CMSC 411
Computer Systems Architecture
Lecture 3
Performance Measurement and Reliability

Alan Sussman
als@cs.umd.edu

Administrivia

- Homework problems for Unit 1 posted today
  - due next Thursday, 2/12
- Start reading Appendix C – Basic Pipelining
- Appendix K useful as a MIPS ISA reference
  - worth reading, but we'll only touch on parts of it in lecture as needed
- And Appendix A useful as a reference on basic features of instruction sets

PERFORMANCE MEASUREMENT

Amdahl's Law (cont.)

- Suppose that the original task runs for 1 second so it takes *f* seconds in the critical piece, and *1-f* in other things
- Then the task on the improved machine will take only *f/s* seconds in the critical piece, but will still take *1-f* seconds in other things
- speedup = $\frac{old\_time}{new\_time} = \frac{1}{(1-f) + \frac{f}{s}}$

Example 1

- Suppose we work very hard improving the square root hardware, and that our task originally spends *1%* of its time doing square roots. Even if the improvement reduces the square root time to zero, the speedup is no better than
- speedup = $\frac{1}{1-f}$ = $\frac{1}{.99}$ = 1.01
- *And we might be better off putting our effort into a more important part of the task*

Example 2

- Suppose that, for the same cost, we can speed up integer arithmetic by a factor of 20, or speed up floating point arithmetic by a factor of 2. If our task spends *10%* of its time in integer arithmetic, and *40%* of its time in floating point arithmetic, which should we do?

Example 2 (cont.)

- Option 1:
  speedup = $\frac{1}{.9+\frac{.1}{20}}$ = 1.105
- Option 2:
  speedup = $\frac{1}{.6+\frac{.4}{2}}$ = 1.25

CPU Performance

- More jargon …
- Something new happens in a computer during every clock cycle
- The clock rate is what manufacturers usually advertise to indicate a chip's speed:
  - e.g., a 2.8GHz Intel i5
- But how fast the machine is depends on the clock rate *and* the number of clock cycles per instruction (CPI)

CPU Performance (cont.)

- So total CPU time = instruction count (IC) times CPI divided by clock rate (MHz/GHz)
  - IC * CPI / GHz
- There's an example on p. 50 that illustrates the overall effect
- Note: CPI is not a good measure of performance all by itself since it varies by type of instruction!

How to design a fast computer

- Amdahl's law says put your effort into optimizing instructions that are often used.
- The locality of reference rule-of-thumb says that a program spends *90%* of its time in *10%* of the code, so we should put effort into taking advantage of this, through a memory hierarchy
- For data accesses, 2 types of locality
  - temporal
  - spatial

Take advantage of parallelism

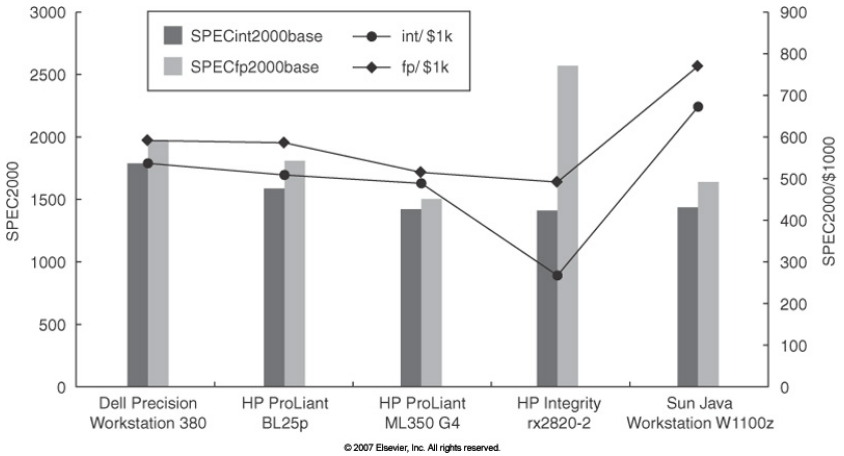
- At system level
  - e.g., use multiple CPUs (Unit 6), disks (Unit 7)
- At processor level
  - among instructions (Unit 5)
  - pipelining (Unit 3)
- At digital design level
  - e.g., set associative caches (Unit 4), carry-lookahead in ALU design

Performance/Price Performance

For desktop systems – 1GB ECC SDRAM, August 2005

Model	Processor	Clock rate (GHz)	Price
Dell 380	Intel PIV Xeon	3.8	$3346
HP BL25p	AMD Opteron 252	2.6	$3099
HP ML350 G4	Intel PIV Xeon	3.4	$2907
HP rx2620-2	Intel Itanium 2	1.6	$5201
Sun Java WS W1100z	AMD Opteron 150	2.4	$2145

SPEC CINT2000 & CFP2000

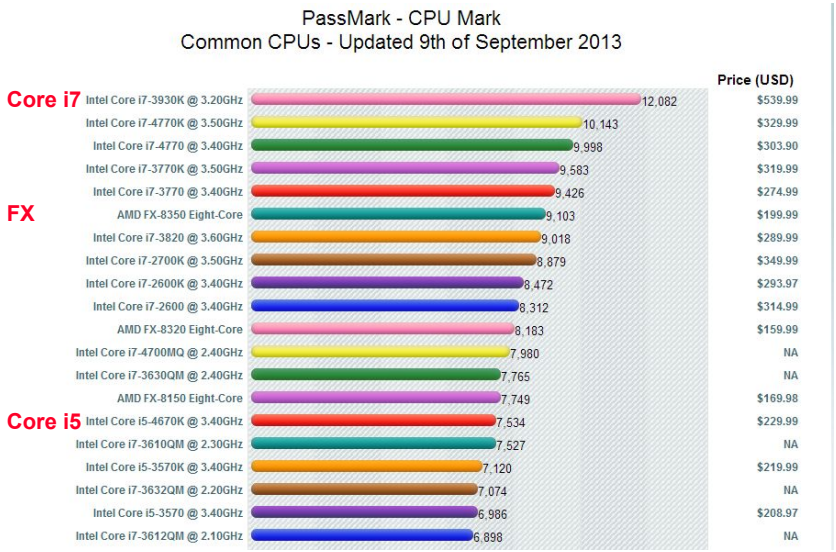


PassMark PerformanceTest Benchmark

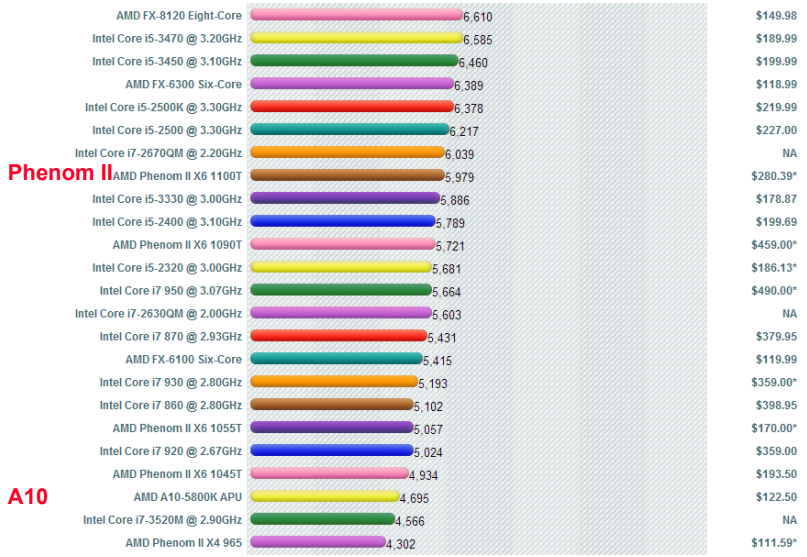
- **Modern benchmark suite for PCs**
- **5 main test suites**
  - **CPU tests**
    - » **Math, compression, encryption, graphics**
  - **2D graphics**
    - » **Drawing lines, bitmaps, fonts**
  - **3D graphics**
    - » **DirectX 3D graphics & animations**
  - **Disk**
    - » **Read, write, seek**
  - **Memory**
    - » **Allocation & accesses**

PASSMARK® SOFTWARE
PerformanceTest™
Easy PC benchmarking

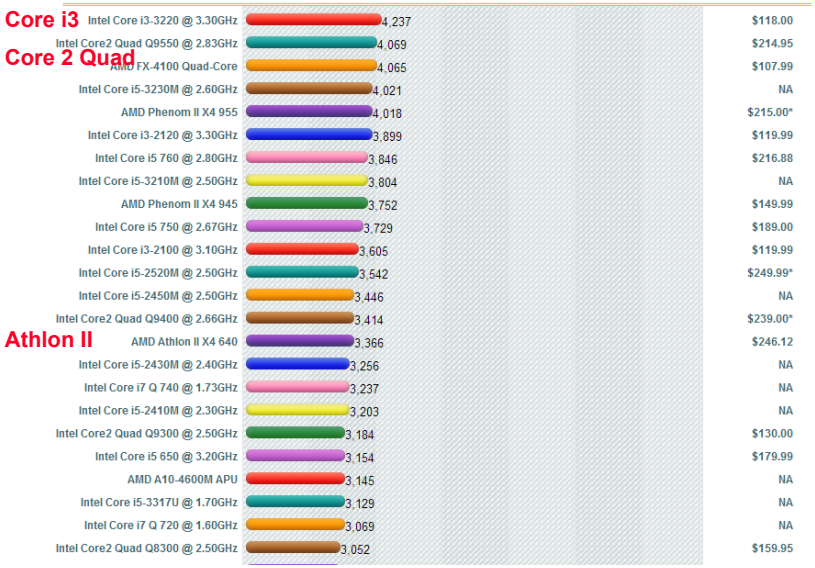
PassMark Results for Common CPUs



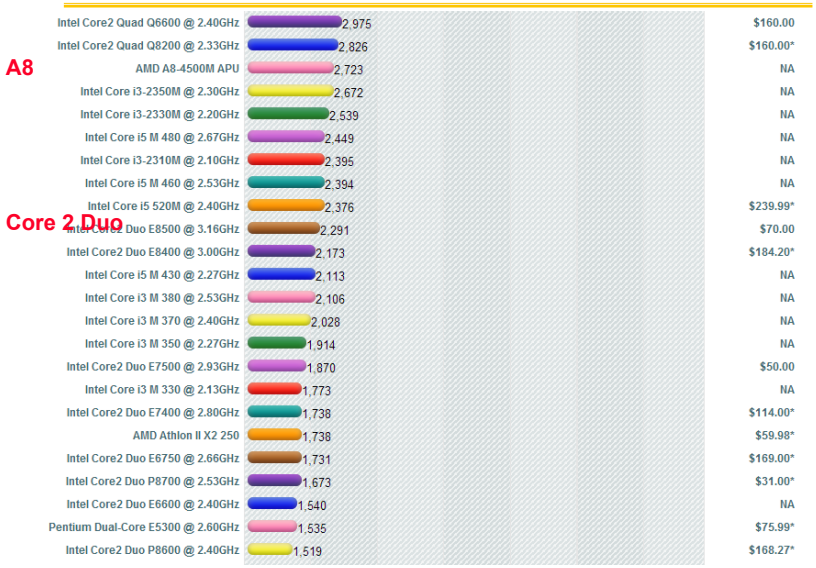
PassMark Results for Common CPUs



PassMark Results for Common CPUs



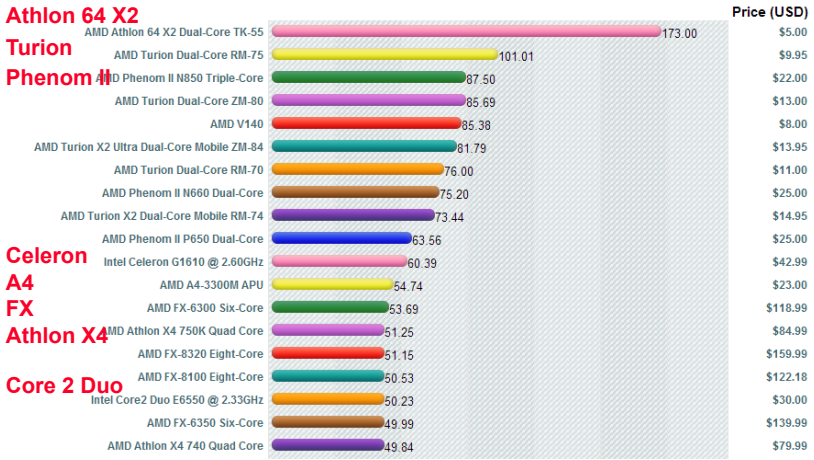
PassMark Results for Common CPUs



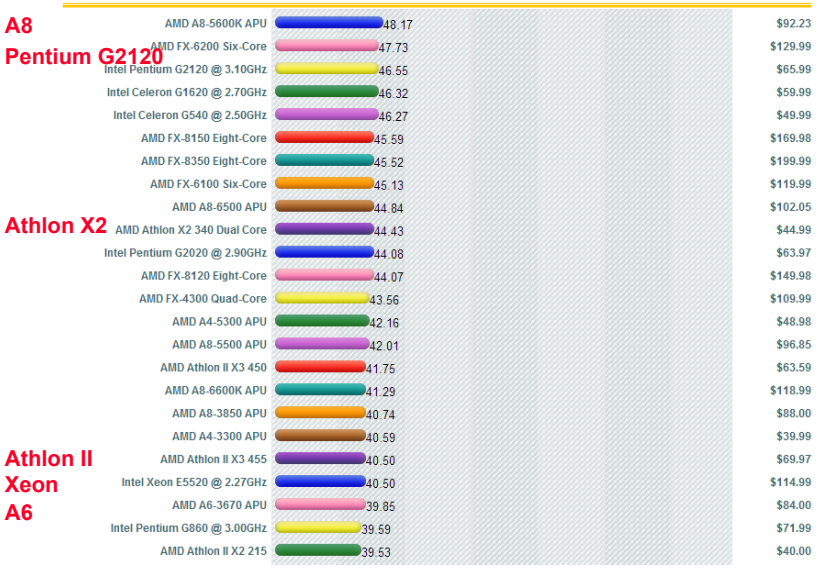
PassMark Results for Common CPUs

PassMark CPU Price / Performance

PassMark - Price Performance (CPU Mark / $Price)
Top 100 Available Price Performance CPUs - Updated 9th of September 2013



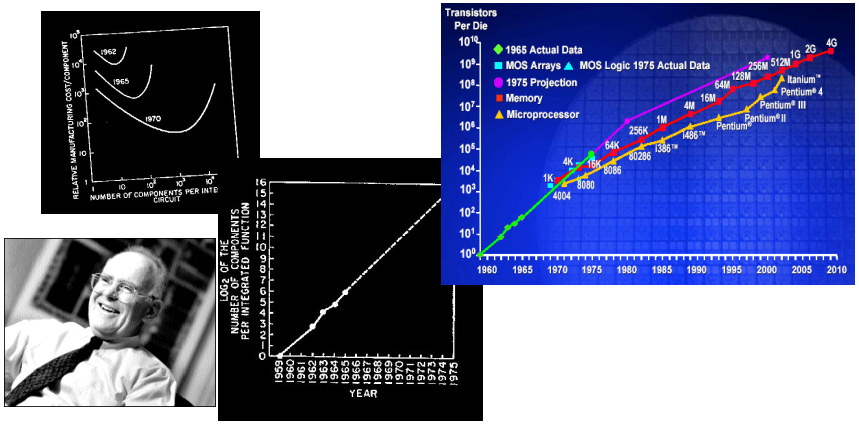
PassMark CPU Price / Performance



Outline

- Moore's Law and Technology Trends
- Reliability and MTTF

Moore's Law: 2X transistors / "year"



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

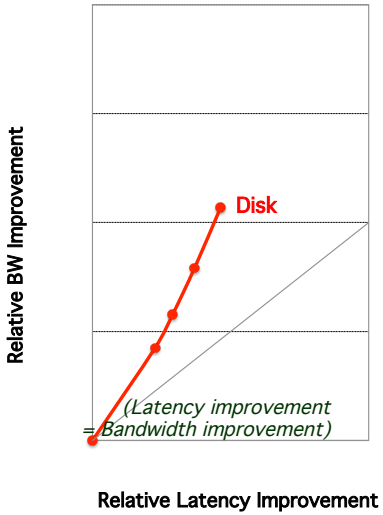
Tracking Technology Performance Trends

- Drill down into 4 technologies:
  - Disks,
  - Memory,
  - Network,
  - Processors
- Compare ~1980 Archaic (Nostalgic) vs. ~2000 Modern (Newfangled)
  - Performance Milestones in each technology
- Compare for Bandwidth vs. Latency improvements in performance over time
- Bandwidth: number of events per unit time
  - E.g., Mbits / second over network, Mbytes / 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: Archaic(Nostalgic) v. Modern(Newfangled)

• 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	

Latency Lags Bandwidth (for last ~20 years)



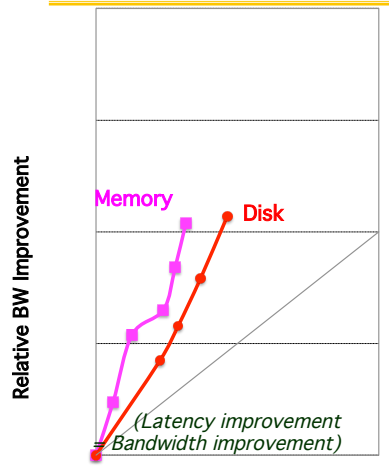
- **Performance Milestones**
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

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

Memory: Archaic (Nostalgic) v. Modern (Newfangled)

• 1980 DRAM (asynchronous)	• 2000 Double Data Rate Synchr. (clocked) DRAM	
• 0.06 Mbits/chip	• 256.00 Mbits/chip	(4000X)
• 64,000 xtors, 35 mm^2	• 256,000,000 xtors, 204 mm^2	
• 16-bit data bus per module, 16 pins/chip	• 64-bit data bus per DIMM, 66 pins/chip	(4X)
• 13 Mbytes/sec	• 1600 Mbytes/sec	(120X)
• Latency: 225 ns	• Latency: 52 ns	(4X)
• (no block transfer)	• Block transfers (page mode)	

Latency Lags Bandwidth (last ~20 years)



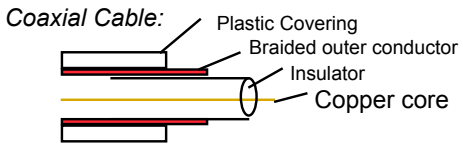
- **Performance Milestones**
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** (4x,120x)
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

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

LANs: Archaic (Nostalgic)v. Modern (Newfangled)

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

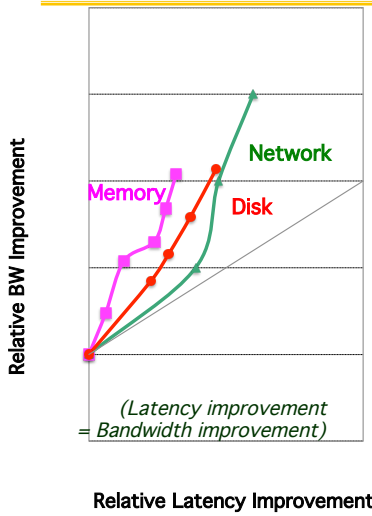
- Ethernet 802.3ae
- Year of Standard: 2003
- 10,000 Mbits/s (1000X) link speed
- Latency: 190 μsec (15X)
- Switched media
- Category 5 copper wire



"Cat 5" is 4 twisted pairs in bundle

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)

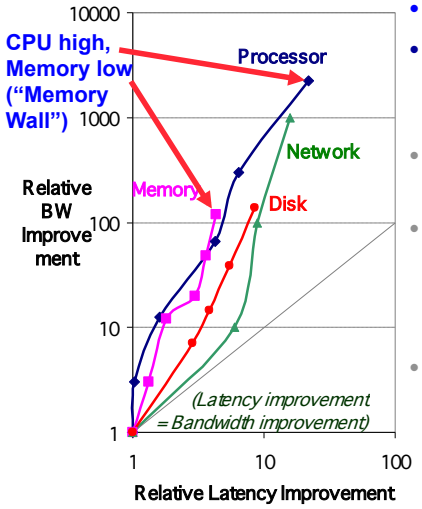
(latency = simple operation w/o contention
BW = best-case)

CPUs: Archaic (Nostalgic) v. Modern (Newfangled)

- 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 (over ~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

6 Reasons Latency Lags Bandwidth

1. Moore's Law helps BW more than latency

- **Faster transistors, more transistors, more pins help Bandwidth**
  - » **CPU Transistors: 0.130 vs. 42 M xtors (300X)**
  - » **DRAM Transistors: 0.064 vs. 256 M xtors (4000X)**
  - » **CPU Pins: 68 vs. 423 pins (6X)**
  - » **DRAM Pins: 16 vs. 66 pins (4X)**
- **Smaller, faster transistors but communicate over (relatively) longer wires: limits latency**
  - » **Feature size: 1.5 to 3 vs. 0.18 micron (8X,17X)**
  - » **CPU 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**

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**