EE 424
DIGITAL SYSTEM ARCHITECTURE

Chapter 2

Performance

Performance

- Measure, Report, and Summarize
- Make intelligent choices
- See through the marketing hype
- Key to understanding underlying organizational motivation

Why is some hardware better than others for different programs?

What factors of system performance are hardware and/or software related?
(e.g., Do we need a new machine, or a new operating system?)

How does the machine's instruction set affect performance?

Which of these airplanes has the best performance?

Airplane	Passengers	Range (mi)	Speed (mph)
Boeing 737-100	101	630	598
Boeing 747	470	4150	610
BAC/Sud Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544

•How much faster is the Concorde compared to the 747?

1350/610 = 2.213

•How much bigger (capacity) is the 747 than the Douglas DC-8?

470/146 = 3.22

Computer Performance: TIME

- Response Time *(Execution Time)*
  - — How long does it take for my job to run?
  - — How long does it take to execute a job?
  - — How long must I wait for the database query?
- Throughput
  - — How many jobs can the machine run at once?
  - — What is the average execution rate?
  - — How much work is getting done?

- *If we upgrade a machine with a new processor what do we increase?*
- *If we add a new machine to the lab what do we increase?*

Execution Time

- Elapsed Time
  - counts everything *(disk and memory accesses, I/O , etc.)*
  - a useful number, but often not good for comparison purposes
- CPU time
  - doesn't count I/O or time spent running other programs
  - can be broken up into system time, and user time
- Our focus: user CPU time
  - time spent executing the lines of code that are "in" our program

Performance

- For some program running on machine X,
  $Performance_X = 1 / Execution\ time_X$
- "X is n times faster than Y"
  $Performance_X / Performance_Y = n$
- Problem:
  - machine A runs a program in 20 seconds
  - machine B runs the same program in 25 seconds

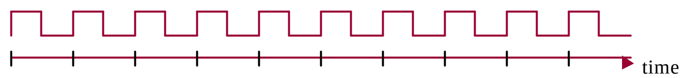
$$\frac{Performance_A}{Performance_B} = \frac{Ex.\ time_B}{Ex.\ time_A} = \frac{25}{20} = \mathbf{1.25}$$

Clock Cycles

- Instead of reporting execution time in seconds, we often use cycles

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

- Clock "ticks" indicate when to start activities (one abstraction):



- cycle time = time between ticks = seconds per cycle
- clock rate (frequency) = cycles per second (1 Hz. = 1 cycle/sec)

A 200 Mhz. clock has a $\frac{1}{200 \times 10^{6}} \times 10^{9} = 5 \text{ nanoseconds}$ cycle time

How to Improve Performance

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

So, to improve performance (everything else being equal) you can either

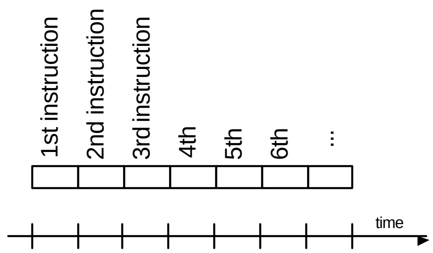
________ the # of required cycles for a program, or

________ the clock cycle time or, said another way,

________ the clock rate.

How many cycles are required for a program?

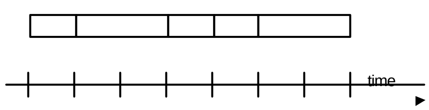
- Could assume that # of cycles = # of instructions



This assumption is incorrect,

different instructions take different amounts of time on different machines.

Instructions and number of cycles



- Multiplication takes more time than addition
- Floating point operations take longer than integer ones
- Accessing memory takes more time than accessing registers

- *Important point: changing the cycle time often changes the number of cycles required for various instructions (more later)*

Example

- Our favorite program runs in 10 seconds on computer A, which has a 400 MHz clock. We are trying to help a computer designer build a new machine B, that will run this program in 6 seconds. The designer can use new (or perhaps more expensive) technology to substantially increase the clock rate, but has informed us that this increase will affect the rest of the CPU design, causing machine B to require 1.2 times as many clock cycles as machine A for the same program. What clock rate should we tell the designer to target?"
- Don't Panic, you can easily work this out from basic principles

Let us take a look at computer A

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$\text{CPU clock cycles}_A = \text{CPU time}_A \times \text{Clock rate}_A$$

$$= 10\ \text{sec} \times 4\text{x}10^8\ \text{cycles/sec}$$

$$\text{CPU clock cycles}_A = 4\text{x}10^9\ \text{cycles}$$

What about computer B

$$\text{CPU time}_B = \frac{\text{CPU clock cycles}_B}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{\text{CPU clock cycles}_B}{\text{CPU time}_B}$$

$$= \frac{1.2 \times \text{CPU clock cycles}_A}{\text{CPU time}_B}$$

$$= \frac{1.2 \times 4 \times 10^9 \text{ cycles}}{6 \text{ sec}} = 800 \text{ MHz}$$

Now that we understand cycles

- A given program will require
  - some number of instructions (machine instructions)
  - some number of cycles
  - some number of seconds
- We have a terminology that relates these quantities:
  - cycle time (seconds per cycle)
  - clock rate (cycles per second)
  - CPI (cycles per instruction)

    a floating point intensive application might have a higher CPI
  - MIPS (millions of instructions per second)

    this would be higher for a program using simple instructions

Performance

- Performance is determined by execution time
- Do any of the other variables equal performance?
  - Number of cycles to execute program?
  - Number of instructions in program?
  - Number of cycles per second?
  - average number of cycles per instruction?
  - average number of instructions per second?

- Common pitfall: thinking one of the variables is indicative of performance when it really isn't.

CPI Example

- Suppose we have two implementations of the same instruction set architecture (ISA).

  For some program,

  Machine A has a clock cycle time of 10 ns. and a CPI of 2.0
  Machine B has a clock cycle time of 20 ns. and a CPI of 1.2

  What machine is faster for this program, and by how much?

- *If two machines have the same ISA which of our quantities (e.g., clock rate, CPI, execution time, # of instructions, MIPS) will always be identica l?*

Answer (p. 62)

$$\text{CPU clock cycles}_A = I \times 2.0$$

$$\text{CPU clock cycles}_B = I \times 1.2$$

Now we can compute the CPU time for each machine:

$$\text{CPU time}_A = \text{CPU clock cycles}_A \times \text{Clock cycle time}_A$$
$$= I \times 2.0 \times 1 \text{ ns} = 2 \times I \text{ ns}$$

Likewise, for B:

$$\text{CPU time}_B = I \times 1.2 \times 2 \text{ ns} = 2.4 \times I \text{ ns}$$

Clearly, machine A is faster. The amount faster is given by the ratio of the execution times:

$$\frac{\text{CPU performance}_A}{\text{CPU performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{2.4 \times I \text{ ns}}{2 \times I \text{ ns}} = 1.2$$

CPU time

IC = Instruction count (executed instructions)

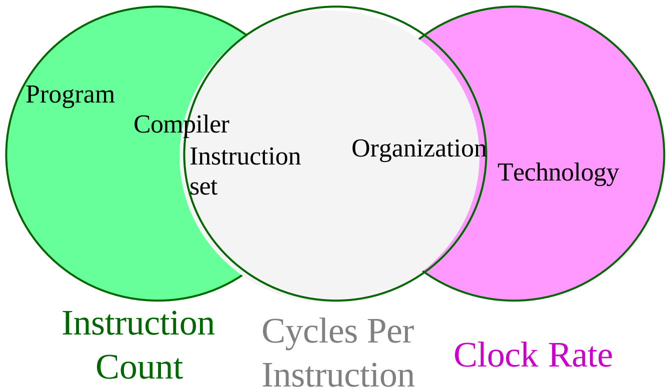
$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{clock cycle}}$$

CPI= Cycles Per Instruction

cct= clock cycle time

CPU time = IC x CPI x cct

CPU time



Number of Instructions Example

- A compiler designer is trying to decide between two code sequences for a particular machine. Based on the hardware implementation, there are three different classes of instructions: Class A, Class B, and Class C, and they require one, two, and three cycles (respectively).

  The first code sequence has 5 instructions:
  - 2 of A, 1 of B, and 2 of C

  The second sequence has 6 instructions:
  - 4 of A, 1 of B, and 1 of C.
- Which sequence will be faster? How much? What is the CPI for each sequence?

MIPS example

- Two different compilers are being tested for a 100 MHz. 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 .

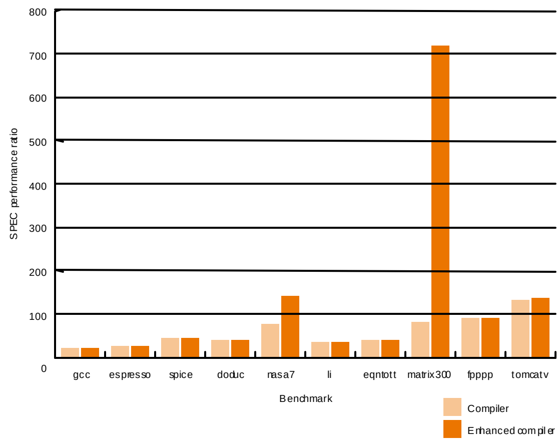
- Which sequence will be faster according to MIPS?
- Which sequence will be faster according to execution time?

Benchmarks

- Performance best determined by running a real application
  - Use programs typical of expected workload
  - Or, typical of expected class of applications
    e.g., compilers/editors, scientific applications, graphics, etc.
- Small benchmarks
  - nice for architects and designers
  - easy to standardize
  - can be abused
- SPEC (System Performance Evaluation Cooperative)
  - companies have agreed on a set of real program and inputs
  - can still be abused (Intel's "other" bug)
  - valuable indicator of performance (and compiler technology)

SPEC '89

- Compiler "enhancements" and performance



SPEC '95

Benchmark	Description
go	Artificial intelligence; plays the game of Go
m88ksim	Motorola 88k chip simulator; runs test program
gcc	The Gnu C compiler generating SPARC code
compress	Compresses and decompresses file in memory
li	Lisp interpreter
ijpeg	Graphic compression and decompression
perl	Manipulates strings and prime numbers in the special-purpose programming language F
vortex	A database program
tomcatv	A mesh generation program
swim	Shallow water model with 513 x 513 grid
su2cor	quantum physics; Monte Carlo simulation
hydro2d	Astrophysics; Hydrodynamic Naiver Stokes equations
mgrid	Multigrid solver in 3-D potential field
applu	Parabolic/elliptic partial differential equations
trub3d	Simulates isotropic, homogeneous turbulence in a cube
apsi	Solves problems regarding temperature, wind velocity, and distribution of pollutant
fpppp	Quantum chemistry
wave5	Plasma physics; electromagnetic particle simulation

SPEC '95

Does doubling the clock rate double the performance?

Can a machine with a slower clock rate have better performance?