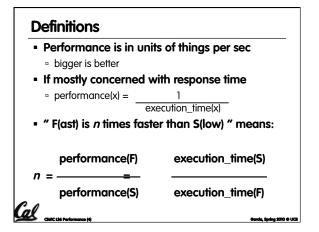
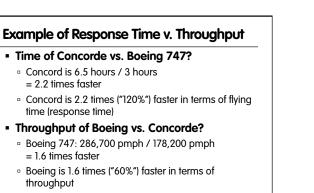


Plane	DC to Paris	Top Speed	Passen- gers	Throughput (pmph)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200
InterestedInterested		er 100 passen many passer	gers? ngers per day as	possible?
 In a comp 	uter, time f nse Time or			





• We will focus primarily on response time.

Words, Words, Words...

Cal CHELDA PAR

Barcia, Spring 2010 © UCB

- Will (try to) stick to "n times faster"; its less confusing than "m % faster"
- As faster means both <u>decreased</u> execution time and <u>increased</u> performance, to reduce confusion we will (and you should) use <u>"improve execution time</u>" or <u>"improve performance</u>"

Garda, Spring 2010 © UCB

What is Time?

Cal _

- 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)
 - "<u>CPU execution time</u>" or "<u>CPU time</u>"
 - Often divided into <u>system CPU time (in OS)</u> and <u>user CPU time</u> (in user program)

How to Measure Time? Real Time ⇒ Actual time elapsed 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., ½ nanoseconds or ½ ns) and clock rate

(e.g., 2 gigahertz, or 2 GHz), which is the inverse of

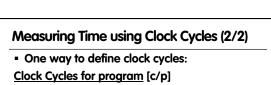
Measuring Time using Clock Cycles (1/2)

CPU execution time for a program
Units of [seconds / program] or [s/p]

Clock Cycles for a program x Clock Period
Units of [s/p] = [cycles / p] x [s / cycle] = [c/p] x [s/c]

Or

Clock Cycles for a program [c / p]
Clock Rate [c / s]



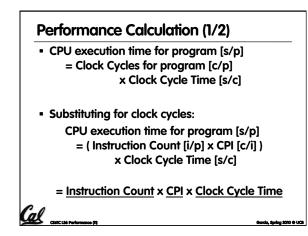
= Instructions for a program [i/p] (called "<u>Instruction Count</u>")

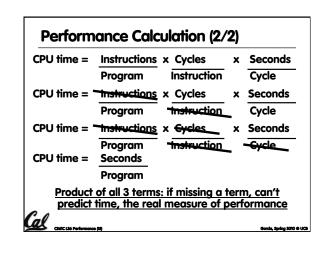
the clock period; use these!

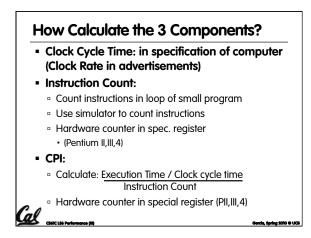
Cal

Cal

- x Average <u>Clock cycles</u> <u>Per</u> <u>Instruction</u> [c/i] (abbreviated "<u>CPI</u>")
- CPI one way to compare two machines with same instruction set, since Instruction Count would be the same







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 (the weighted sum)

Garcia, Spring 2010 & UCI

Garda, Spring 2010 © UCB

Ор	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%)
<u>In</u>	struction	2.2	(Where time spent)	
Vhat if Br	anch instr	uctions	twice o	as fast?

What Programs Measure for Comparison?Ideally run typical programs with typical input

before purchase, or before even build machine

Called a "workload"; For example:

Cal Cal Cal Performance (M)

Cal .

- 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

Cal Charles Parlies

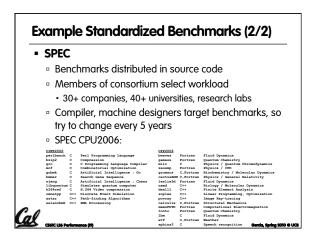
- Obviously, apparent speed of processor depends on code used to test it
- Need industry standards so that different processors can be fairly compared
- Companies exist that create these benchmarks: "typical" code used to evaluate systems
- Need to be changed every ~5 years since designers could (and do!) target for these standard benchmarks

Example Standardized Benchmarks (1/2) Standard Performance Evaluation Corporation (SPEC) SPEC CPU2006 CINT2006 12 integer (perl, bzip, gcc, go, ...) CFP2006 17 floating-point (povray, bwaves, ...) All relative to base machine (which gets 100) Sun Ultra Enterprise 2 w/296 MHz UltraSPARC II Thay magneture

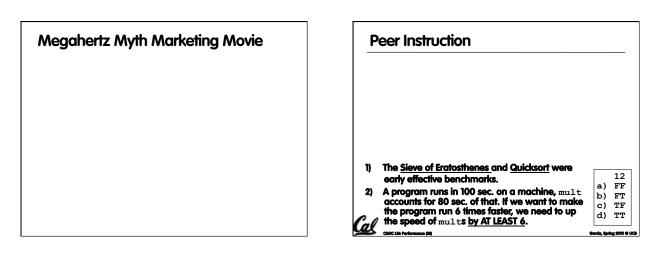
They measure

Call CLAS Parform

- System speed (SPECint2006)
- System throughput (SPECint_rate2006)
- www.spec.org/osg/cpu2006/



Performance Evaluation: The Demo If we're talking about performance, let's discuss the ways shady salespeople have fooled consumers (so you don't get taken!) 5. Never let the user touch it 4. Only run the demo through a script 3. Run it on a stock machine in which "no expense was spared" 2. Preprocess all available data 1. Play a movie



Cal .

"And in conclusion"							
CPU time =	Instructions	x Cycles	x	Seconds			
Program	Instruction	Cycle	-				
 Latency 	v. Throughput						
 Perform Instruction Rate to get 	ance doesn't de on Count, Clocks get valid estimat	pend on any s Per Instructio ions	single (n (CPI)	factor: need and Clock			
 User Tin depends 	e: time user wa heavily on how	its for program OS switches	n to e> betwe	ecute: en tasks			
	e: time spent ex solely on desig g effectiveness,	ecuting a sing n of processo caches, etc.)	gle pro r (data	gram: path,			
 Benchr 	narks						
	to predict perf, l						
■ Measur prograr ■ Megah ■ MHz ≠	e everything from ns to battery life ertz Myth performance, it's	n simulation of s iust one facto	deskto	op graphics			
ax				Garda, Sarina 2010 O L			