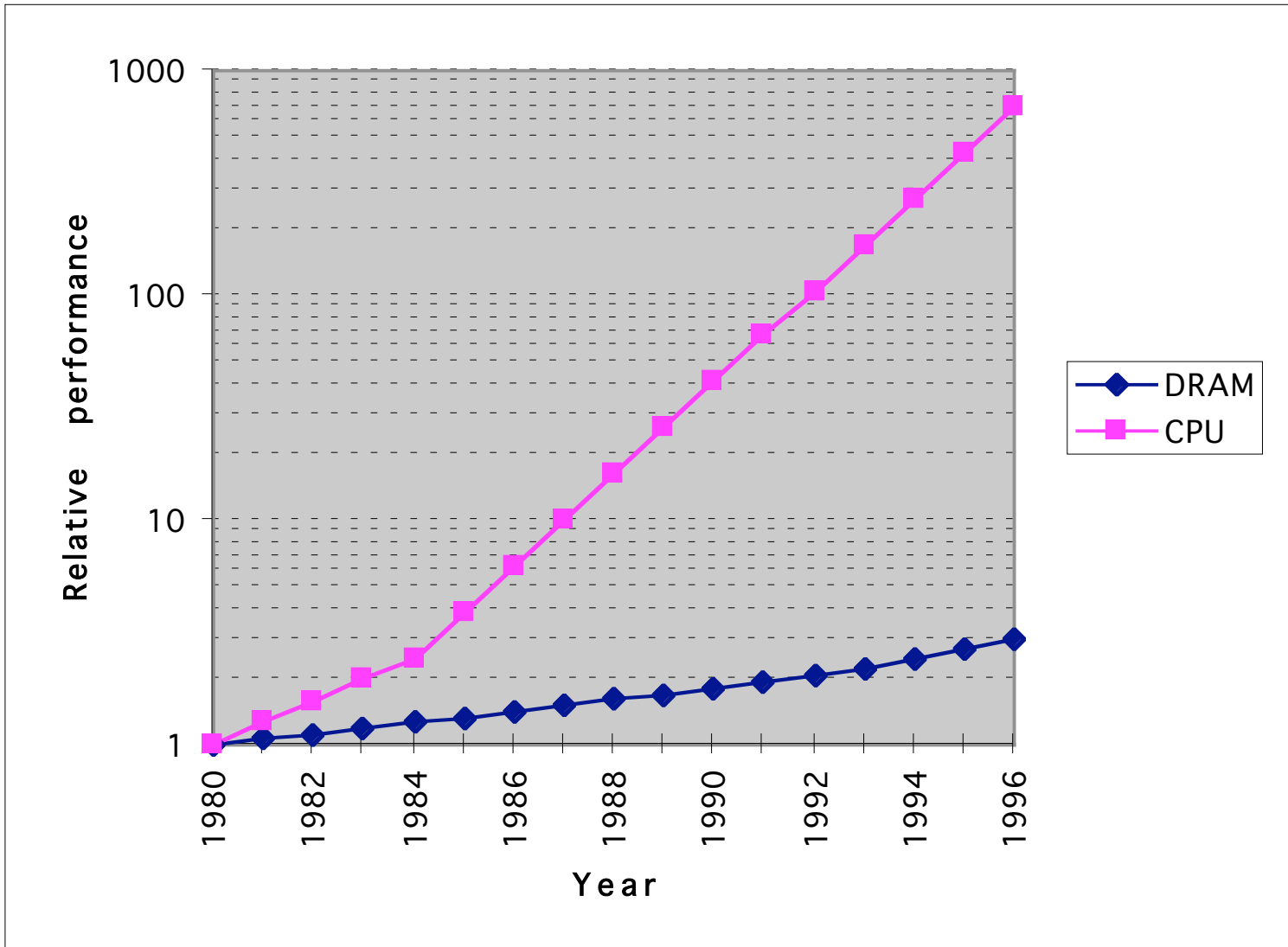
- can't we do better
- helping the programming problem

■ Memory systems

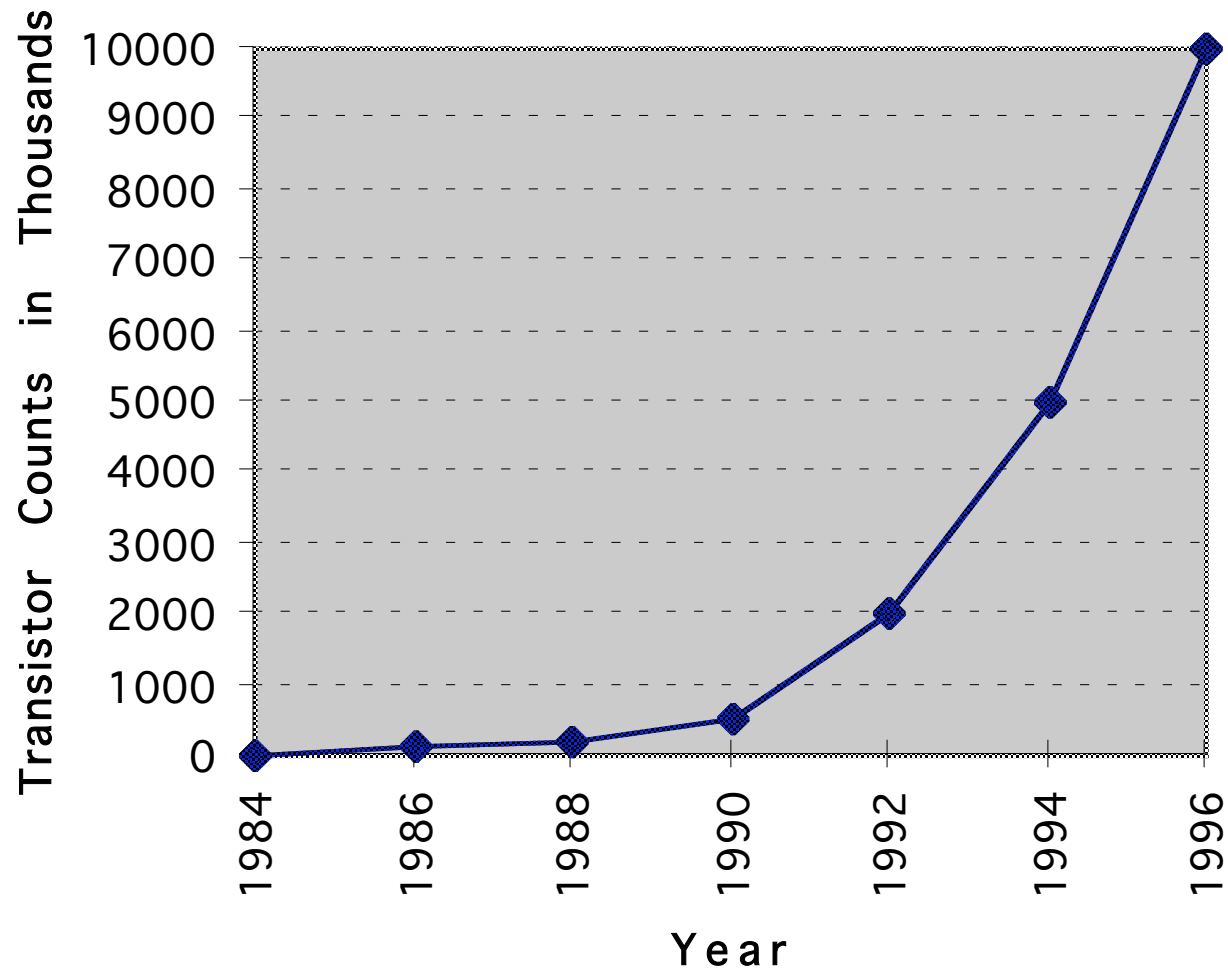
- what about the apps left behind
- what about I/O

Memory Technologies

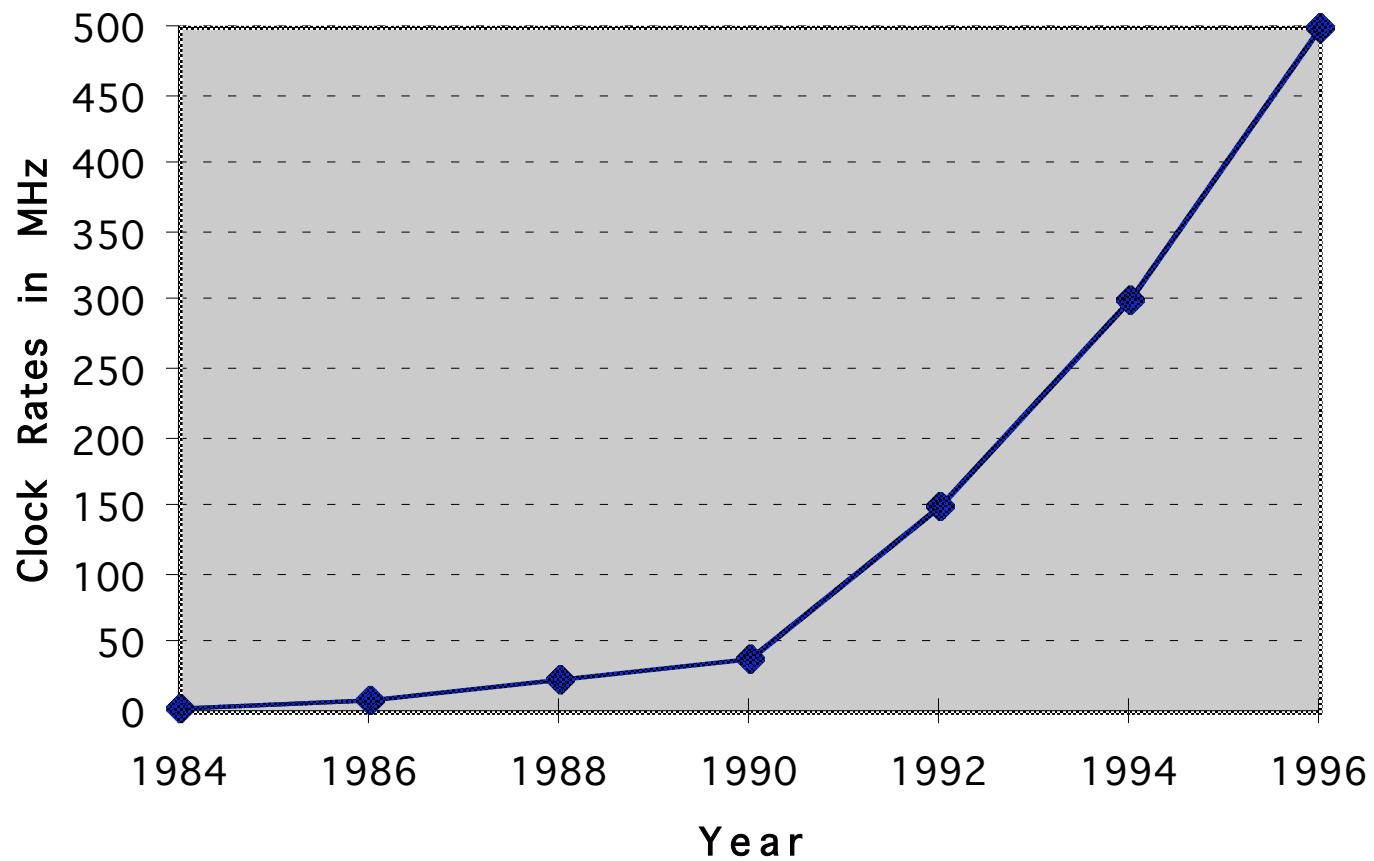




Microprocessor Transistor Counts



Microprocessor Clock Rates



Microprocessors : past 15 years

- **Microprocessor performance has been growing at an unprecedented pace for nearly 15 years**
 - Year to year improvements $\geq 1.5 \times$
- **Performance growth has come from combining**
 - direct advantages of technology (e.g., faster transistors)
 - architectural improvements that exploit increased transistor counts
 - architectural ideas are probably responsible for at least half of the performance growth

Key Architectural Trends

■ Memory hierarchies:

- **increasingly large and sophisticated memory hierarchies**
 - on-chip: from 128 bytes to 100K+ bytes
 - off-chip: from 32KB to 4-16 MB
- **advances in caching techniques:**
 - past five years: microprocessors develop new approaches

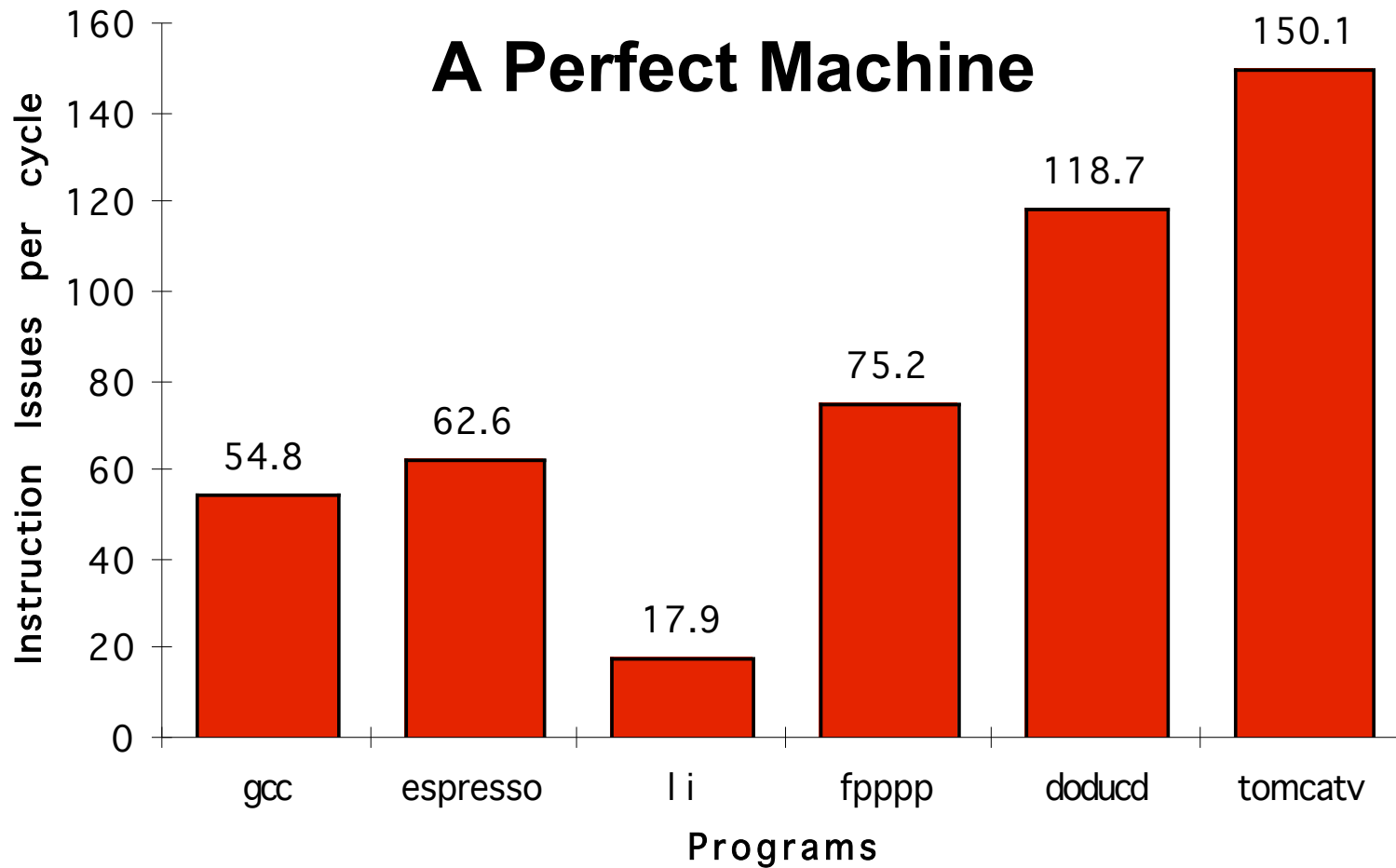
■ Exploiting Instruction Level Parallelism (ILP)

- **Simple, shallow integer pipelines (1985) to complex, deep pipelines**
- **From 1 instruction issue/clock to 4-6 instruction issues/clock**
- **From compiler-driven static schemes to dynamic schemes:**
 - branch prediction
 - pipeline scheduling
 - alias analysis

The Challenge

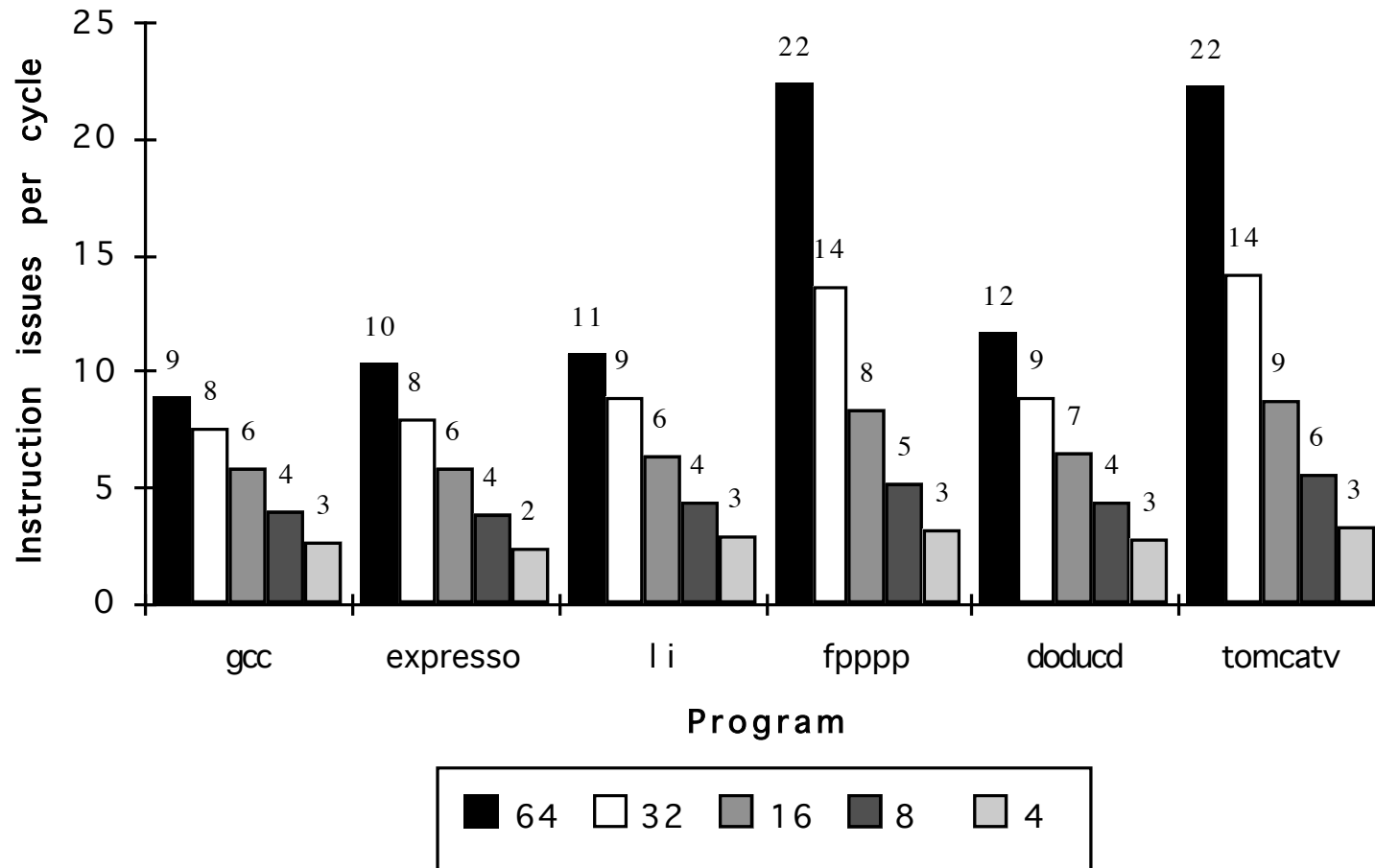
- **Diminishing returns from attempts to increase instructions per clock**
 - coupled with increasing complexity and clock speed penalty
 - The existing approaches will suffice for another generation, but probably not much beyond.
- **Memory systems increasingly become the bottleneck**
 - New applications strain the limits of existing approaches.
 - Continuing incremental innovation will work for a narrower range of applications.
- **Design after 2000:**
 - 10^8 – 10^9 transistors!
 - New ideas are needed to exploit this capability.

ILP Measurements



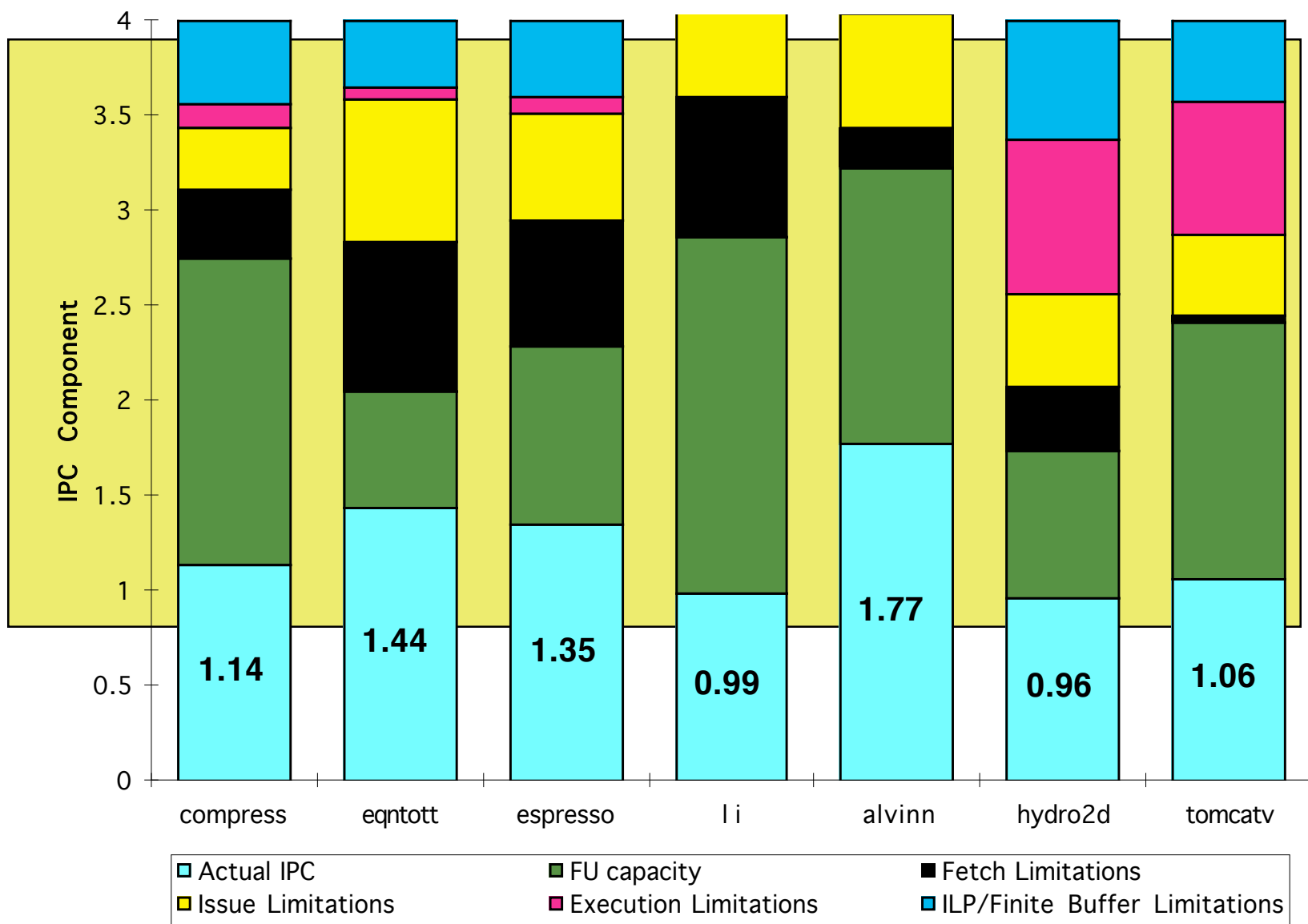
Parallelism is there!

Putting It All Together



Very difficult to get anywhere near the limits!

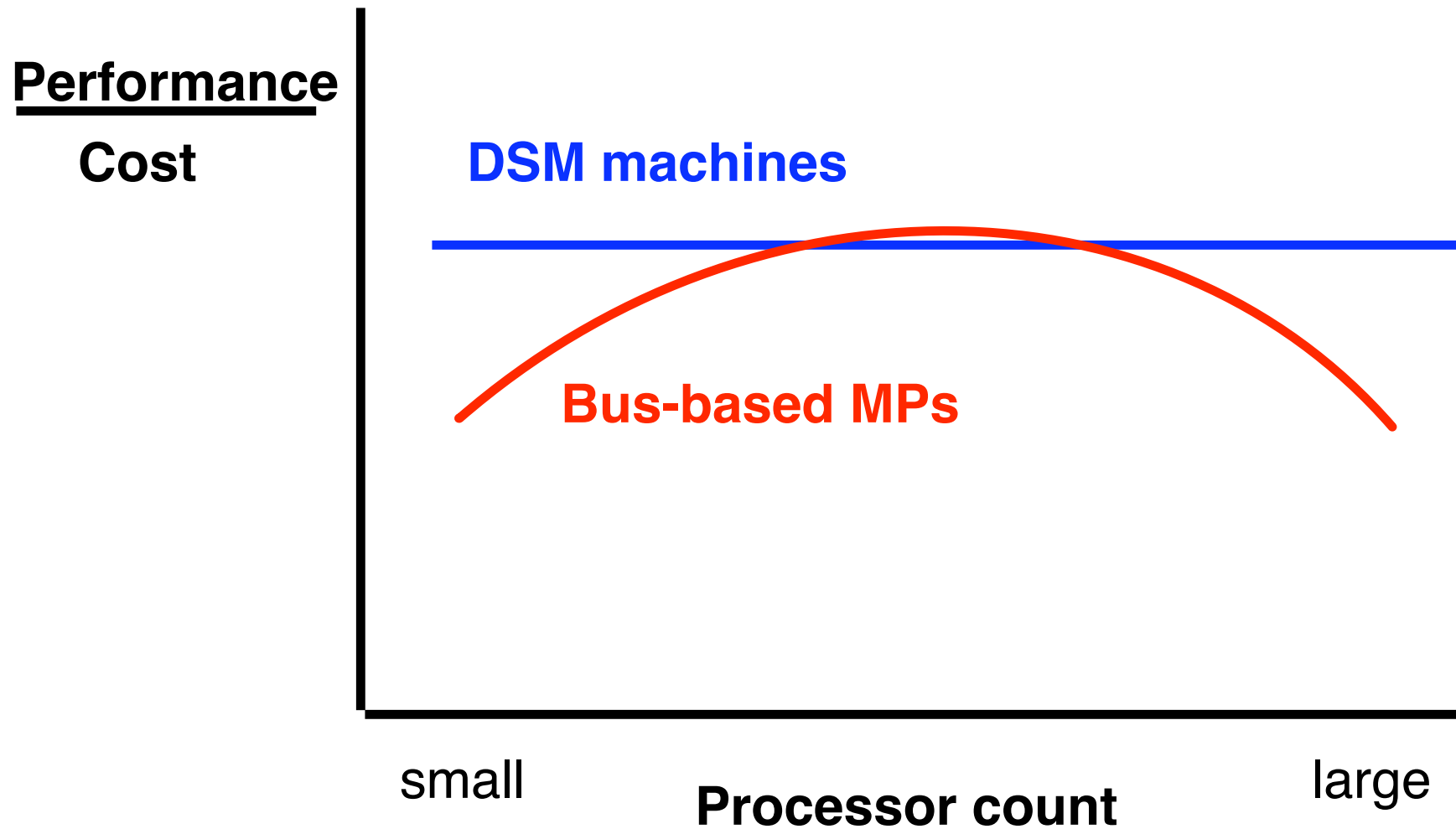
PowerPC 620 Performance



Multiprocessors

- **From 1985 until 1995, two separate approaches:**
 - **Small-scale (4-16) multiprocessors using bus-based interconnect and shared-memory programming model.**
 - **Scalable (64-1,024) multiprocessors using message-passing model**
 - **widespread belief that scalable shared-memory is impossible**
 - **Divergence => no standards, software incompatible**
- **Development of Distributed Shared Memory (DSM)**
 - **Shared memory with cache coherency (all processors think they are sharing one large shared memory)**
 - **Shared-memory programming model with scalability of message-passing approach**
 - **Research prototype demonstrated in 1992.**
 - **DSM is being broadly commercially adopted both for small and large machines**

Why DSM is attractive



Multiprocessor Challenges

- **Finding and conveying parallelism**
 - long-term problem with solid progress since 1985.
- **Remote memory access time:**
 - Time to access memory increasing with each generation:
 - Bus-based machines (1990): 50 cycles
 - DSM experiments (1992): 100 cycles
 - DSM products (1996): 180 cycles
- **Programming challenge**
 - Describe parallel tasks
 - Deal with data locality and distribution
 - Balance programmer effort and performance
 - Good progress in developing compiler technology for this
 - but programmer still must play role
 - Scalable SW is another challenge

New Applications Environments

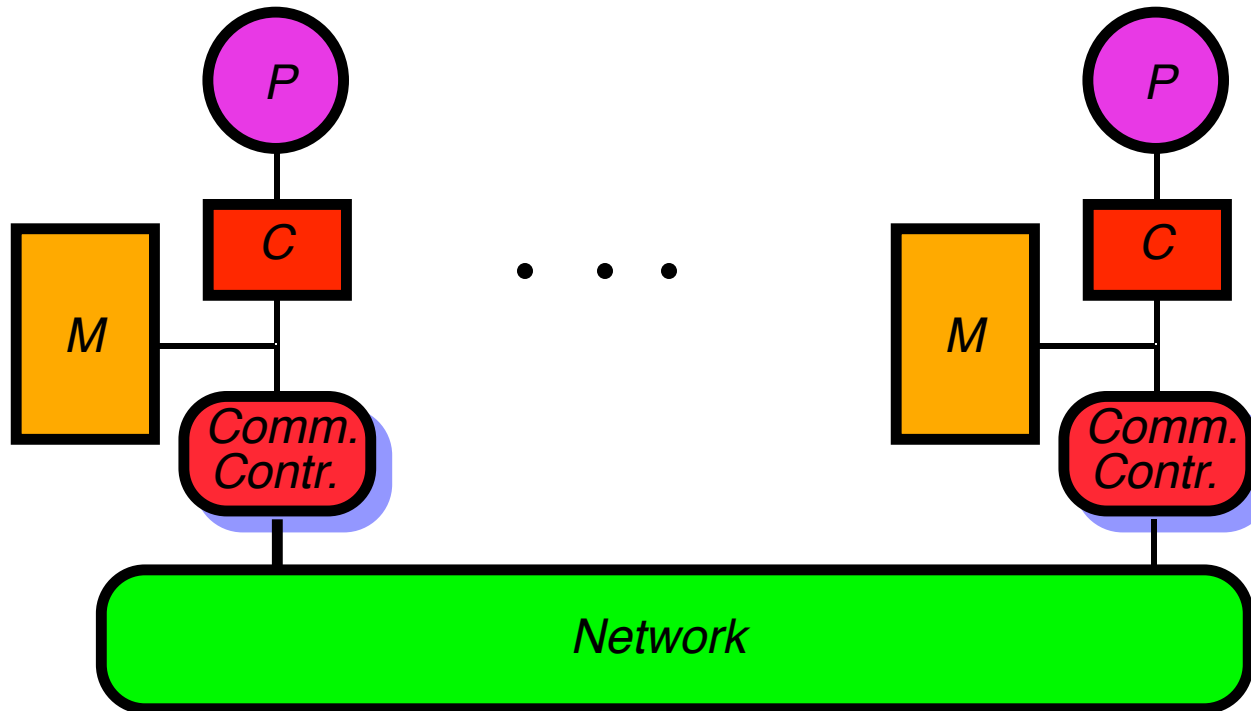
- **Scientific and engineering applications are traditional drivers**
- **New focus: “information intensive” applications**
 - **Large-scale information sources (e.g., data bases, libraries)**
 - **Web servers**
 - **Real-time, high data rate applications (e.g., video)**
 - **Complex multiscale, multidisciplinary simulations**
- **Such applications strain several aspects of systems:**
 - **Memory system**
 - **more data with less predictable access**
 - **caching of I/O systems**
 - **I/O systems**
 - **networks/graphics demand high bandwidth, low latency**

Microprocessors

- **The *single-chip multiprocessor* will be the future**
 - **Consumes transistors without increasing HW complexity**
 - **Challenge: software, software, and architecture**
 - **Chicken-and-egg problem**
 - parallel programming and ubiquitous multiprocessors
 - **What about nonparallel codes?**
 - interesting new work on speculative parallelism
 - **Potential to improve some memory bottlenecks**
 - **The big challenge: take advantage of single-chip integration to build multiprocessors that are**
 - simpler to design,
 - easier to program, and
 - yield better performance

Convergence Architectures: Beyond DSM?

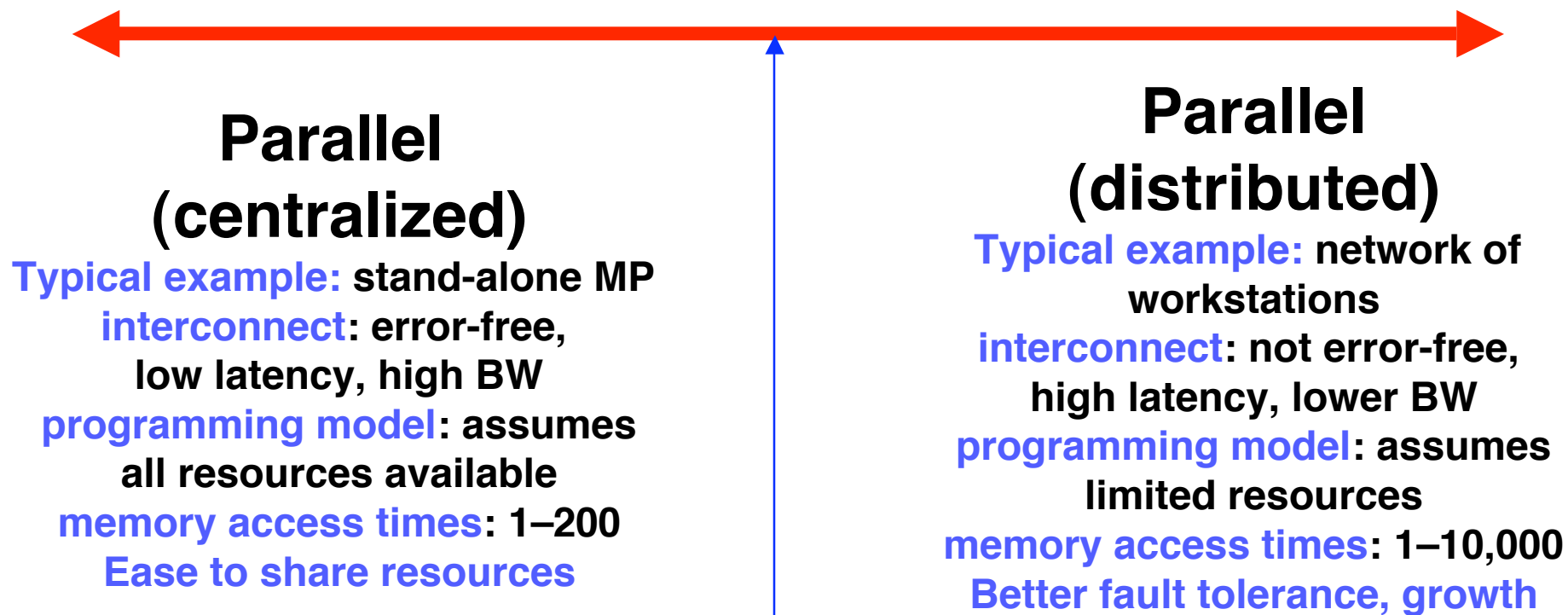
Shared memory, message passing, networks of workstations



Key issue: functionality and performance of the network interface/controller.

Consider: a flexible, programmable controller.

Integration of Distributed & Parallel Approaches?



Is there an intermediate ground, i.e. a “distributed, parallel” system that allows efficient sharing of resources, incremental growth, and is cost effective.

The Future of Computer Systems

- **Past 15 years have shown unrivaled growth in performance and capability.**
- **Next 15 years:**
 - **Continued dominance of microprocessors**
 - **Increasing use of parallelism as key to performance**
 - **Greater awareness of memory and data locality**
 - **Increased need for unified hardware and software technologies.**
 - **Important growth in other aspects of system**
 - **graphics, media capability, user interface, networking, etc.**
- **Beyond 15 years:**
 - **Technology issues may become a major challenge**