Performance – measurement and prediction

– Throughput and latency

$$\text{Performance} = \frac{1}{\text{execution time}}$$

$$\text{Performance}_x > \text{Performance}_y \iff \text{execution time}_x < \text{execution time}_y$$

factors in CPU performance.

CPU execution time for P = CPU clock cycles for P * Clock period

Clock period = 1/clock rate

CPU clock cycles for P = # Instructions for P * Average clock cycles / instruction for P

Computer A: Clock period .3 ns

Computer B: Clock period .5 ns

$CPI_B = 1.2 \Rightarrow .6$ ns/Inst } same execution time

$CPI_A = 2 \Rightarrow .6$ ns/Inst }

$$\text{Total Time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\#\text{Inst}}{\text{Program}} \cdot \frac{\text{cycles}}{\text{Inst}} \cdot \frac{\text{seconds}}{\text{cycle}}$$

Alg affects: #Inst, CPI

Prog Lang: #Inst, CPI

Compiler: #Inst, CPI

Architecture: #Inst, clock rate, CPI

Performance Evaluation

The art of benchmarking:

"big enough" to be meaningful
- want to test overall system
- want to test on what you care about

"too many variables" – HW, compiler, OS, cache, memory speed, disk speed, bus speed.
- had to generalize

Traps for the unwary

	Comp A	Comp. B
Prog 1	1	10 seconds
Prog 2	1000	100 seconds
	1001	110

How much faster than Computer A is computer B?
10, 1/10, 9.1

what if we don't run an equal number of Prog 1 & Prog 2 jobs?

⟹ Have to look at your own workload

SPEC Benchmarks (CPU)

integer

- gzip
- circuit placement
- gcc
- comb. opt.
- Chess
- computer visual.
- perl benchmark
- group theory
- OO. database
- bzip2
- place & route

floating point

SpecWeb (server whole system BM)

- Client request stream
- measures HW + SW of the web server

} measure transactions / sec

Database benchmarks

performance of different kinds of ~~the~~ database transactions
— read, write, bank transaction ⇒ withdrawl, funds transfer.

Graphics Benchmarks (gaming)

- frames per second
- rectangles / second
- triangles / second

} ideally — have to measure what you care about

Other performance criteria:

a. Cost of operation } energy-related
b. heat produced
c. battery lifetime
d. ability to trade off a.b.c. for cpu/system performance.

Used to talk lots about MIPS — Millions of instructions per second.
arose in days of IBM 360/370 — most computers had same architecture —
only semi-valid even when same ISA architecture —

why

Same ISA architecture:
- different implementations have different CPI for different instructions
- different ~~applications~~ programs have different mixes of instructions
- non-CPU perf. also matters.

different ISA:
- same HLL program have different # of instructions

⇒ IPS does not make much sense as a ~~good~~ very rough gauge of performance.

Notes Based on Patterson and Hennessy Computer Org and Design. Ch. 4