CS654 Advanced Computer Architecture

Lec 3 - Introduction

Peter Kemper

Adapted from the slides of EECS 252 by Prof. David Patterson
Electrical Engineering and Computer Sciences
University of California, Berkeley

Outline

- Computer Science at a Crossroads
- Computer Architecture v. Instruction Set Arch.
- **What Computer Architecture brings to table**
- **Technology Trends**

What Computer Architecture brings to Table

- Other fields often borrow ideas from architecture
- Quantitative Principles of Design
  1. Take Advantage of Parallelism
  2. Principle of Locality
  3. Focus on the Common Case
  4. Amdahl's Law
  5. The Processor Performance Equation
- Careful, quantitative comparisons
  – Define, quantify, and summarize relative performance
  – Define and quantify relative cost
  – Define and quantify dependability
  – Define and quantify power
- Culture of anticipating and exploiting advances in technology
- Culture of well-defined interfaces that are carefully implemented and thoroughly checked

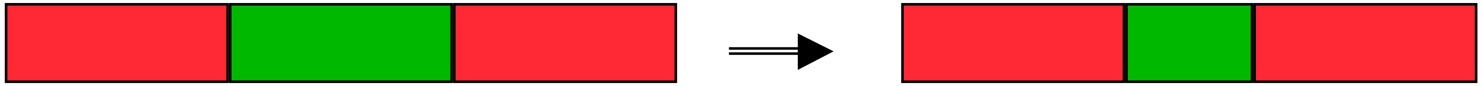
4) Amdahl's Law

$$\text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} \times \left[\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right]$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Best you could ever hope to do:

$$\text{Speedup}_{\text{maximum}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right)}$$



Amdahl’s Law example

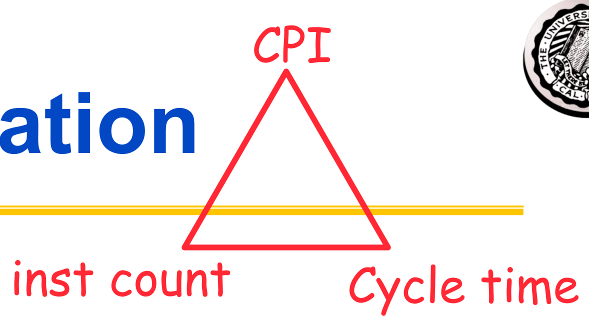
- New CPU 10X faster
- I/O bound server, so 60% time waiting for I/O

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \dfrac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$= \frac{1}{\left(1 - 0.4\right) + \dfrac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

- Apparently, its human nature to be attracted by 10X faster, vs. keeping in perspective its just 1.6X faster

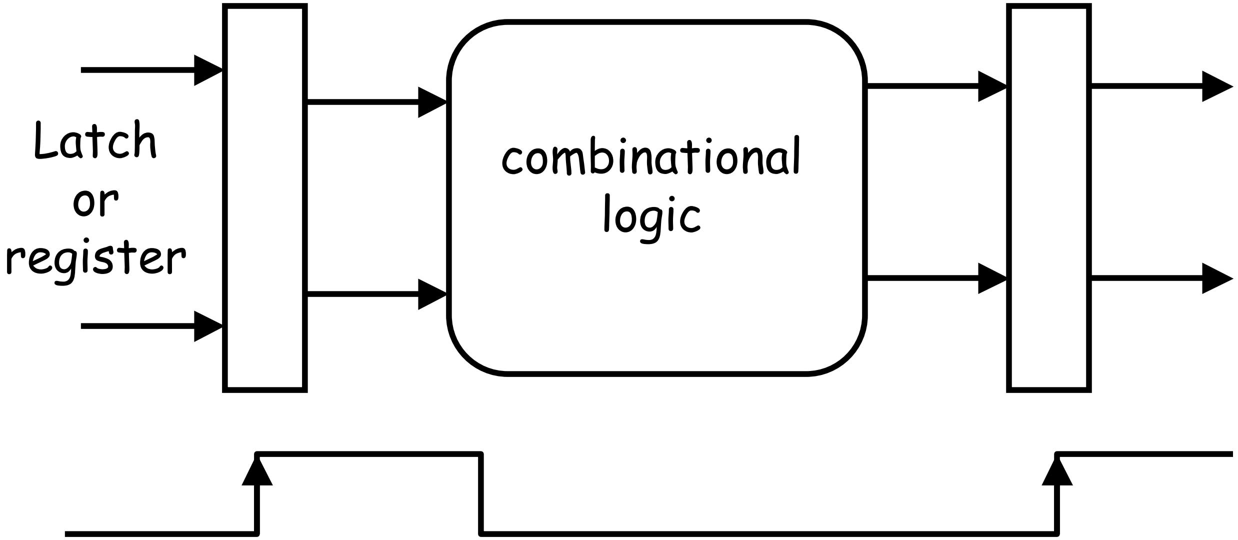
5) Processor performance equation



$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Inst Count	CPI	Clock Rate
Program	X		
Compiler	X	(X)	
Inst. Set.	X	X	
Organization		X	X
Technology			X

What's a Clock Cycle?



- **Old days: 10 levels of gates**
- **Today: determined by numerous time-of-flight issues + gate delays**
    - **clock propagation, wire lengths, drivers**

At this point …

- Computer Architecture >> instruction sets
- Computer Architecture skill sets are different
  - 5 Quantitative principles of design
  - Quantitative approach to design
  - Solid interfaces that really work
  - Technology tracking and anticipation
- Computer Science at the crossroads from sequential to parallel computing
  - Salvation requires innovation in many fields, including computer architecture
- However for CS654, we have to go through the state of the art first:
  - Material:

    read Chapter 1, then Appendix A in Hennessy/Patterson

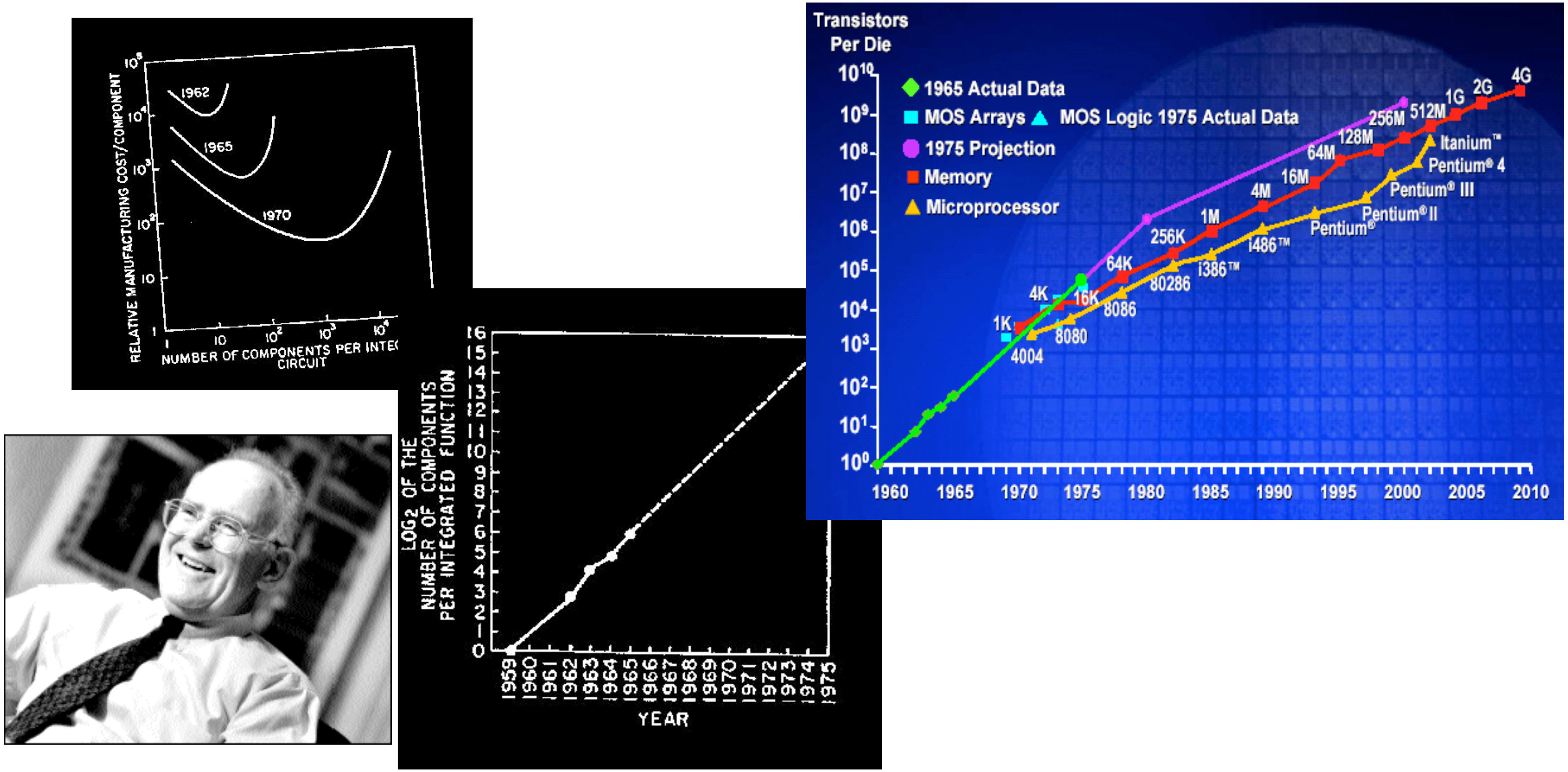
Outline

- Technology Trends: Culture of tracking, anticipating and exploiting advances in technology

- Careful, quantitative comparisons:

   1. Define, quantify, and summarize relative performance
   2. Define and quantify relative cost
   3. Define and quantify dependability
   4. Define and quantify power

Moore’s Law: 2X transistors / “year”



- “Cramming More Components onto Integrated Circuits”
  - Gordon Moore, Electronics, 1965
- # on transistors / cost-effective integrated circuit double every N months (12 ≤ N ≤ 24)

Tracking Technology Performance Trends

- **Drill down into 4 technologies:**
  - Disks,
  - Memory,
  - Network,
  - Processors
- **Compare ~1980 vs. ~2000 technology**
  - Performance Milestones in each technology
- **Compare for Bandwidth vs. Latency improvements in performance over time**
- **Bandwidth: number of events per unit time**
  - E.g., M bits / second over network, M bytes / second from disk
- **Latency: elapsed time for a single event**
  - E.g., one-way network delay in microseconds, average disk access time in milliseconds

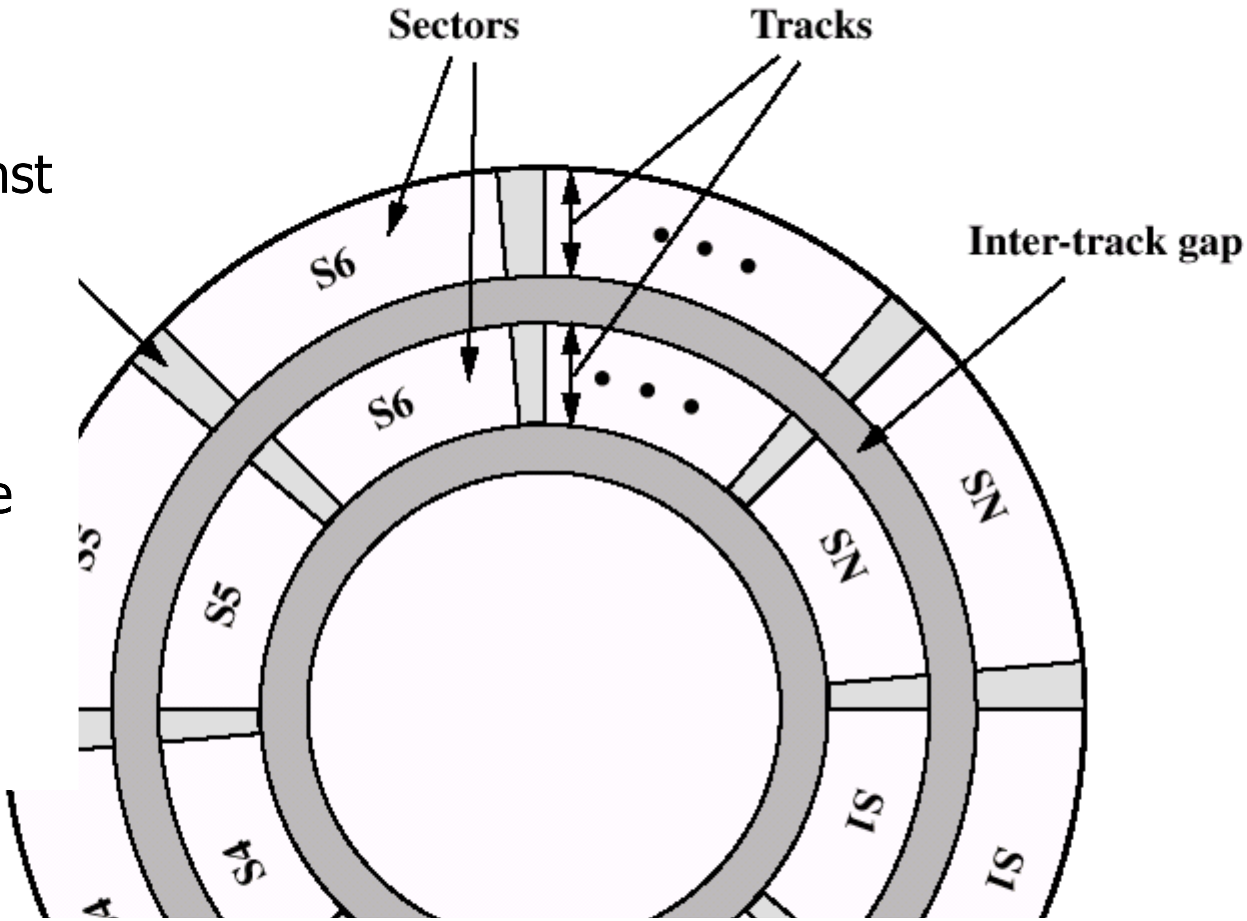
Disks: ~1980 vs ~2000 technology

CDC Wren I, 1983	Seagate 373453, 2003	
3600 RPM	15000 RPM	(4X)
0.03 GBytes capacity	73.4 GBytes	(2500X)
Tracks/Inch: 800	Tracks/Inch: 64000	(80X)
Bits/Inch: 9550	Bits/Inch: 533,000	(60X)
Three 5.25” platters	Four 2.5” platters (in 3.5” form factor)	
Bandwidth: 0.6 MBytes/sec	Bandwidth: 86 MBytes/sec	(140X)
Latency: 48.3 ms	Latency: 5.7 ms	(8X)
Cache: none	Cache: 8 MBytes	

Hard disk

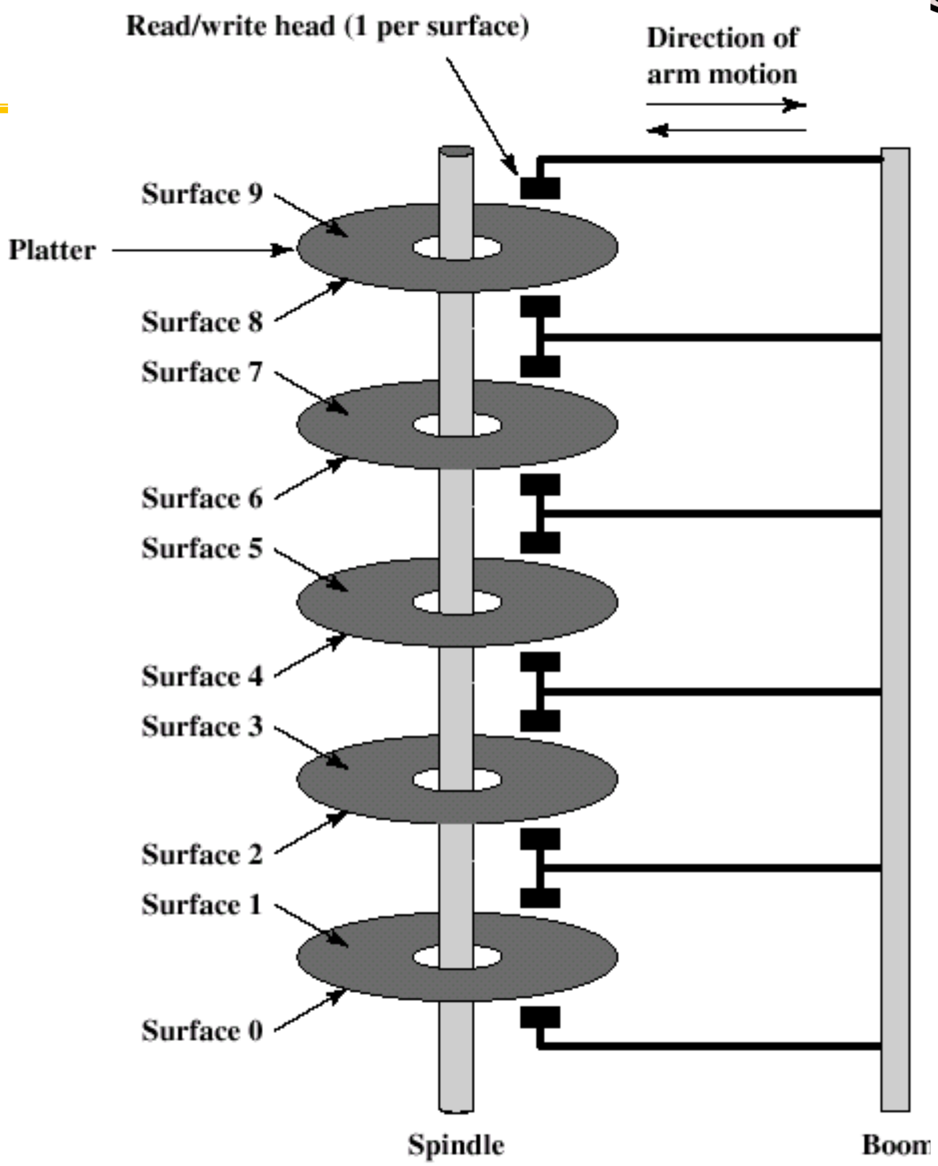
Track: Ring with data
Partitioned into sectors of same size
Virtual Geometry (for OS):
x cylinders, y heads, z sectors eg Pentium-PC, max x=65535, y=16, z=63
Alternative: logical block addressing (LBA): 0,1,…, sectors

Physical Geometry
(intern for controller):
old: #sectors/track const
now: n zones
(eg n=16),
In each zone #sectors
per track same.
Outer zones have more
than innner..
Figure:
virtuell->physical
by controller



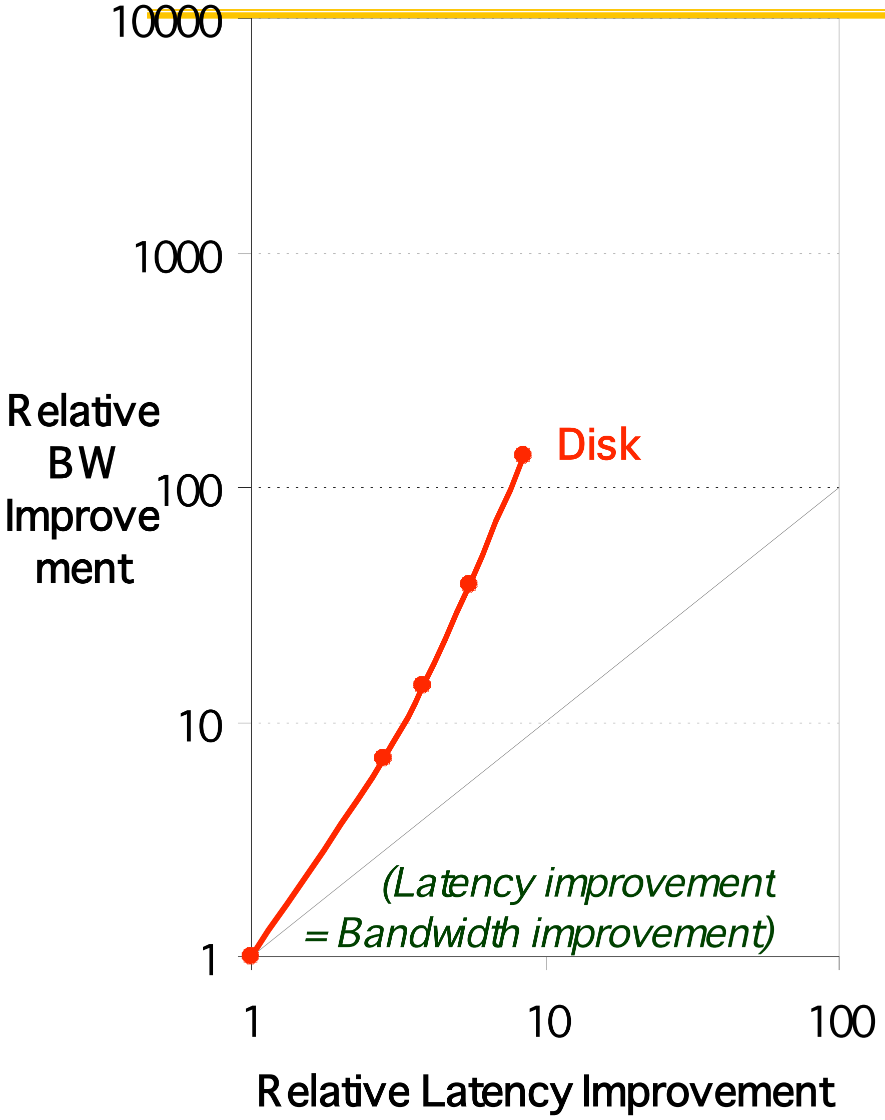
Hard disk

- disks in vertikal order, moving together,
- rotation speed in rpm is const (eg IDE 7200 rpm, SCSI 10000, 15000 rpm),
- Read/write heads moved together, access same track
-> cylinder, i.e. all tracks with same distance to center
- data up to 500 GB



Transfer times for sequential and random access patterns differ significantly due to seek time!

Latency Lags Bandwidth (for last ~20 years)



- **Performance Milestones**

- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** **(8x, 143x)**

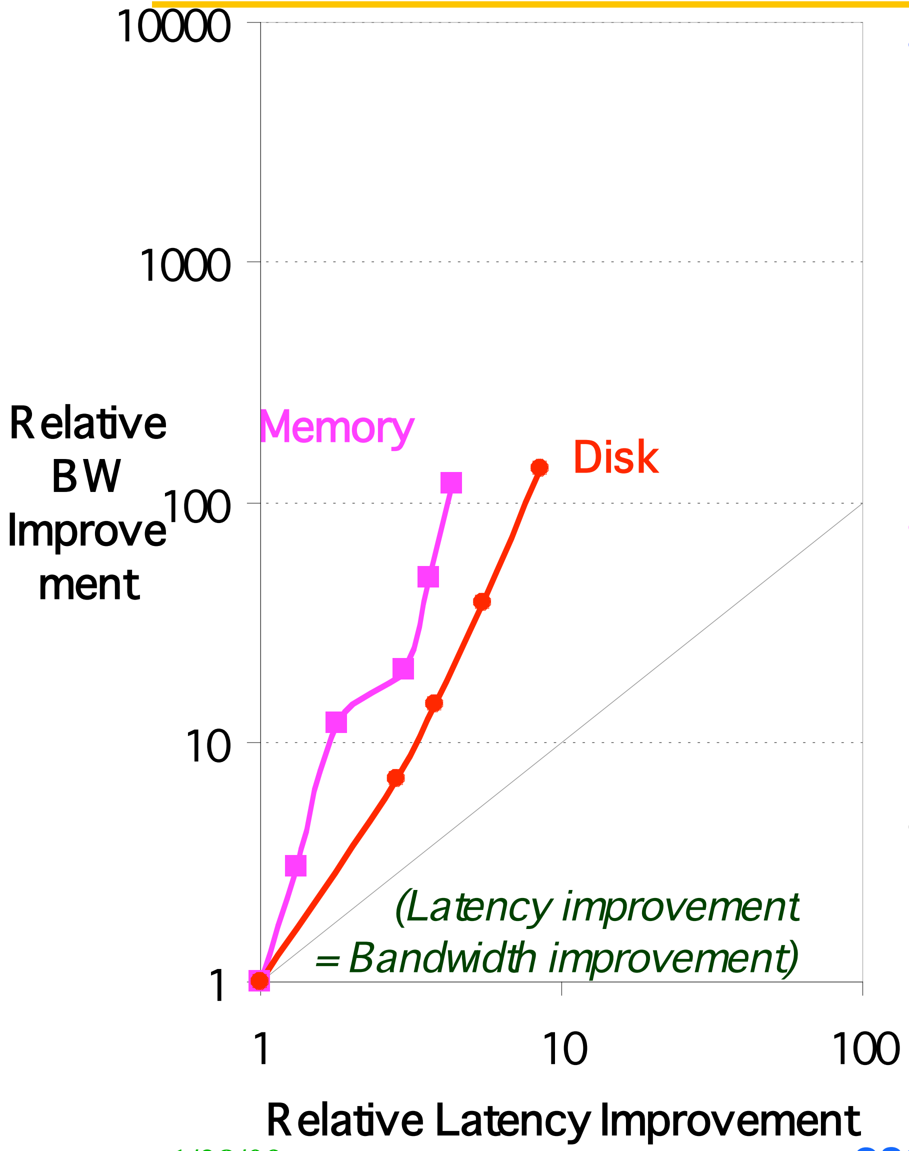
  (latency = simple operation w/o contention
  BW = best-case)

Memory: ~1980 vs ~2000 technology

- 1980 DRAM (asynchronous)
- 0.06 Mbits/chip
- 64,000 xtors, 35 mm^2
- 16-bit data bus per module, 16 pins/chip
- 13 Mbytes/sec
- Latency: 225 ns
- (no block transfer)

- 2000 Double Data Rate Synchr. (clocked) DRAM
- 256.00 Mbits/chip (4000X)
- 256,000,000 xtors, 204 mm^2
- 64-bit data bus per DIMM, 66 pins/chip (4X)
- 1600 Mbytes/sec (120X)
- Latency: 52 ns (4X)
- Block transfers (page mode)

Latency Lags Bandwidth (last ~20 years)



- **Performance Milestones**
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** **(4x,120x)**
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** **(8x, 143x)**

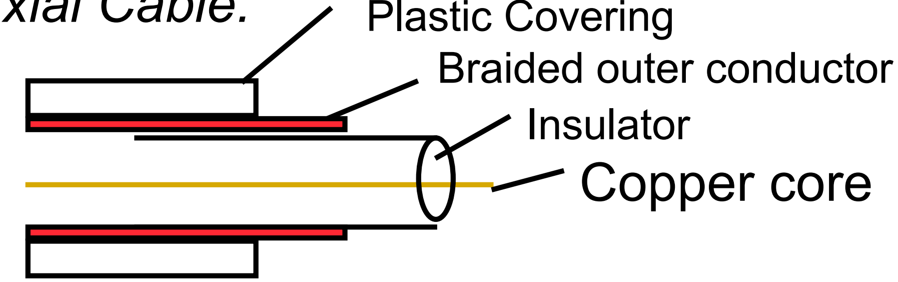
(latency = simple operation w/o contention
BW = best-case)

LANs: ~1980 vs. ~2000 technology

- Ethernet 802.3
- Year of Standard: 1978
- 10 Mbits/s link speed
- Latency: 3000 μsec
- Shared media
- Coaxial cable

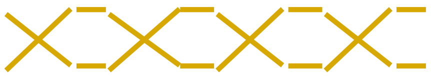
- Ethernet 802.3ae
- Year of Standard: 2003
- 10,000 Mbits/s (1000X) link speed
- Latency: 190 μsec (15X)
- Switched media
- Category 5 copper wire

Coaxial Cable:



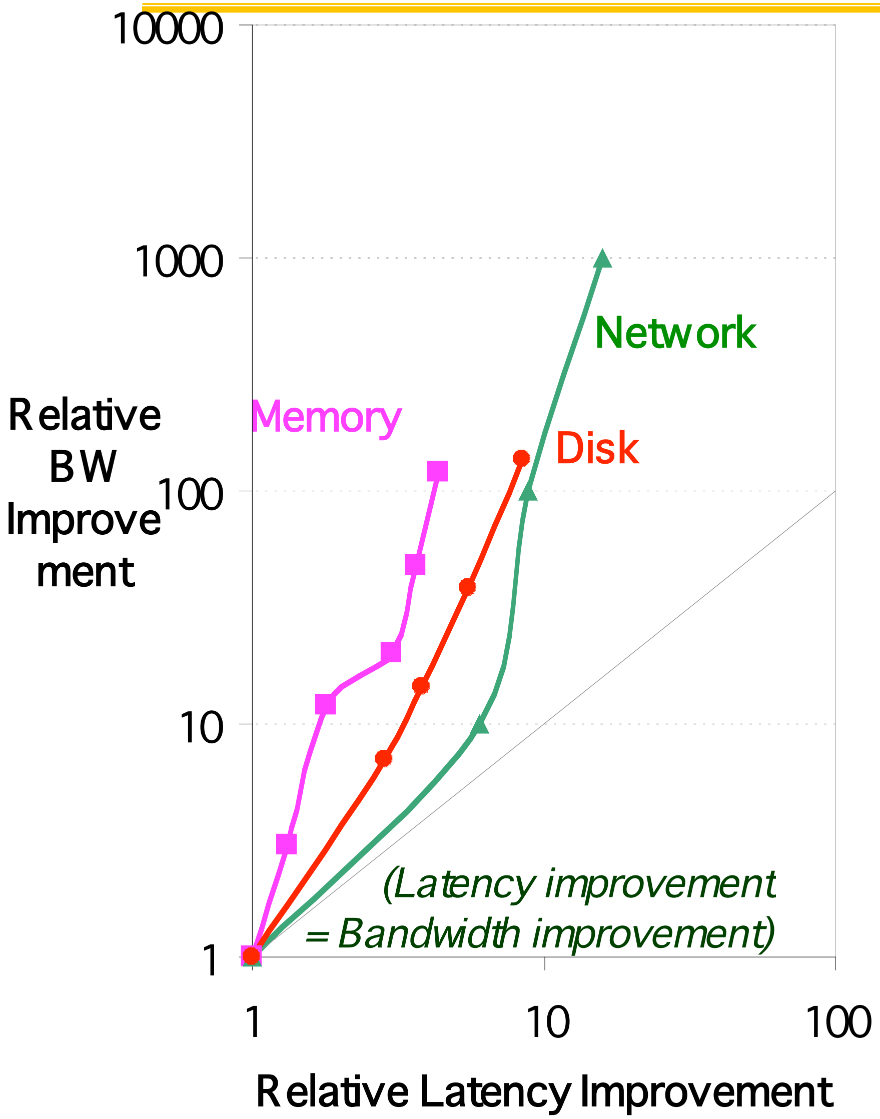
"Cat 5" is 4 twisted pairs in bundle

Twisted Pair:



Copper, 1mm thick,
twisted to avoid antenna effect

Latency Lags Bandwidth (last ~20 years)



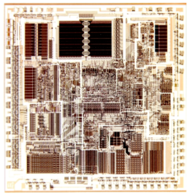
- **Performance Milestones**

- **Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s** (16x,1000x)
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** (4x,120x)
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

(latency = simple operation w/o contention
BW = best-case)

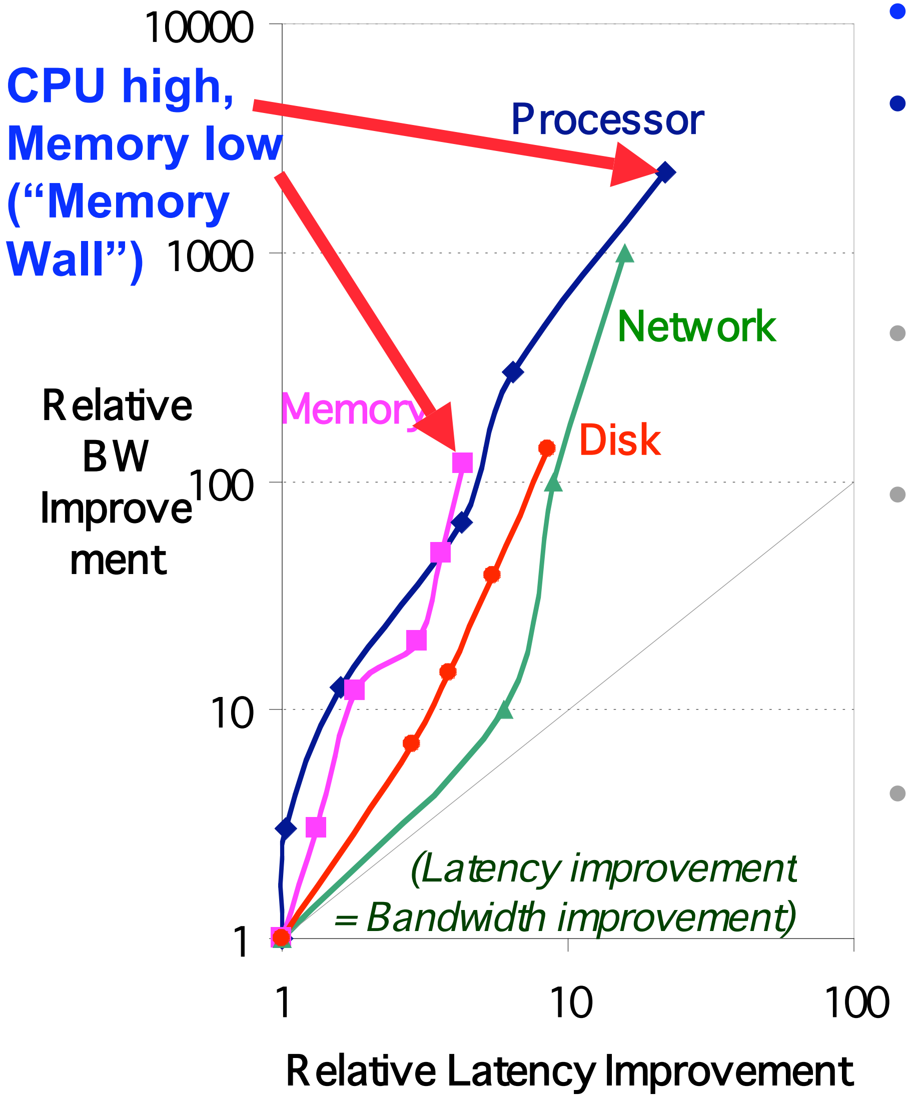
CPUs: ~1980 vs. ~2000 technology

- 1982 Intel 80286
- 12.5 MHz
- 2 MIPS (peak)
- Latency 320 ns
- 134,000 xtors, 47 mm^2
- 16-bit data bus, 68 pins
- Microcode interpreter, separate FPU chip
- (no caches)



- 2001 Intel Pentium 4
- 1500 MHz (120X)
- 4500 MIPS (peak) (2250X)
- Latency 15 ns (20X)
- 42,000,000 xtors, 217 mm^2
- 64-bit data bus, 423 pins
- 3-way superscalar, Dynamic translate to RISC, Superpipelined (22 stage), Out-of-Order execution
- On-chip 8KB Data caches, 96KB Instr. Trace cache, 256KB L2 cache

Latency Lags Bandwidth (last ~20 years)



- **Performance Milestones**
- **Processor: '286, '386, '486, Pentium, Pentium Pro, Pentium 4** (21x,2250x)
- Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s (16x,1000x)
- Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
- Disk : 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

Rule of Thumb for Latency Lagging BW

- **In the time that bandwidth doubles, latency improves by no more than a factor of 1.2 to 1.4**

  (and capacity improves faster than bandwidth)

- **Stated alternatively:**
  Bandwidth improves by more than the square of the improvement in Latency

Computers in the News

- “Intel loses market share in own backyard,” By Tom Krazit, CNET News.com, 1/18/2006
- “Intel's share of the U.S. retail PC market fell by 11 percentage points, from 64.4 percent in the fourth quarter of 2004 to 53.3 percent. … Current Analysis' market share numbers measure U.S. retail sales only, and therefore exclude figures from Dell, which uses its Web site to sell directly to consumers. …
  AMD chips were found in 52.5 percent of desktop PCs sold in U.S. retail stores during that period.”
- Technical advantages of AMD Opteron/Athlon vs. Intel Pentium 4 as we’ll see in this course.

6 Reasons Latency Lags Bandwidth

1. Moore's Law helps BW more than latency

- **Faster transistors, more transistors, more pins help Bandwidth**

»	**MPU Transistors:**	**0.130 vs. 42 M xtors**	**(300X)**
»	**DRAM Transistors:**	**0.064 vs. 256 M xtors**	**(4000X)**
»	**MPU Pins:**	**68 vs. 423 pins**	**(6X)**
»	**DRAM Pins:**	**16 vs. 66 pins**	**(4X)**

- **Smaller, faster transistors but communicate over (relatively) longer lines: limits latency**

»	**Feature size:**	**1.5 to 3 vs. 0.18 micron**	**(8X,17X)**
»	**MPU Die Size:**	**35 vs. 204 mm^2**	**(ratio sqrt ⇒ 2X)**
»	**DRAM Die Size:**	**47 vs. 217 mm^2**	**(ratio sqrt ⇒ 2X)**

6 Reasons Latency Lags Bandwidth (cont’d)

2. Distance limits latency

- **Size of DRAM block ⇒ long bit and word lines ⇒ most of DRAM access time**
- **Speed of light and computers on network**
- **1. & 2. explains linear latency vs. square BW?**

3. Bandwidth easier to sell (“bigger=better”)

- **E.g., 10 Gbits/s Ethernet (“10 Gig”) vs.**
  10 μsec latency Ethernet
- **4400 MB/s DIMM (“PC4400”) vs. 50 ns latency**
- **Even if just marketing, customers now trained**
- **Since bandwidth sells, more resources thrown at bandwidth, which further tips the balance**

6 Reasons Latency Lags Bandwidth (cont'd)

4. Latency helps BW, but not vice versa

- **Spinning disk faster improves both bandwidth and rotational latency**
  - » **3600 RPM ⇒ 15000 RPM = 4.2X**
  - » **Average rotational latency: 8.3 ms ⇒ 2.0 ms**
  - » **Things being equal, also helps BW by 4.2X**
- **Lower DRAM latency ⇒**
  More access/second (higher bandwidth)
- **Higher linear density helps disk BW**
  (and capacity), but not disk Latency
  - » **9,550 BPI ⇒ 533,000 BPI ⇒ 60X in BW**

6 Reasons Latency Lags Bandwidth (cont'd)

5. Bandwidth hurts latency

- Queues help Bandwidth, hurt Latency (Queuing Theory)
- Adding chips to widen a memory module increases Bandwidth but higher fan-out on address lines may increase Latency

6. Operating System overhead hurts Latency more than Bandwidth

- Long messages amortize overhead; overhead bigger part of short messages

Summary of Technology Trends

- For disk, LAN, memory, and microprocessor, bandwidth improves by square of latency improvement
  - In the time that bandwidth doubles, latency improves by no more than 1.2X to 1.4X
- Lag probably even larger in real systems, as bandwidth gains multiplied by replicated components
  - Multiple processors in a cluster or even in a chip
  - Multiple disks in a disk array
  - Multiple memory modules in a large memory
  - Simultaneous communication in switched LAN
- HW and SW developers should innovate assuming Latency Lags Bandwidth
  - If everything improves at the same rate, then nothing really changes
  - When rates vary, require real innovation