Power / Energy

Energy → measured in Joules / watt-sec

Power → energy per unit time (watts)
↳ related to performance (which is a "per unit time" metric)

power
↙ peak power (when you actually take a picture)
↘ average power

Dynamic Power = Active switching of transistors
Static power = leakage of transistors even while inactive

Energy $\propto$ Capacitive load $\times$ Voltage^2

Power = energy $\times$ frequency of transistors

Power $\propto$ capacitive load $\times$ voltage^2 $\times$ frequency

capacitive load ↓ # of transistors connected / technology

frequency ↓ function of clock rate

SPEC Benchmarks

$$\text{Execution Time Ratio}_i = \frac{\text{Exec. Time Referece Machine}}{\text{Exec. Time Given Machine}_i}$$

$$\sqrt[n]{\prod \text{execution time ratio}_i}$$

Why Geometric Mean → Same relative answer no matter what computer is used to normalize the results

Arithmetic Mean → Result depend on the choice of the machine

	Time on A	Time on B	Normalize to A: A	Normalize to A: B	Normalize to B: A	Normalize to B: B
Program 1	1	10	1	10	0.1	1
Program 2	1000	10	1	0.1	10	1
Arithmetic Mean on Norm. Time	500.5	55	1	5.05	5.05	1
Geometric Mean on Norm. Time	31.6	31.6	1	1	1	1

Arithmetic Mean
- when normalized to A → A is faster by 5.05
- when normalized to B → B is faster by 5.05

(BOTH WRONG)

Geometric Mean is independent of normalization

Amdahl's LAW

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

Speedup = Measure of how a machine performs after some enhancement relative to how it performed previously.

$$\text{Speedup} = \frac{T_{unimproved}}{T_{improved}}$$

Question 1

Suppose we enhance a machine making all floating-point instructions run five times faster. If the execution time of some benchmark before the floating-point enhancement is 10 seconds, what will the speedup be if half of the 10 seconds is spent executing floating-point instructions?

Solution

$$T_{unimproved} = 10 \text{ sec.}$$

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

$$T_{improved} = \frac{5 \text{ sec}}{5} + 5 = 6$$

$$\text{Speedup} = \frac{10}{6} \approx 1.6$$

Question 2

We are looking for a benchmark to show off the new floating-point unit described above, and want the overall benchmark to show a speedup of 3. One benchmark we are considering runs for 100 seconds with the old floating-point hardware. How much of the execution time would floating-point instructions have to account for in this program in order to yield our desired speedup on this benchmark?

Solution

$$\text{Speedup} = \frac{T_{\text{unimproved}}}{T_{\text{improved}}} = 3 \qquad 100\text{sec.}$$

$$T_{\text{improved}} = 100/3$$

$$T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improved factor}} + T_{\text{unaffected}}$$

$$= \frac{x}{5} + (100 - x) = 100/3$$

$$\frac{x}{5} + (100 - x) = 100/3 \qquad x \cong 83\text{ sec.}$$

Question 3

- Two different compilers are being tested for a 4 GHz. machine with three different classes of instructions: Class A, Class B, and Class C, which require one, two, and three cycles (respectively). Both compilers are used to produce code for a large piece of software.

  The first compiler's code uses 5 million Class A instructions, 1 million Class B instructions, and 1 million Class C instructions.

  The second compiler's code uses 10 million Class A instructions, 1 million Class B instructions, and 1 million Class C instructions.

- Which sequence will be faster according to MIPS?
- Which sequence will be faster according to execution time?

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

SOLUTION

Class A = 1 Class B = 2 Class C = 3

Code 1 → 5M A, 1M B, 1M C

Code 2 → 10M A, 1M B 1M C

$$\text{MIPS}_1 = \frac{\text{Clock rate}}{\text{CPI} \times 10^6} = \frac{4 \times 10^9}{\frac{(5\times1 + 1\times2 + 1\times3)}{7} \times 10^6} = 2800$$

$$\text{MIPS}_2 = \frac{4 \times 10^9}{\frac{(10\times1 + 1\times2 + 1\times3)}{12} \times 10^6} = 3200$$

Compiler 2 is better (according to MIPS)

$$\text{CPU time} = \underbrace{IC \times CPI}_{(\#\text{of clock cycles})} \times CCT$$

$$\text{CPU time}_1 = (5\times1 + 1\times2 + 1\times3) \times 10^6 \times \frac{1}{4\times10^9}$$

$$= 0.0025 \text{ sec.}$$

$$\text{CPU time}_2 = (10\times1 + 1\times2 + 1\times3) \times 10^6 \times \frac{1}{4\times10^9}$$

$$= 0.003 \text{ sec.}$$

Compiler 1 is better (according to Execution Time)

Solution for Midterm Question

add \$s1, \$s2, \$s3 → lwnew
lw \$s5, 0(\$s1)

add \$s1, \$s2, \$s3 → swnew
sw \$s5, 0(\$s1)

add \$s1, \$s2, \$s3 ⇒ \$s1 = \$s2 + \$s3
lw \$s5, 0(\$s1) ⇒ \$s5 = Memory[0 + \$s1]
sw \$s5, 0(\$s1) ⇒ Memory[0 + \$s1] ← \$s5

Part A

Applicable for 25% of LW/SW

LW & SW affected

$IC_{new} = IC_{old} -$

$IC_{old} \times (0.24 \times 0.25 +$
$0.16 \times 0.25)$

$= IC_{old} - IC_{old} \times (0.06 +$
$0.04)$

$IC_{new} = IC_{old} - IC_{old} \times 0.1$

$= \boxed{0.9\, IC_{old}}$

Part B

$CPUTime_{old} = IC_{old} \times CPI_{old} \times CCT_{old}$

$CPUTime_{new} = IC_{new} \times CPI_{new} \times$
CCT_{new}

$CPUTime_{new} = 0.9 \times IC_{old} \times CPI_{old}$
$\times (1.06 \times CPI_{old})$

CPUTime old

$= (0.9 \times 1.06) \times \boxed{IC_{old} \times CPI_{old} \times CCT_{old}}$

$= 0.954 \times CPUTime_{old}$ (New is faster)