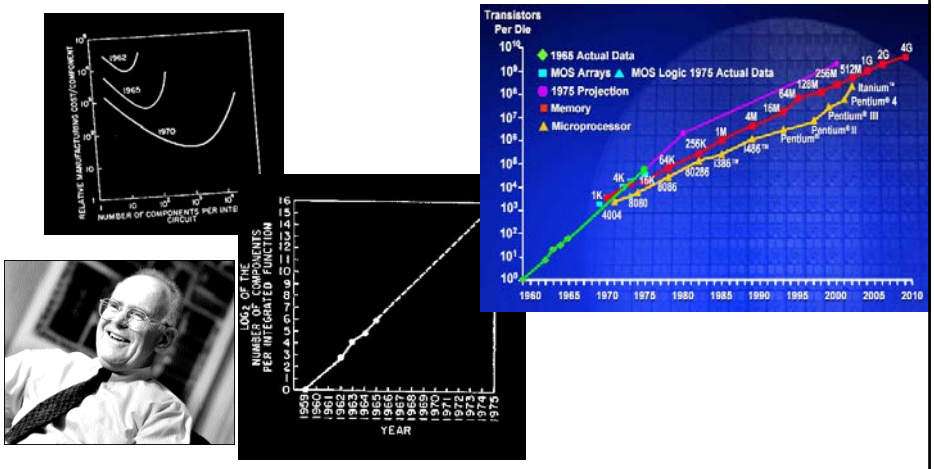
Software Engineering for Future Computer Architectures

Novica Nosovic
ETF Sarajevo

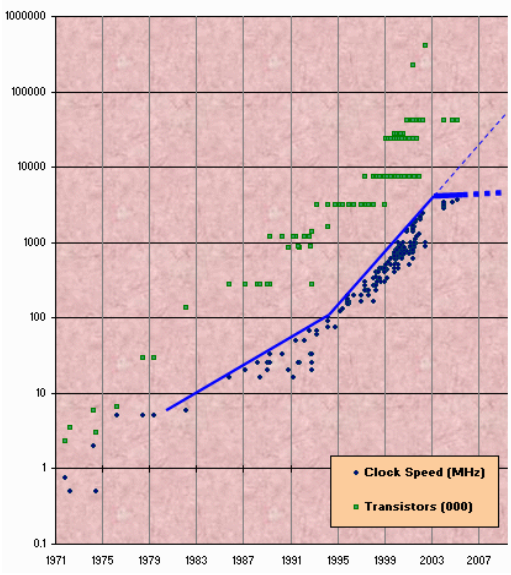
7th Workshop
"Software Engineering Education and Reverse Engineering"
Risan, Montenegro, 8 – 15 September 2007

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)**

Why there is no 20GHz processor today!



Walls all around!

- **power wall,**
- **memory wall,**
- **transistor wall...**

Tracking Technology Performance Trends

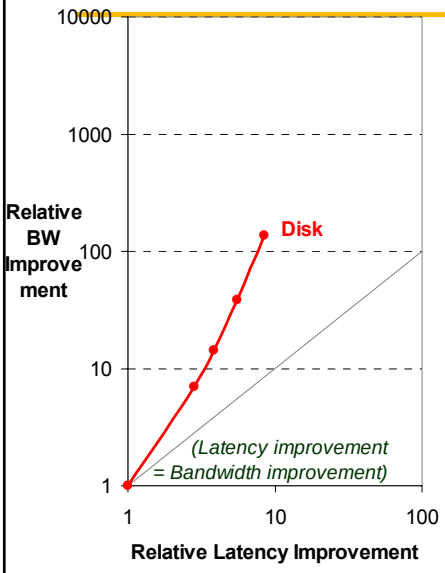
- 4 technologies – key components:
  - Disks,
  - Memory,
  - Network,
  - Processors
- Compare ~1980 Archaic vs. ~2000 Modern
  - Performance Milestones in each technology
- Compare for Bandwidth vs. Latency improvements in performance over time
- Bandwidth: number of events per unit time
  - E.g., 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: Archaic v. Modern

- CDC Wren I, 1983
- 3600 RPM
- 0.03 GBytes
- 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



- **Performance Milestones**

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

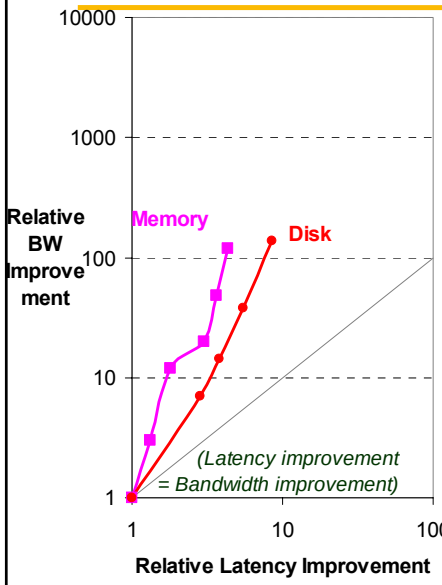
(latency = simple operation w/o contention
BW = best-case)

Memory: Archaic v. Modern

- 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 Synchronous 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



- **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)

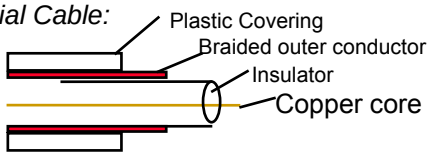
(latency = simple operation w/o contention
BW = best-case)

LANs: Archaic v. Modern

- 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:

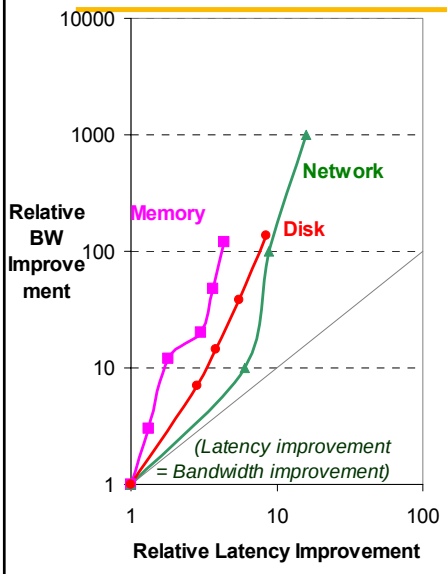


"Cat 5" is 4 twisted pairs in bundle

Twisted Pair:

Copper, 1mm thick,
twisted to avoid antenna effect

Latency Lags Bandwidth



- **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)

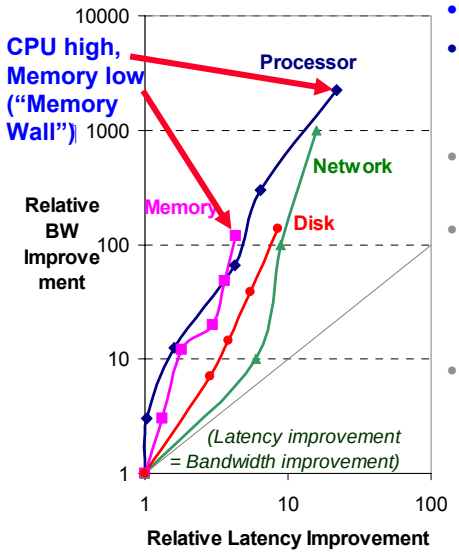
(latency = simple operation w/o contention
BW = best-case)

CPUs: Archaic v. Modern

- 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



- **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)

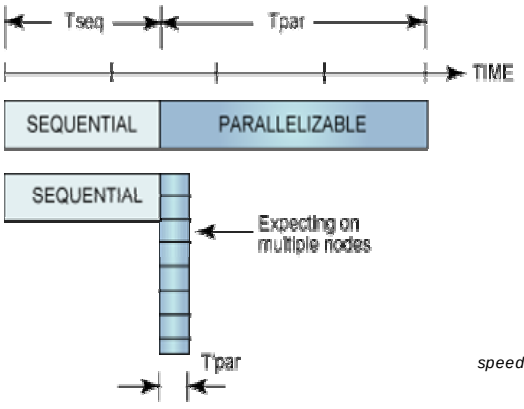
Rule of Thumb for Latency Lagging BW

- **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

Summary of Technology Trends

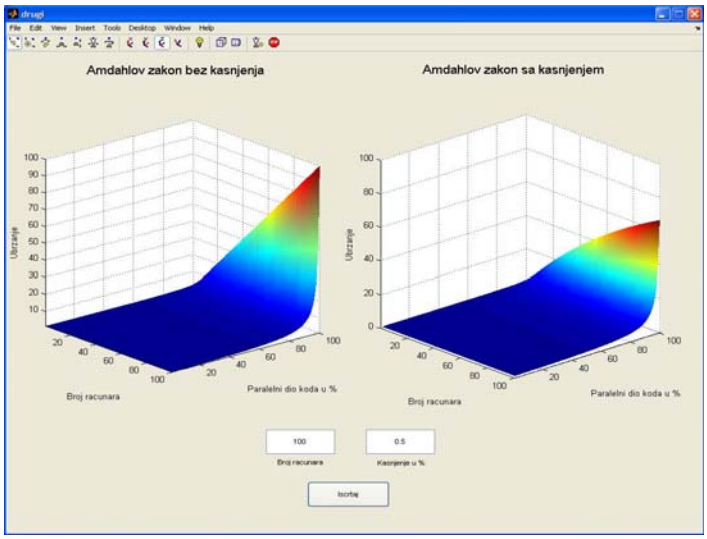
- For disk, LAN, memory, and microprocessor, bandwidth improves by square of latency improvement
  - In the time that bandwidth doubles, latency improves by no more than 1.2X to 1.4X
- Lag probably even larger in real systems, as bandwidth gains multiplied by replicated components
  - Multiple processors 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

Amdahl's Law



$$speedup = \frac{1}{S + \frac{P}{N}}$$

Amdahl's Law + latency



20 years of "free lunch"

- no need for more processors
- just wait a year and the processor gets faster

Multicore processors today

- **Intel and AMD sell multicore only!**
- **first multicore - two processors on a chip (slap together),** not very tightly integrated
- four-core chips where it's really a redesign

Manycore to come

- **Not only cores that double like chromosomes**
- communication network on chip
- very tightly coupled
- memory architecture is changing - bandwidth has increased dramatically
- GPUs, Cell... different memory model and cache coherency

Software is not ready!!!

- **traditional model - threads! works well with shared memory**
- **distributed memory ... threads do not do...**
- **but VM! like JVM!?**
- **VM manages processors, distributed memory... for photo editing, multimedia on desktop, speech recognition (lacks floating point footage!!)**

The bigest wall!

- How can SE keep pace with these evolving HW that are rendering the existing application base obsolete?
- Eentirely different way to program is needed
- It is not something developers are used to
- There is a real void in the tools world on how to program

For Java lovers!

- So we're starting to move to processors that have distributed memory...
- ...where that thread shared memory model doesn't work

Which way to go?