Why Latency Lags Bandwidth, and What it Means to Computing

David Patterson

U.C. Berkeley

patterson@cs.berkeley.edu

October 2004

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 FEB 2005	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Why Latency Lags Bandwidth, and What it Means to Computing	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **RU.C. Berkeley**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004. , The original document contains color images.

14. ABSTRACT

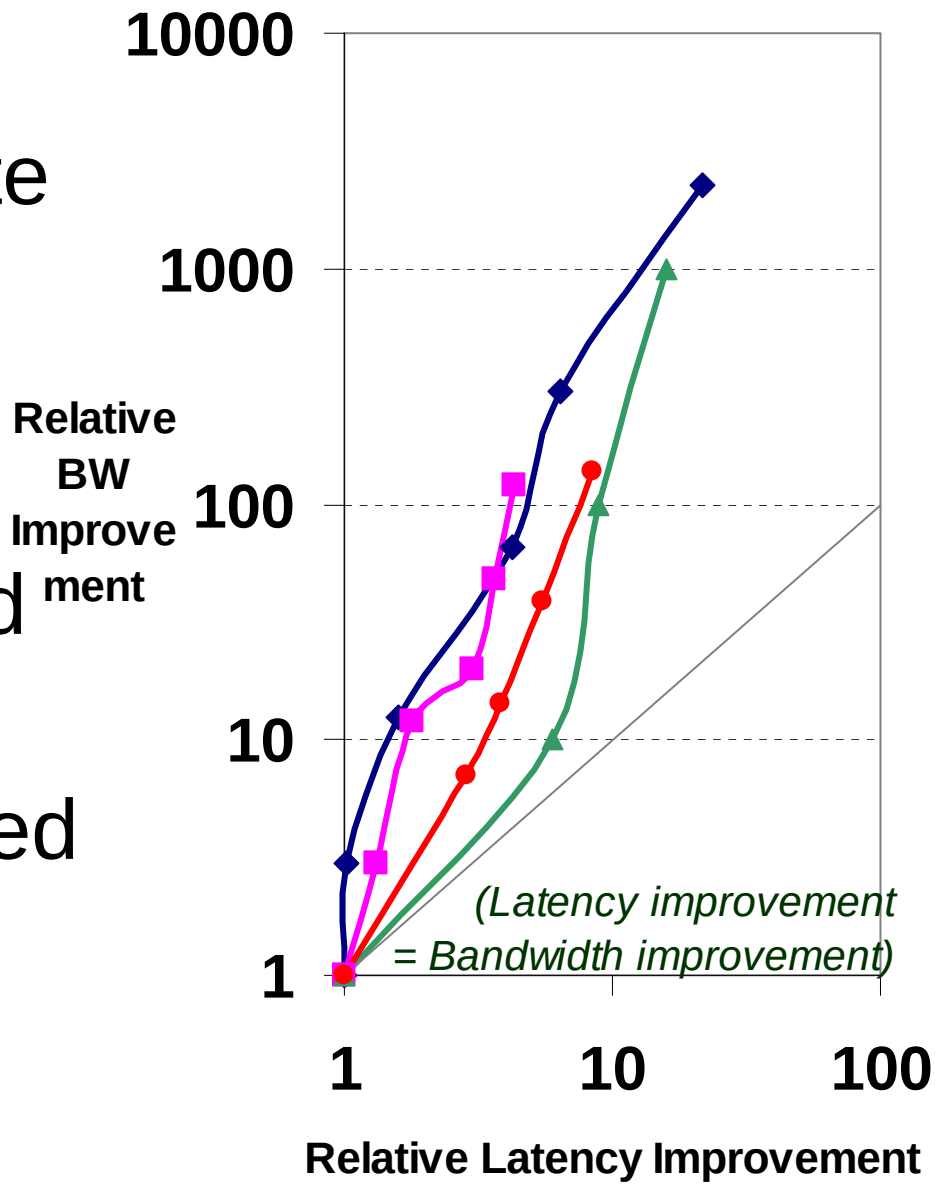
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**25**	

Preview: Latency Lags Bandwidth

Over last 20 to 25 years, for 4 disparate technologies, Latency Lags Bandwidth:

- Bandwidth Improved 120X to 2200X
- But Latency Improved only 4X to 20X
- Talk explains why and how to cope



Outline

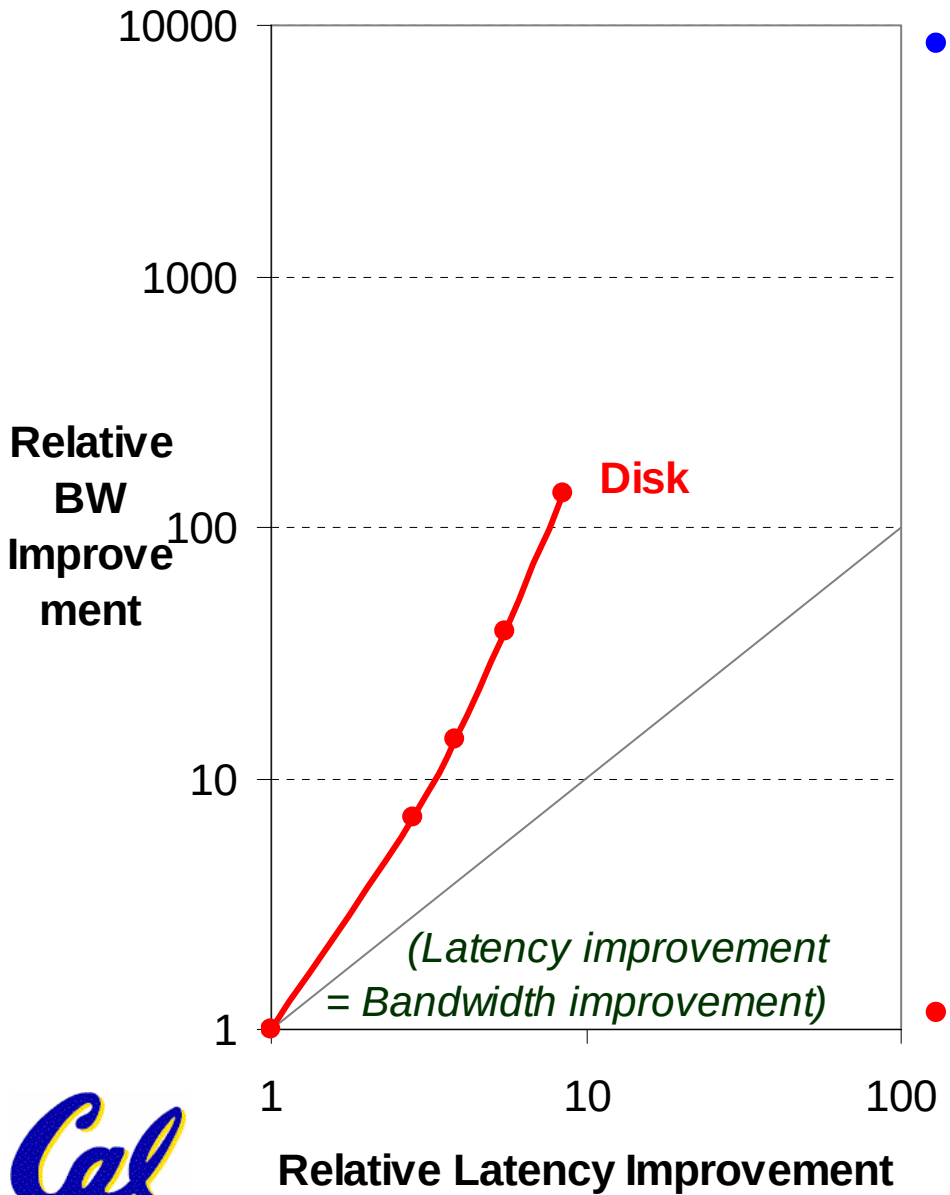
- Drill down into 4 technologies:
  ~1980 Archaic (Nostalgic) vs.
  ~2000 Modern (Newfangled)
  - Performance Milestones in each technology
- Rule of Thumb for BW vs. Latency
- 6 Reasons it Occurs
- 3 Ways to Cope
- 2 Examples BW-oriented system design
- Is this too Optimistic (its even Worse)?
- FYI: “Latency Labs Bandwidth” appears in October, 2004 *Communications of ACM*

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

- CDC Wren I, 1983
- 3600 RPM
- 0.03 GBytes capacity
- Tracks/Inch: 800
- Bits/Inch: 9550
- Three 5.25” platters
- Bandwidth: 0.6 MBytes/sec
- Latency: 48.3 ms
- Cache: none

- Seagate 373453, 2003
- 15000 RPM (4X)
- 73.4 GBytes (2500X)
- Tracks/Inch: 64000 (80X)
- Bits/Inch: 533,000 (60X)
- Four 2.5” platters (in 3.5” form factor)
- Bandwidth: 86 MBytes/sec (140X)
- Latency: 5.7 ms (8X)
- Cache: 8 MBytes

Latency Lags Bandwidth (for last ~20 years)



- Performance Milestones

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

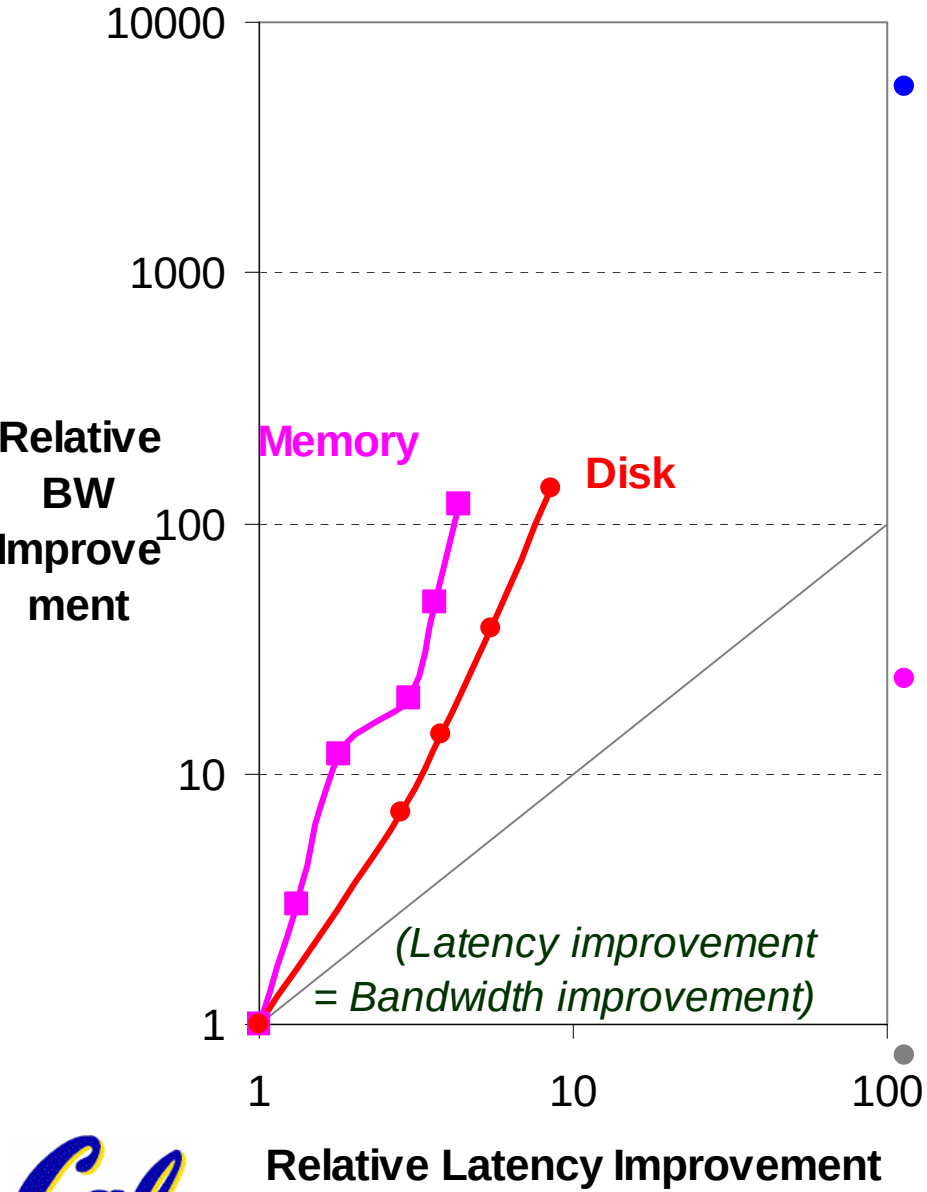
(latency = simple operation w/o contention
BW = best-case)

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

- 1980 DRAM (asynchronous)
- 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)

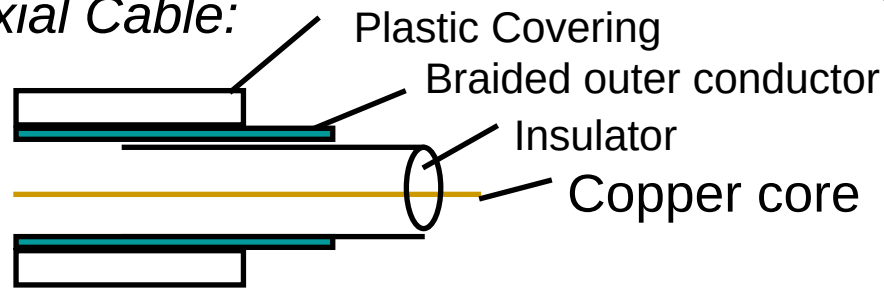
(latency = simple operation w/o contention
BW = best-case)

LANs: 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

Coaxial Cable:

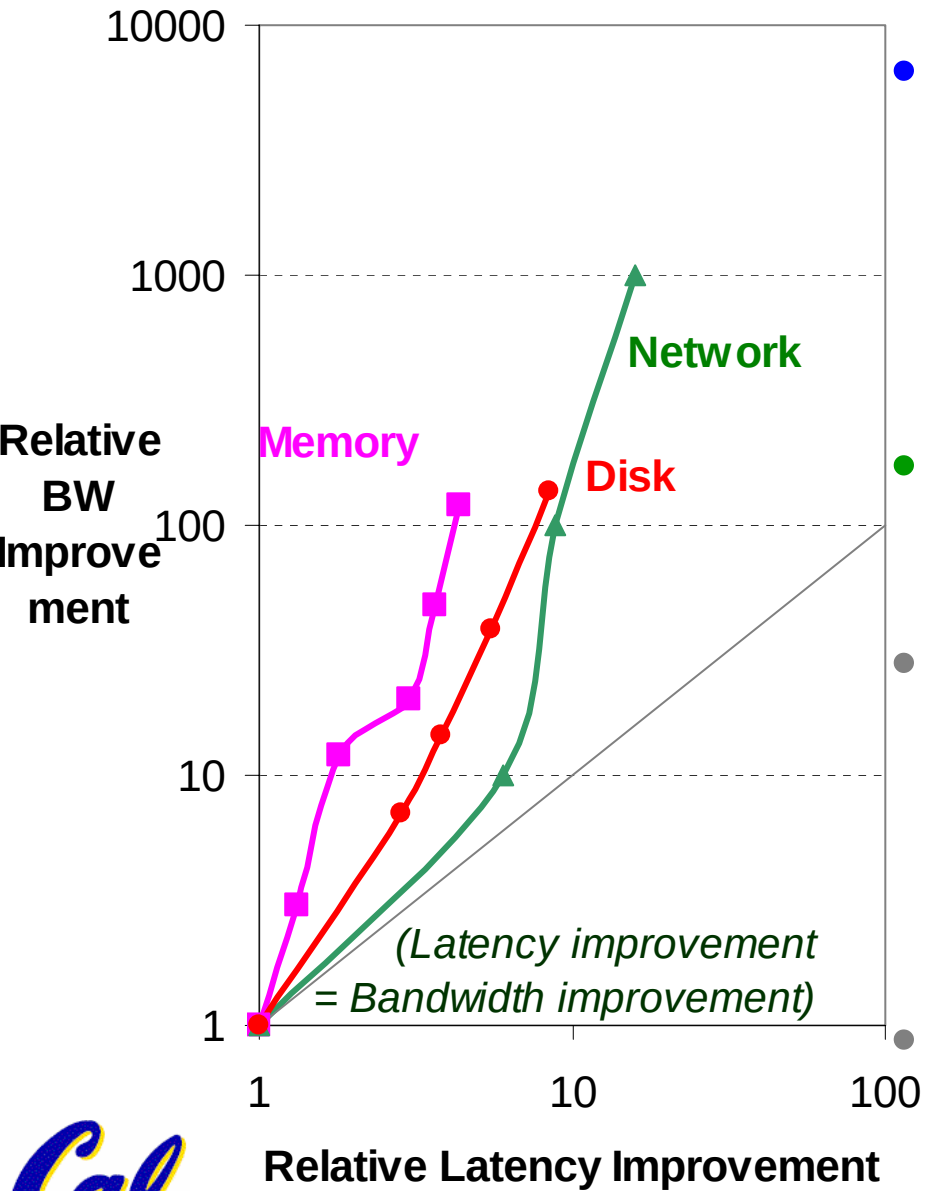


"Cat 5" is 4 twisted pairs in bundle

Twisted Pair:

Copper, 1mm thick,
twisted to avoid antenna effect

Latency Lags Bandwidth (last ~20 years)

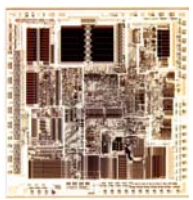


- Performance Milestones
- Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s (16x,1000x)
- Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
- Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

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

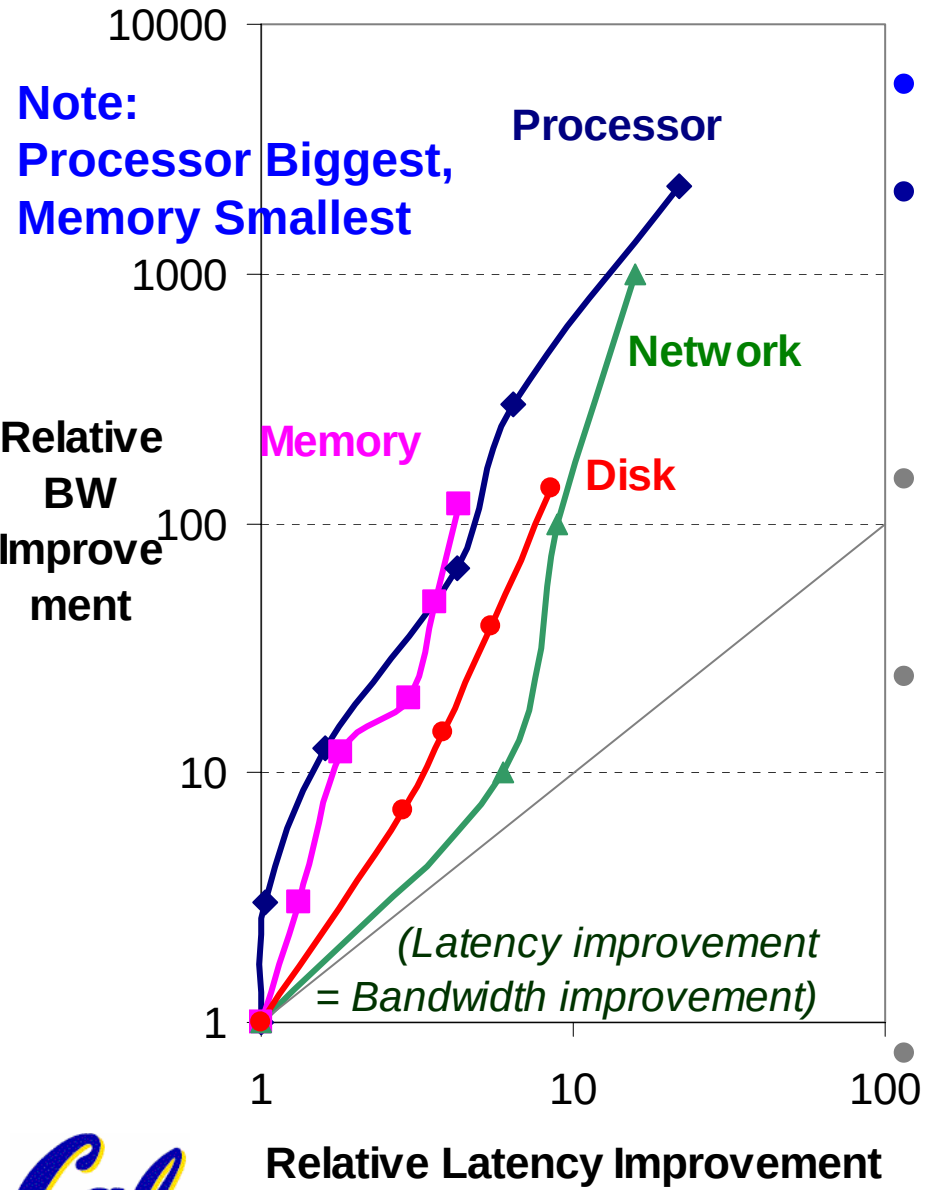
CPUs: Archaic(Nostalgic) v. Modern(Newfangled)

- 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 (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)

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

Annual Improvement per Technology

	CPU	DRAM	LAN	Disk
Annual Bandwidth Improvement (all milestones)	1.50	1.27	1.39	1.28
Annual Latency Improvement (all milestones)	1.17	1.07	1.12	1.11

- Again, CPU fastest change, DRAM slowest
- But what about recent BW, Latency change?

Annual Bandwidth Improvement (last 3 milestones)	1.55	1.30	1.78	1.29
Annual Latency Improvement (last 3 milestones)	1.22	1.06	1.13	1.09

- How summarize BW vs. Latency change?

Towards a Rule of Thumb

- **How long for Bandwidth to Double?**

Time for Bandwidth to Double (Years, all milestones)	1.7	2.9	2.1	2.8

- **How much does Latency Improve in that time?**

Latency Improvement in Time for Bandwidth to Double (all milestones)	1.3	1.2	1.3	1.3

- **But what about recently?**

Time for Bandwidth to Double (Years, last 3 milestones)	1.6	2.7	1.2	2.7
Latency Improvement in Time for Bandwidth to Double (last 3 milestones)	1.4	1.2	1.2	1.3

- **Despite faster LAN, all 1.2X to 1.4X**

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

What if Latency Didn't Lag BW?

- Life would have been simpler for designers if Latency had kept up with Bandwidth
  - E.g., 0.1 nanosecond latency processor,
    2 nanosecond latency memory,
    3 microsecond latency LANs,
    0.3 millisecond latency disks
- Why does it Lag?

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 $\Rightarrow$ 2X)
  - DRAM Die Size: 47 vs. 217 mm^2 (ratio sqrt $\Rightarrow$ 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

3 Ways to Cope with Latency Lags Bandwidth

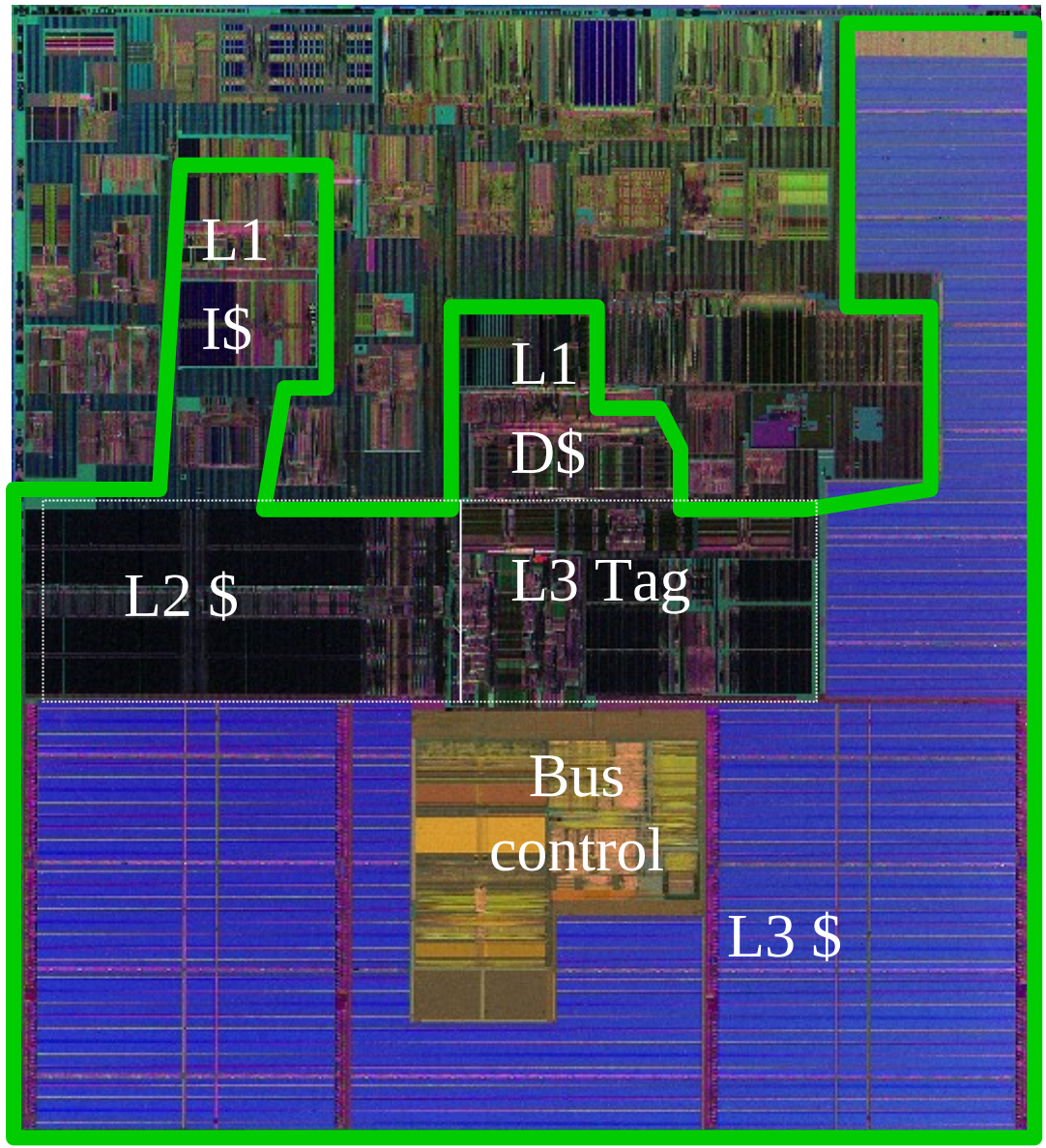
"If a problem has no solution, it may not be a problem, but a fact--not to be solved, but to be coped with over time"

— Shimon Peres ("Peres's Law")

1. **Caching (Leveraging Capacity)**
   - Processor caches, file cache, disk cache
2. **Replication (Leveraging Capacity)**
   - Read from nearest head in RAID, from nearest site in content distribution
3. **Prediction (Leveraging Bandwidth)**
   - Branches + Prefetching: disk, caches

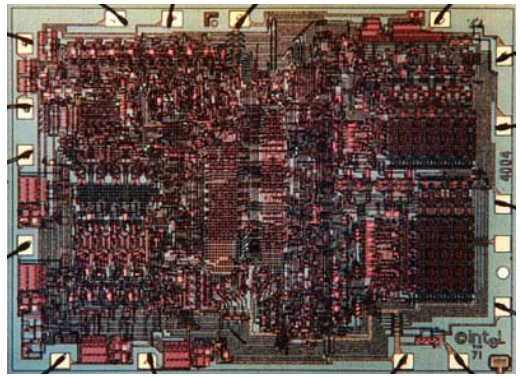
BW vs. Latency: MPU "State of the art?"

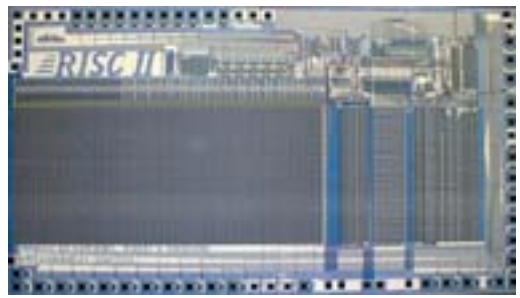
- Latency via caches
- Intel Itanium II has 4 caches on-chip!
  - 2 Level 1 caches: 16 KB I and 16 KB D
  - Level 2 cache: 256 KB
  - Level 3 cache: 3072 KB
- 211M transistors
  ~85% for caches
- Die size 421 mm^2
- 130 Watts @ 1GHz
- 1% die to change data, 99% to move, store data?



HW BW Example: Micro Massively Parallel Processor (μMMP)

- Intel 4004 (1971): 4-bit processor, 2312 transistors, 0.4 MHz, 10 micron PMOS, 11 mm^2 chip
- RISC II (1983): 32-bit, 5 stage pipeline, 40,760 transistors, 3 MHz, 3 micron NMOS, 60 mm^2 chip
  - 4004 shrinks to ~ 1 mm^2 at 3 micron
- 250 mm^2 chip, 0.090 micron CMOS = 2312 RISC IIs + Icache + Dcache
  - RISC II shrinks to ~ 0.05 mm^2 at 0.09 mi.
  - Caches via DRAM or 1 transistor SRAM (www.t-ram.com)
  - Proximity Communication via capacitive coupling at > 1 TB/s (Ivan Sutherland@Sun)





Processor = new transistor?

Cost of Ownership, Dependability, Security v. Cost/Perf. => μMPP

SW Design Example: Planning for BW gains

Goal: Dependable storage system keeps multiple replicas of data at remote sites

- Caching (obviously) to reducing latency
- Replication: multiple requests to multiple copies and just use the quickest reply
- Prefetching to reduce latency
- Large block sizes for disk and memory
- Protocol: few very large messages
  - vs. chatty protocol with lots small messages
- Log-structured file sys. at each remote site

Too Optimistic so Far (its even worse)?

- Optimistic: Cache, Replication, Prefetch get more popular to cope with imbalance
- Pessimistic: These 3 already fully deployed, so must find next set of tricks to cope; hard!
- Its even worse: bandwidth gains multiplied by replicated components ⇒ parallelism
  - simultaneous communication in switched LAN
  - multiple disks in a disk array
  - multiple memory modules in a large memory
  - multiple processors in a cluster or SMP

Conclusion: Latency Lags Bandwidth

- For disk, LAN, memory, and MPU, in the time that bandwidth doubles, latency improves by no more than 1.2X to 1.4X
  - BW improves by square of latency improvement
- Innovations may yield one-time latency reduction, but unrelenting BW improvement
- If everything improves at the same rate, then nothing really changes
  - When rates vary, require real innovation
- HW and SW developers should innovate assuming Latency Lags Bandwidth