CS61C - Machine Structures

Lecture 22 - Introduction to Performance

November 17, 2000

David Patterson

http://www-inst.eecs.berkeley.edu/~cs61c/

Review (1/2)

° Optimal Pipeline

- Each stage is executing part of an instruction each clock cycle.
- One instruction finishes during each clock cycle.
- On average, execute far more quickly.

° What makes this work?

- Similarities between instructions allow us to use same stages for all instructions (generally).
- Each stage takes about the same amount of time as all others: little wasted time.

Review (2/2)

° Pipelining a Big Idea: widely used concept

° What makes it less than perfect?

- Structural hazards: suppose we had only one cache?
  ⇒ Need more HW resources
- Control hazards: need to worry about branch instructions?
  ⇒ Delayed branch
- Data hazards: an instruction depends on a previous instruction?

Outline

° Performance Calculation

° Benchmarks

° Virtual Memory Review

Performance

° Purchasing Perspective: given a collection of machines, which has the

- best performance ?
- least cost ?
- best performance / cost ?

° Computer Designer Perspective: faced with design options, which has the

- best performance improvement ?
- least cost ?
- best performance / cost ?

° Both require: basis for comparison and metric for evaluation

Two Notions of "Performance"

Plane	DC to Paris	Top Speed	Passen-gers	Throughput (pmph)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200

- Which has higher performance?
- Time to deliver 1 passenger?
- Time to deliver 400 passengers?
- In a computer, time for 1 job called Response Time or Execution Time
- In a computer, jobs per day called Throughput or Bandwidth

Definitions

° Performance is in units of things per sec
- bigger is better

° If we are primarily concerned with response time
- performance(x) = $\frac{1}{\text{execution\_time(x)}}$

" X is n times faster than Y" means

$$n = \frac{\text{Performance(X)}}{\text{Performance(Y)}}$$

Example of Response Time v. Throughput

- Time of Concorde vs. Boeing 747?
  - Concord is 6.5 hours / 3 hours = 2.2 times faster
- Throughput of Boeing vs. Concorde?
  - Boeing 747: 286,700 pmph / 178,200 pmph = 1.6 times faster
- Boeing is 1.6 times ("60%") faster in terms of throughput
- Concord is 2.2 times ("120%") faster in terms of flying time (response time)

We will focus primarily on execution time for a single job

Confusing Wording on Performance

° Will (try to) stick to "n times faster"; its less confusing than "m % faster"

° As faster means both increased performance and decreased execution time, to reduce confusion will use "improve performance" or "improve execution time"

What is Time?

° Straightforward definition of time:
- Total time to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, ...
- "real time", "response time" or "elapsed time"

° Alternative: just time processor (CPU) is working only on your program (since multiple processes running at same time)
- "CPU execution time" or "CPU time"
- Often divided into system CPU time (in OS) and user CPU time (in user program)

How to Measure Time?

° User Time ⇒ seconds

° CPU Time: Computers constructed using a clock that runs at a constant rate and determines when events take place in the hardware
- These discrete time intervals called clock cycles (or informally clocks or cycles)
- Length of clock period: clock cycle time (e.g., 2 nanoseconds or 2 ns) and clock rate (e.g., 500 megahertz, or 500 MHz), which is the inverse of the clock period; use these!

Measuring Time using Clock Cycles (1/2)

° CPU execution time for program

= Clock Cycles for a program x Clock Cycle Time

° or

$$= \frac{\text{Clock Cycles for a program}}{\text{Clock Rate}}$$

Measuring Time using Clock Cycles (2/2)

° One way to define clock cycles:

Clock Cycles for program

= Instructions for a program
(called "Instruction Count")

x Average Clock cycles Per Instruction
(abbreviated "CPI")

° CPI one way to compare two machines with same instruction set, since Instruction Count would be the same

Performance Calculation (1/2)

° CPU execution time for program
= Clock Cycles for program
x Clock Cycle Time

° Substituting for clock cycles:

CPU execution time for program
= (Instruction Count x CPI)
x Clock Cycle Time

= Instruction Count x CPI x Clock Cycle Time

Performance Calculation (2/2)

$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

$$\text{CPU time} = \frac{\cancel{\text{Instructions}}}{\text{Program}} \times \frac{\text{Cycles}}{\cancel{\text{Instruction}}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

$$\text{CPU time} = \frac{\cancel{\text{Instructions}}}{\text{Program}} \times \frac{\cancel{\text{Cycles}}}{\cancel{\text{Instruction}}} \times \frac{\text{Seconds}}{\cancel{\text{Cycle}}}$$

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}}$$

- Product of all 3 terms: if missing a term, can't predict time, the real measure of performance

Administrivia: Rest of 61C

• Rest of 61C slower pace

- 1 project, 1 lab, no more homeworks

F 11/17 Performance; Cache Sim Project
W 11/24 X86, PC buzzwords and 61C; RAID Lab

W 11/29 Review: Pipelines; Feedback "lab"
F 12/1 Review: Caches/TLB/VM; Section 7.5

M 12/4 Deadline to correct your grade record

W 12/6 Review: Interrupts (A.7); Feedback lab
F 12/8 61C Summary / Your Cal heritage / HKN Course Evaluation

Sun 12/10 Final Review, 2PM (155 Dwinelle)
Tues 12/12 Final (5PM 1 Pimintel)

How Calculate the 3 Components?

° Clock Cycle Time: in specification of computer (Clock Rate in advertisements)

° Instruction Count:

- Count instructions in loop of small program
- Use simulator to count instructions
- Hardware counter in spec. register (Pentium II)

° CPI:

- Calculate: $\frac{\text{Execution Time / Clock cycle time}}{\text{Instruction Count}}$
- Hardware counter in special register (PII)

Calculating CPI Another Way

° First calculate CPI for each individual instruction (add, sub, and, etc.)

° Next calculate frequency of each individual instruction

° Finally multiply these two for each instruction and add them up to get final CPI

Example (RISC processor)

Op	$Freq_i$	CPI_i	Prod	(% Time)
ALU	50%	1	.5	(23%)
Load	20%	5	1.0	(45%)
Store	10%	3	.3	(14%)
Branch	20%	2	.4	(18%)
			2.2	

Instruction Mix (Where time spent)

- What if Branch instructions twice as fast?

Example: What about Caches?

- Can Calculate Memory portion of CPI separately
- Miss rates: say L1 cache = 5%, L2 cache = 10%
- Miss penalties: L1 = 5 clock cycles, L2 = 50 clocks
- Assume miss rates, miss penalties same for instruction accesses, loads, and stores
- CPI_{memory}
  = Instruction Frequency * L1 Miss rate *
(L1 miss penalty + L2 miss rate * L2 miss penalty)
+ Data Access Frequency * L1 Miss rate *
(L1 miss penalty + L2 miss rate * L2 miss penalty)
= 100%*5%*(5+10%*50)+(20%+10%)*5%*(5+10%*50)
= 5%*(10)+(30%)*5%*(10) = 0.5 + 0.15 = 0.65

Overall CPI = 2.2 + 0.65 = 2.85

What Programs Measure for Comparison?

° Ideally run typical programs with typical input before purchase, or before even build machine

- Called a "workload"; For example:
- Engineer uses compiler, spreadsheet
- Author uses word processor, drawing program, compression software

° In some situations its hard to do

- Don't have access to machine to "benchmark" before purchase
- Don't know workload in future

Benchmarks

° Obviously, apparent speed of processor depends on code used to test it

° Need industry standards so that different processors can be fairly compared

° Companies exist that create these benchmarks: "typical" code used to evaluate systems

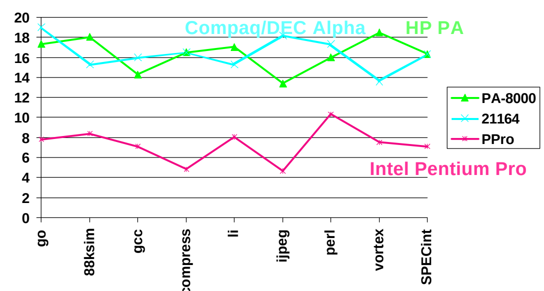
° Need to be changed every 2 or 3 years since designers could target these standard benchmarks

Example Standardized Workload Benchmarks

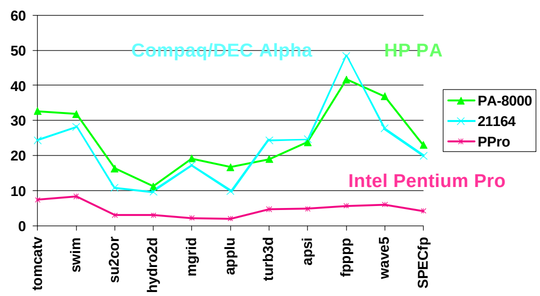
° Workstations: Standard Performance Evaluation Corporation (SPEC)

- SPEC95: 8 integer (gcc, compress, li, ijpeg, perl, ...) & 10 floating-point programs (hydro2d, mgrid, applu, turbo3d, ...)
- www.spec.org
- Separate average for integer (CINT95) and FP (CFP95) relative to base machine
- Benchmarks distributed in source code
- Company representatives select workload
- Compiler, machine designers target benchmarks, so try to change every 3 years

SPECint95base Performance (Oct. 1997)



SPECfp95base Performance (Oct. 1997)



Example PC Workload Benchmark

° PCs: Ziff Davis WinStone 99 Benchmark

- "Winstone 99 is a system-level, application-based benchmark that measures a PC's overall performance when running today's top-selling Windows-based 32-bit applications through a series of scripted activities and uses the time a PC takes to complete those activities to produce its performance scores. Winstone's tests don't mimic what these programs do; they run actual application code."
- www1.zdnet.com/zdbop/winstone/winstone.html
- (See site)

From Sunday Chronicle Ads (4/18/99)

Company	Clock	Processor	Price
emachines	*333	Cyrix MII	$ 499
CompUSA	400	Intel Celeron	$ 780
Compaq	350	AMD K6-2	$ 900
HP	366	Intel Celeron	$ 1,100
Compaq	450	AMD K6-2	$ 1,530
Compaq	400	AMD K6-3	$ 1,599
HP	400	Intel Pentium II	$ 1,450
NEC	400	Intel Pentium III	$ 1,800

(Ads from Circuit City, CompUSA, Office Depot, Staples)

From Sunday Chronicle Ads (4/18/99)

Company	Clock	Processor	Price	Adj Price
emachines	*333	Cyrix MII	$ 499	$ 653
CompUSA	400	Intel Celeron	$ 780	$ 764
Compaq	350	AMD K6-2	$ 900	$ 902
HP	366	Intel Celeron	$1,100	$ 1,070
Compaq	450	AMD K6-2	$1,530	$ 1,453
Compaq	400	AMD K6-3	$1,599	$ 1,479
HP	400	Intel Pentium II	$1,450	$ 1,483
NEC	400	Intel Pentium III	$1,800	$ 1,680

(Ads from Circuit City, CompUSA, Office Depot, Staples)

° Adjusted Price: 128 MB (+$1/MB if less), 10 GB disk ($18/GB), -$100 if included printer, 15" monitor: -$120 if 17", +$50 if 14" monitor

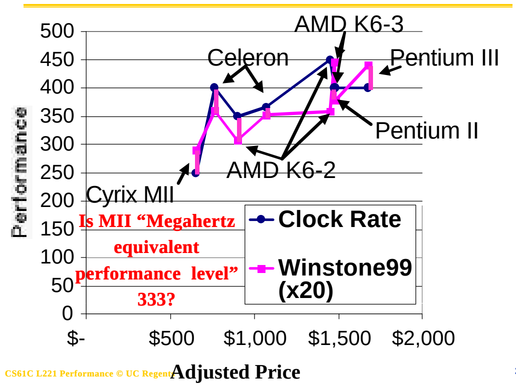
* "Megahertz equivalent performance level."
(Actually 250 MHz Clock Rate)

Winstone 99 (W99) Results

Company	Processor	Price	Clock	W99
emachines	Cyrix MII	$ 653	250	14.5
CompUSA	Intel Celeron	$ 764	400	18.0
Compaq	AMD K6-2	$ 902	350	15.4
HP	Intel Celeron	$1,070	366	17.6
Compaq	AMD K6-2	$1,453	450	17.9
Compaq	AMD K6-3	$1,479	400	22.3
HP	Intel Pentium II	$1,483	400	18.9
NEC	Intel Pentium III	$1,680	400	22.0

° Note: 2 Compaq Machines using K6-2 v. 6-3: K6-2 Clock Rate is 1.125 times faster, but K6-3 Winstone 99 rating is 1.25 times faster!

Adjusted Price v. Clock Rate, Winstone99



Performance Evaluation

- Good products created when have:
  - Good benchmarks
  - Good ways to summarize performance
- Given sales is a function of performance relative to competition, should invest in improving product as reported by performance summary?
- If benchmarks/summary inadequate, then choose between improving product for real programs vs. improving product to get more sales; Sales almost always wins!

Things to Remember

- Latency v. Throughput
- Performance doesn't depend on any single factor: need to know Instruction Count, Clocks Per Instruction and Clock Rate to get valid estimations
- User Time: time user needs to wait for program to execute: depends heavily on how OS switches between tasks
- CPU Time: time spent executing a single program: depends solely on design of processor (datapath, pipelining effectiveness, caches, etc.)