CS 61C: Great Ideas in Computer Architecture (Machine Structures)

Instructors:
Randy H. Katz
David A. Patterson

http://inst.eecs.Berkeley.edu/~cs61c/fa10

Agenda

- Review
- Defining Performance, Latency vs. Bandwidth
- Administrivia
- Technology Break
- Measuring Performance

Review

- Translate from text that easy for programmers to understand into code that machine executes efficiently: Compilers, Assemblers
- Linkers allow separate translation of modules
- Interpreters for debugging, but slow execution
- Hybrid (Java): Compiler + Interpreter to try to get best of both

Defining Performance

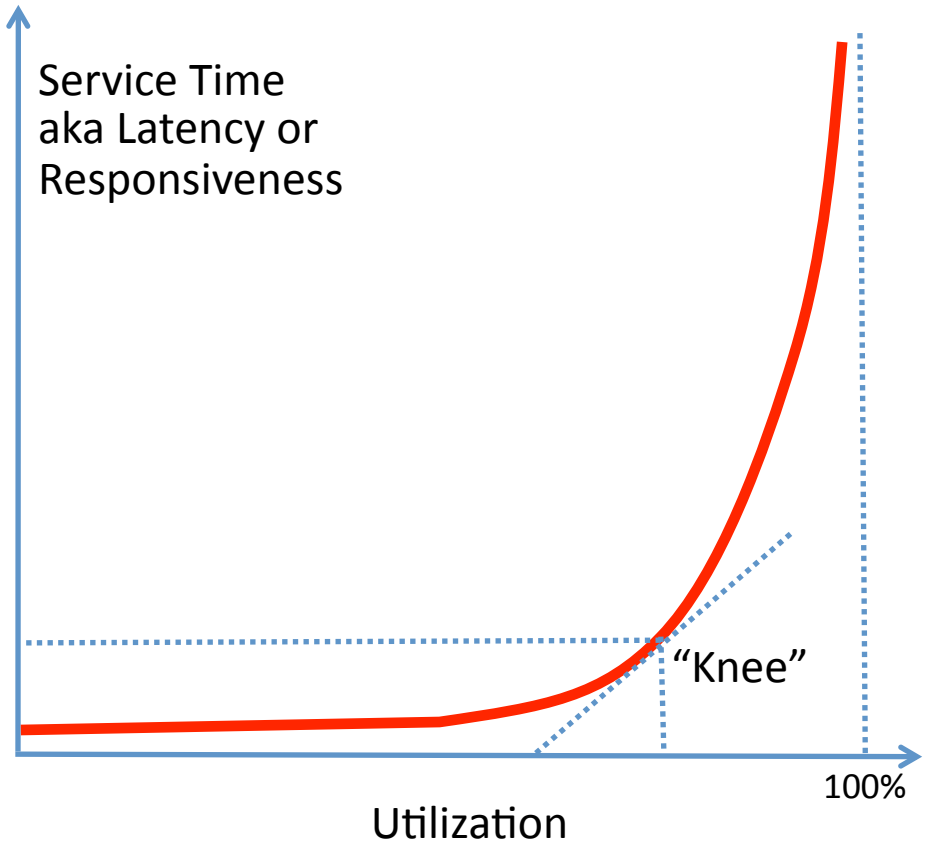
- What does it mean to say X is faster than Y?
- Ferrari vs. School Bus?
- 2009 Ferrari 599 GTB
  - 2 passengers, 11.1 secs in quarter mile
- 2009 Type D school bus
  - 54 passengers, quarter mile time?

http://www.youtube.com/watch?v=KwyCoQuhUNA

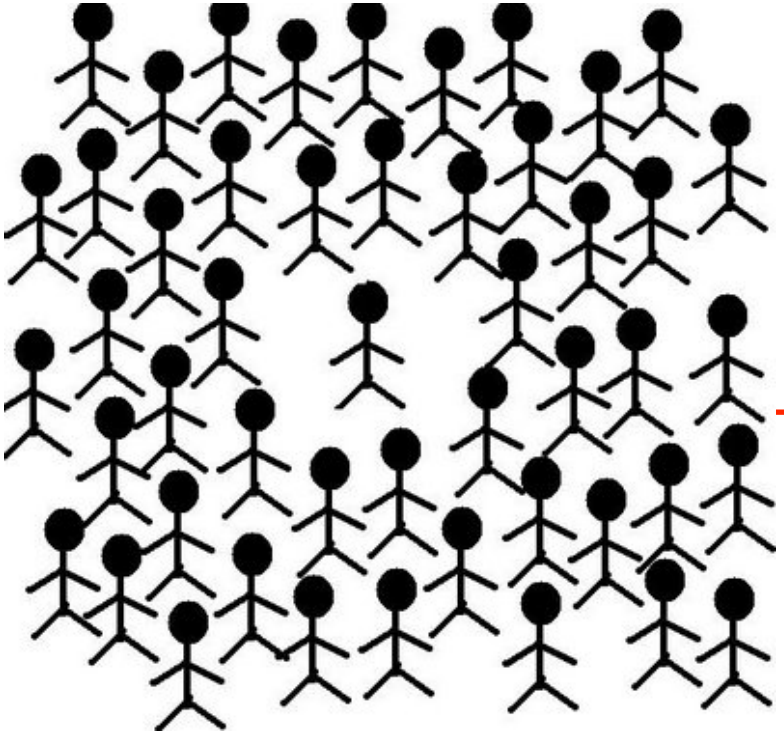
- *Response Time* or *Latency*: time between start and completion of a task (time to move vehicle ¼ mile)
- *Throughput* or *Bandwidth*: total amount of work in a given time (passenger-miles in 1 hour)

Running Systems to 100% Utilization

- Implication of the graph at the right?
- Can you explain why this happens?



The Iron Law of Queues (aka Little's Law)



$L = \lambda W$

Average number of customers in system (L) = average interarrival rate (λ) x average service time (W)

Cloud Performance: Why Application Latency Matters

Server Delay (ms)	Increased time to next click (ms)	Queries/ user	Any clicks/ user	User satisfac- tion	Revenue/ User
50	--	--	--	--	--
200	500	--	-0.3%	-0.4%	--
500	1200	--	-1.0%	-0.9%	-1.2%
1000	1900	-0.7%	-1.9%	-1.6%	-2.8%
2000	3100	-1.8%	-4.4%	-3.8%	-4.3%

Figure 6.10 Negative impact of delays at Bing search server on user behavior [Brutlag and Schurman 2009].

- Key figure of merit: application responsiveness
  - Longer the delay, the fewer the user clicks, the less the user happiness, and the lower the revenue per user

Defining Relative Performance

- $\text{Performance}_X = 1/\text{Program Execution Time}_X$
- $\text{Performance}_X > \text{Performance}_Y \Rightarrow$
  $1/\text{Execution Time}_X > 1/\text{Execution Time}_y \Rightarrow$
  $\text{Execution Time}_Y > \text{Execution Time}_X$
- Computer X is N times faster than Computer Y
  $\text{Performance}_X / \text{Performance}_Y = N$ or
  $\text{Execution Time}_Y / \text{Execution Time}_X = N$
- Ferrari is 12/11.1 or 1.08 times faster than bus

Measuring Performance

- Computers use a clock to determine when events takes place within hardware
- *Clock cycles* – discrete time intervals
  - a.k.a. clocks, cycles, clock periods, clock ticks
- *Clock rate* or *clock frequency* – clock cycles per second (inverse of clock cycle time)
- 3 GigaHertz clock rate
  => clock cycle time = $1/3 \times 10^9$ seconds
  clock cycle time = 333 picoseconds (ps)

CPU Performance Factors

- To distinguish between processor time and I/O, *CPU time* is time spent in processor
- CPU time/program
  = Clock cycles/program x clock cycle time
- Or CPU time/program
  = Clock cycles/program ÷ clock rate

CPU Performance Factors

- But a program executes instructions
- CPU time/program
  = Clock cycles/program x clock cycle time
  = Instructions/program
  x Average Clock cycles/Instruction
  x clock cycle time
- 1st term called *Instruction Count*
- 3rd term is 1 / Clock rate
- 2nd term abbreviated *CPI* for average *Clock cycles Per Instruction*
- Why CPI = 1? Why CP1 > 1? Can CPI be < 1?

Restating Performance Equation

- $$\text{Time} = \frac{\text{Seconds}}{\text{Program}}$$

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock Cycle}}$$

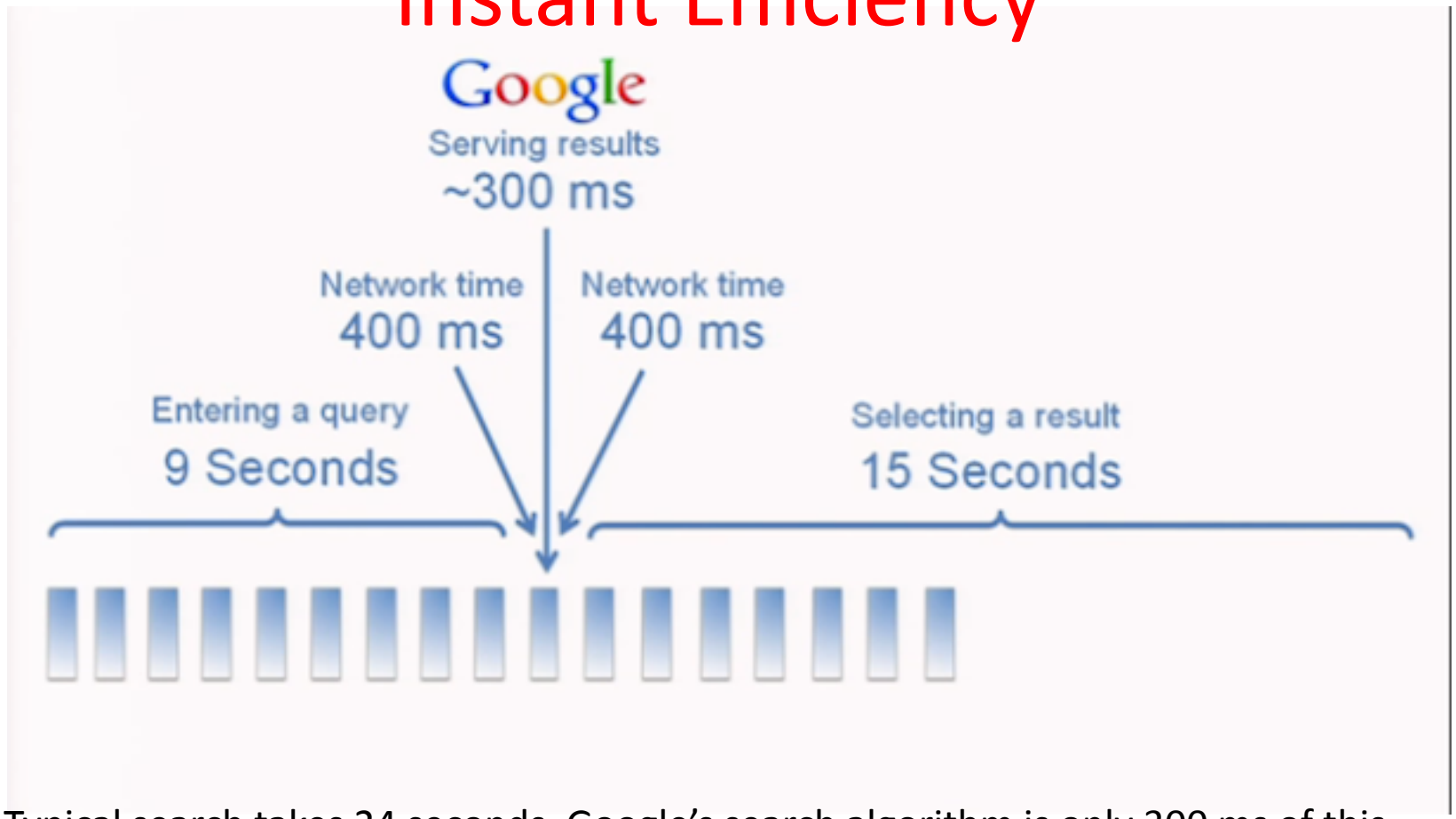
What affects each component? Instruction Count, CPI, Clock Rate

Hardware or software component?	Affects What?
Algorithm	
Programming Language	
Compiler	
Instruction Set Architecture	

Agenda

- Review
- Defining Performance, Latency vs. Bandwidth
- Administrivia
- Technology Break
- Measuring Performance

Breaking News: Google Instant Search "Instant Efficiency"



Typical search takes 24 seconds, Google's search algorithm is only 300 ms of this
"It's not search 'as you type', but 'search before you type'!"
"We can predict what you are likely to type and give you those results in real time"

Peer Instruction

- Computer A clock cycle time 250 ps, $CPI_A = 2$
- Computer B clock cycle time 500 ps, $CPI_B = 1.2$
- Assume A and B have same instruction set
- Which statement is true?

A. Computer A is ~1.2 times faster than B
B. Computer A is ~4.0 times faster than B
C. Computer B is ~1.7 times faster than A
D. Computer B is ~3.4 times faster than A
E. None of the above

Workload and Benchmark

- *Workload:* A set of programs run on a computer
  - The actual collection of applications run or made from real programs to approximate such a mix
  - Specifies both programs and relative frequencies
- *Benchmark* A program selected for use in comparing computer performance
  - Benchmarks form a workload
  - Usually standardized so that many use them

SPEC (System Performance Evaluation Cooperative)

- Computer Vendor cooperative for benchmarks, started in 1989
- SPECCPU2006
  - 12 Integer Programs
  - 17 Floating-Point Programs
- Often turn into number where bigger is faster
- *SPECratio*: reference execution time on old reference computer divide by execution time on new computer

SPECINT2006 on AMD Barcelona

Description	Instruc-tion Count (B)	CPI	Clock cycle time (ps)	Execu-tion Time (s)	Refer-ence Time (s)	SPEC-ratio
Interpreted string processing	2,118	0.75	400	637	9,770	15.3
Block-sorting compression	2,389	0.85	400	817	9,650	11.8
GNU C compiler	1,050	1.72	400	724	8,050	11.1
Combinatorial optimization	336	10.0	400	1,345	9,120	6.8
Go game	1,658	1.09	400	721	10,490	14.6
Search gene sequence	2,783	0.80	400	890	9,330	10.5
Chess game	2,176	0.96	400	837	12,100	14.5
Quantum computer simulation	1,623	1.61	400	1,047	20,720	19.8
Video compression	3,102	0.80	400	993	22,130	22.3
Discrete event simulation library	587	2.94	400	690	6,250	9.1
Games/path finding	1,082	1.79	400	773	7,020	9.1
XML parsing	1,058	2.70	400	1,143	6,900	6.0

Summary

- Time (seconds/program) is measure of performance

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock Cycle}}$$

- Algorithms, Programming Languages, Compilers, Instruction Set affect Instruction Count and CPI
- Instruction Set affects Clock Period/ Clock Rate
- Benchmarks stand in for real workloads to as standardized measure of relative performance