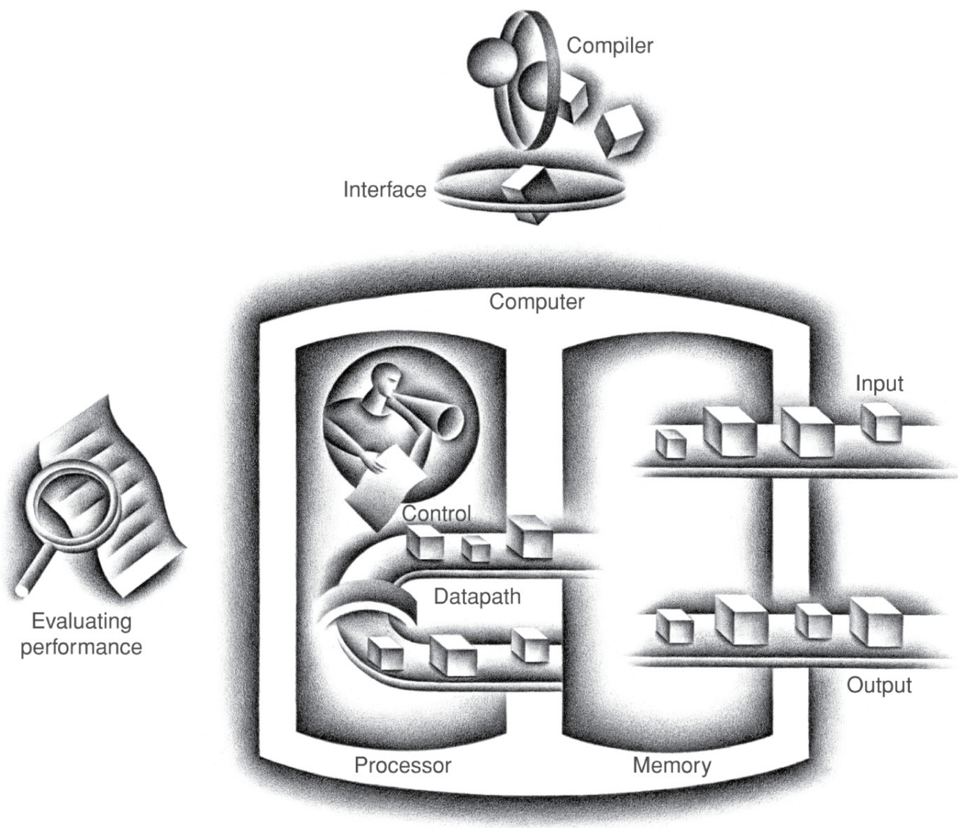
Model of a Computer



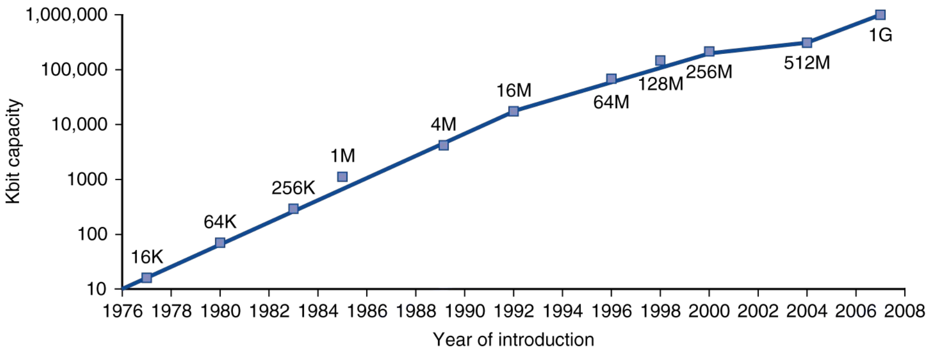
- von Neumann model
- CPU
  - control & data path
- I/O
  - user, storage, network
- memory

program & data stored in memory

Trends and Challenges

Technology Trends

- electronics technology continues to evolve
  - increased capacity and performance
  - reduced cost



DRAM capacity

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated Circuit	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

Performance: Latency vs. Throughput

- Tim Horton's
  - time to coffee vs. customers/hour
  - low latency => high throughput
  - but not vice versa
  - faster coffee makers vs. more (and more space)
- latency (response time)
  - completion time of specific task
- throughput
  - total work done over time period

Performance

- reduce latency?
  - faster processor
  - better algorithm (software)
  - more processors (needs parallelization)
  - generally increases throughput
- increase throughput?
  - more processors
  - rearrange system components (scheduling): often increases latency

Efficiency Matters

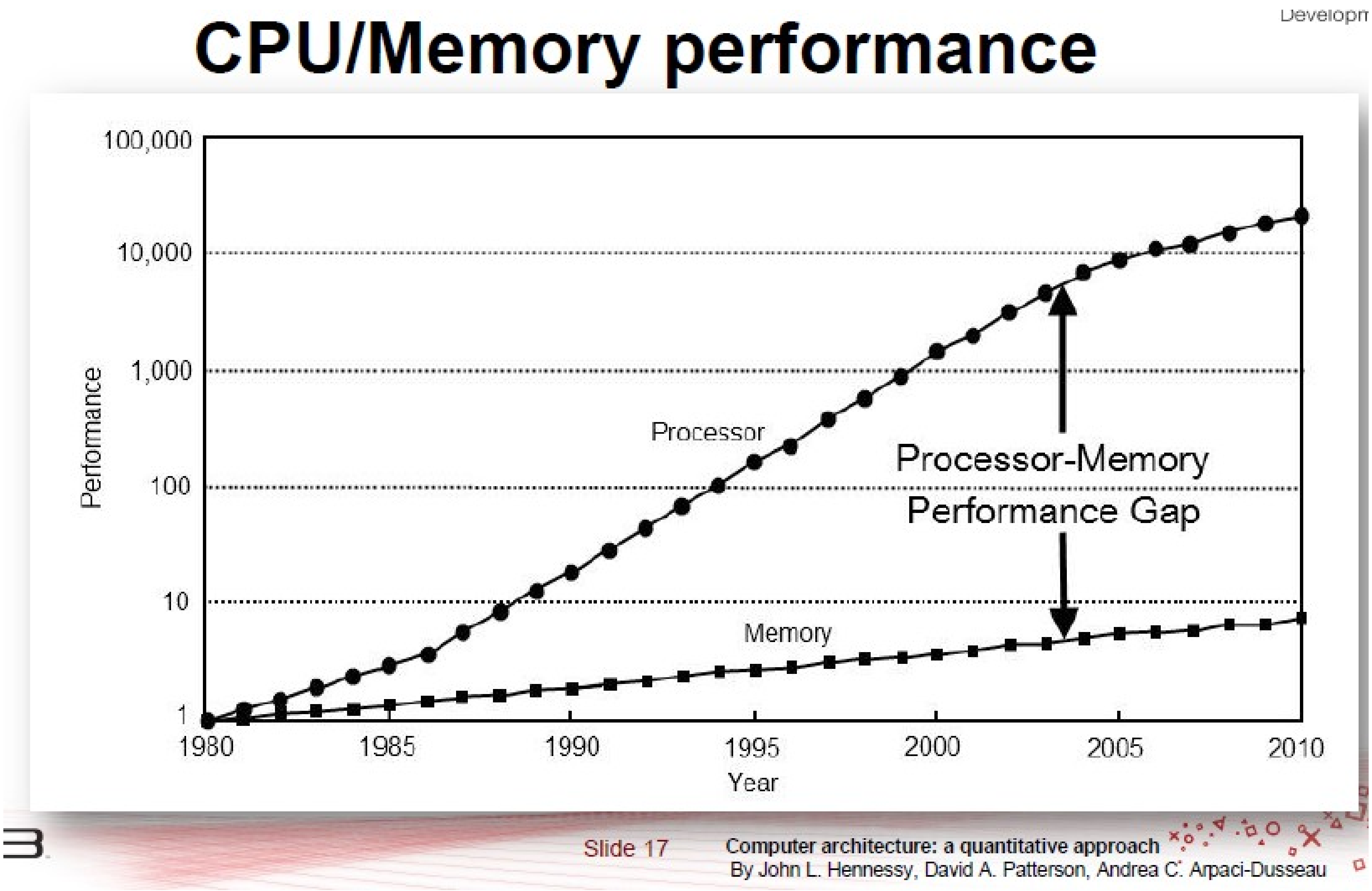
- network-centric computing, Internet

  -> large data centers
- hardware cheap, but
  - power consumption -> heat
  - heat -> cooling -> more power consumption
  - money and environment costs
- often:
  software performance (throughput) ~ efficiency

Moore's Law

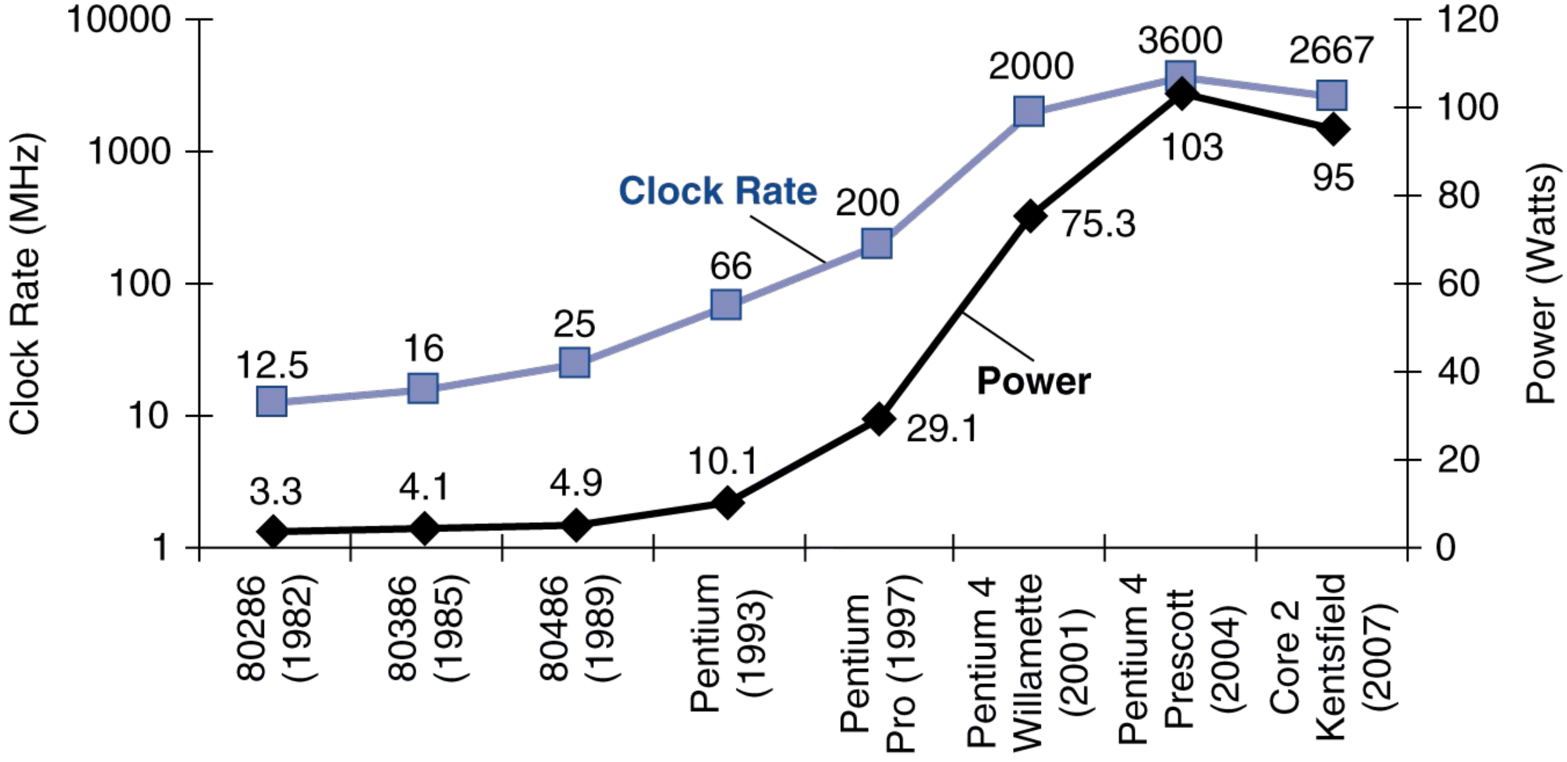
- transistor density doubles every two years
  - every year 1959-1975
- in the past
  - transistor density translated into processing power
  - almost double speed every 2 years...
  - reduce latency, increase throughput
- recently: memory wall
- more recently: power wall

Memory Wall



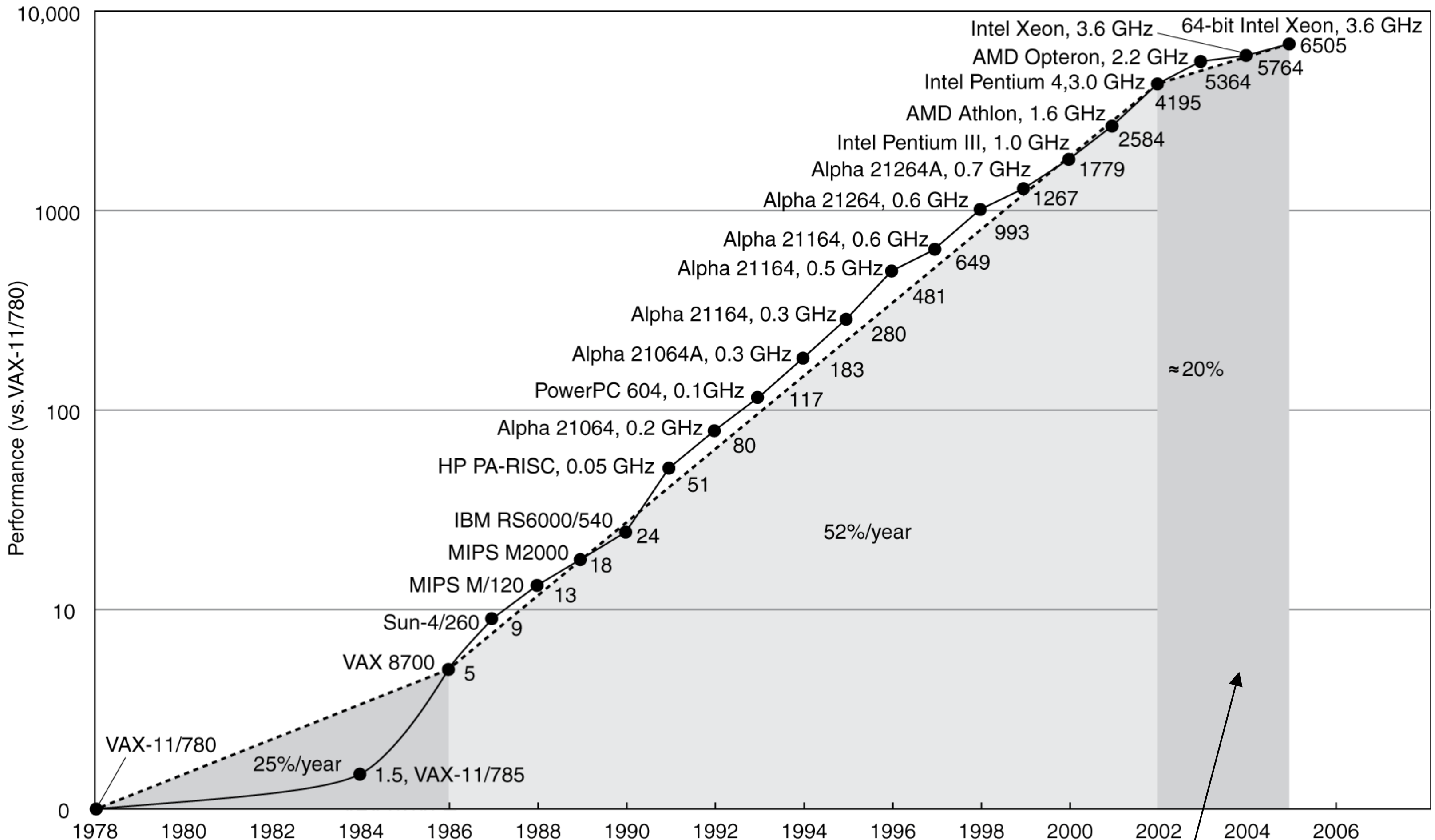
Developm
CPU/Memory performance
100,000
10,000
1,000
100
10
1
Performance
Processor
Memory
Processor-Memory
Performance Gap
1980
1985
1990
1995
2000
2005
2010
Year
Slide 17
Computer architecture: a quantitative approach
By John L. Hennessy, David A. Patterson, Andrea C. Arpaci-Dusseau

Power Wall



- power = capacitive load x voltage^2 x frequency
  - cannot reduce voltage further (path length)
  - cannot remove more heat

Uniprocessor Performance



Constrained by power, instruction-level parallelism, memory latency

Multiprocessors

- multicore microprocessors
  - more than one processor per chip
- requires explicitly parallel programming
  - compare with instruction level parallelism (hidden)
- hard to do
  - programming for performance
  - load balancing
  - optimizing communication and synchronization

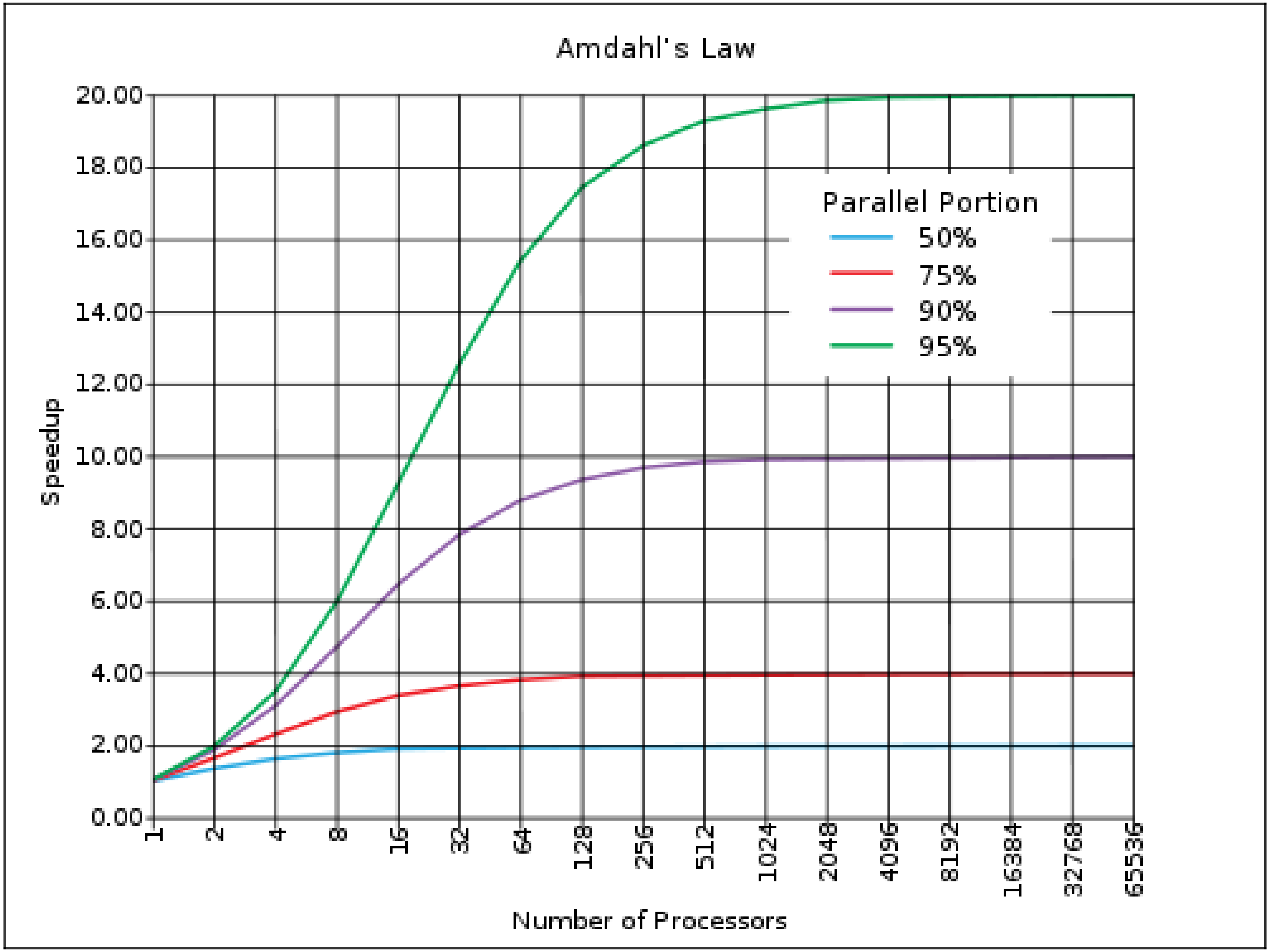
Amdahl's Law

- improve some part of a computer program
  - or it's execution speed (e.g., through parallelization)

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

- limits overall performance improvement

Amdahl's Law



Source: Wikimedia Commons

Trade-Offs

- almost everything in CS is a trade-off
  - very few absolute truths
- “fast, good, or cheap – pick two”