CS61C
Performance

Lecture 23

April 21, 1999

Dave Patterson
(http.cs.berkeley.edu/~patterson)

www-inst.eecs.berkeley.edu/~cs61c/schedule.html

Outline

- Review HP-PA, Intel 80x86 instruction sets
- Motivate Understanding Performance
- What is Faster?
- How Measure Time?
- Elapsed Time vs. User Time
- Administrivia, "What's it good for?"
- "Iron Triangle" of CPU Performance
- Selecting Programs for Comparisons
- Conclusion

Review 1/1

- Once you've learned one RISC instruction set, easy to pick up the rest
  - ARM, Compaq/DEC Alpha, Hitatchi SuperH, HP PA, IBM/Motorola PowerPC, Sun SPARC, ...
  - 16-32 Registers, Load-Store, fixed-length instructions, 3-address instr., aligned data
  - RISC emphasis: performance, HW simplicity
  - HP-PA: more complex addressing than MIPS
- Intel 80x86 is a horse of another color
  - 8 Registers, Register-memory, variable-length instructions, 2-address instr., unaligned data
  - 80x86 emphasis: code size

From Sunday Chronicle Ads (4/18/99)

Company		Processor	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)

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 Concodre	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 <u>Response Time</u> or <u>Execution Time</u>
- 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
  - $\text{performance}(x) = \frac{1}{\text{execution\_time}(x)}$

" 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, ...
  - "wall-clock 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)

Example showing CPU, Elapsed Times

- Unix `time` command: (measure `cache2` while another `cache2` process in background)

```
% time ~cs61c/lib/cache2

% 152.1u 0.9s 4:58 51%
```

- User CPU time is 152.1 seconds, System CPU time is 0.9 seconds, Elapsed time is 4 minutes and 58 seconds (298 seconds), and the percentage of elapsed time that is CPU time is (152.1 + 0.9) / 298 = 51%

How Measure Time?

- User ⇒ seconds
- CPU designer ⇒ measure relating to how fast hardware can perform basic functions
  - Simpler-to-remember, fair measure?
- 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

- CPU execution time for program

  = Clock Cycles for a program
  x Clock Cycle Time

- or

  = Clock Cycles for a program / Clock Rate

Administrivia

- Project 6 (last): MIPS sprintf; Due Wed April 28
- Next Readings: A.6 Friday; 6.1 for Wed.
- 10th homework: Due Today 4/21 7PM
  - Exercises 4.43, 3.17
- 11th homework: Due Friday 4/30 7PM
  - Exercises 2.6, 2.13, 6.1, 6.3, 6.4

Administrivia: Rest of 61C

F 4/23 Review: Procedures, Variable Args; A.6
(Due: x86/HP ISA lab, homework 10)

W 4/28 Processor Pipelining; Section 6.1
F 4/30 Review: Caches/TLB/VM; Section 7.5
(Due: Project 6-sprintf in MIPS, homework 11)

M 5/3 Deadline to correct your grade record

W 5/5 Review: Interrupts / Polling; A.7
F 5/7 61C Summary / Your Cal heritage
(Due: Final 61C Survey in lab)

Sun 5/9 Final Review starting 2PM (1 Pimintel)

W 5/12 Final (5PM 1 Pimintel)

- Need Alternative Final? Contact mds@cory

"What's This Stuff Good For?"

How far will computers/ Internet change our economy?

Buyers Flock to Online Auctions Millions of Americans are sweating silently at their computers, bidding electronically on things they have never seen, tendered by people they have never met.While retailers continue to spend furiously to draw more customers online, the Internet is suddenly teeming with buyers and sellers making their own markets. (Bill Steinhouer jokes on the hood of his 1987 Mercedes 300 turbo diesel he bought for $5,800 on EBAY.com.)
N.Y. Times, 4/13/99

Measuring Time using Clock Cycles

- 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")

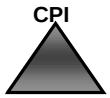
- CPI one way to compare two machines with same instruction set, since Instruction Count would be the same
- CPI also gives insight into style of ISA: RISC higher instruction count, lower CPI

"Iron Triangle" of Performance

- 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



Showing Iron Triangle Derivation

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

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

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

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

- Product of all 3 terms: if missing a term, can't predict time, the real measure of performance
- What's missing from Slide 4?

How Calculate the 3 Components?

- Clock Cycle Time: in specification of computer (Clock Rate in advertisements)
- Instruction Count:
  - Part of Lab 11: count instructions in loop of small program
  - 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 per instruction
  (know how long each instruction takes?)

$$\text{CPU time} = \text{ClockCycleTime} * \sum_{i=1}^{n} \text{CPI}_i * I_i$$

- Then Calculate frequency per instruction
  (know how often each instruction occurs?)

$$\text{CPI} = \sum_{i=1}^{n} \text{CPI}_i * F_i \text{ where } F_i = \frac{I_i}{\text{Instruction Count}}$$

F_i called "instruction frequency"

Example (RISC processor)

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

"Instruction Mix"

- 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
- $\text{CPI}_{\text{memory}}$
  = 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

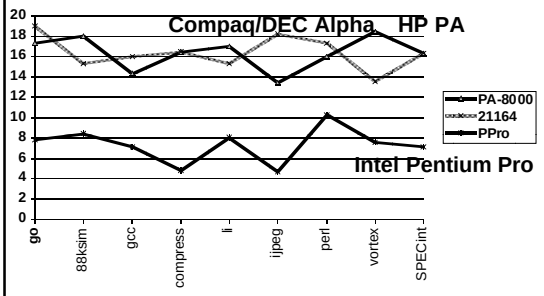
What Programs Measure for Comparison?

- 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

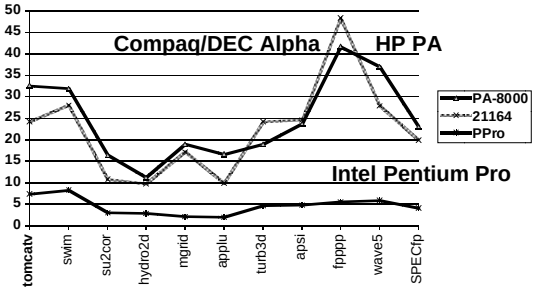
Example Standardized Workload Benchmarks

- 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

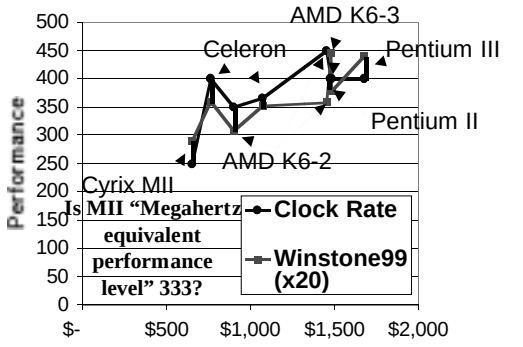
- PCs: Ziff Davis WinStone 99 Benchmark
  - "Winstone 99 is a system-level, application-based benchmark that measures a PC's overall performance when running today's top-selling Windows-based 32-bit applications through a series of scripted activities and uses the time a PC takes to complete those activities 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

Winstone 99 (W99) Results

Company	Processor	Price		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		$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!

"And in Conclusion.." 1/1

- Response Time v. Throughput
- "X times faster than Y" is Time(Y)/Time(X) or Performance(X)/Perfomance(Y)
- Iron triangle: need product of all terms to have meaningful comparison: Instruction Count, Clocks Per Instruction, Clock Rate
- "For Better or Worse, Benchmarks Shape a Field"
  - Benchmark selection important as well as proper metrics
- Next: Review of Stack Parameter Passing