100 Performance

ENGR 3410 - Computer Architecture

Mark L. Chang

Fall 2009

What things are important when buying a computer?

What features do you look for when buying one?

- Warranty/veliability/vobustness

Computer "Performance"

- MIPS (Million Instructions Per Second) vs. MHz (Million Cycles Per Second)
- Throughput (jobs/seconds) vs. Latency (time to complete a job)
- Measuring, Metrics, Evaluation what is "best"?

Hyper
Pipelined
Technology

3.09 GHz Pentium 4

The PowerBook G4 outguns Pentium III-based notebooks by up to 30 percent.*

* Based on Adobe Photoshop tests comparing a 500MHz PowerBook G4 to 850MHz Pentium III-based portable computers

Performance Example: Planes

Airplane	Passenger Capacity	Cruising Range (miles)	Cruising Speed (mph)	Passenger Throughput (passengermile/hour)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
Concorde	132	4000	1350	178,200
Douglas DC8	146	8720	544	79,424

- Which is the "best" plane?
 - Which gets one passenger to the destination first?



- Which moves the most passengers? 747
- Which goes the furthest? DCS
- Which is the "speediest" plane (between Seattle and NY)?
 - Latency: how fast is one person moved?
 - Throughput: number of people per time moved?

Computer Performance

• Primary goal: execution time (time from program start to program completion)

$$Perf \ orman e = \frac{1}{Execution Time}$$

To compare machines, we say "X is n times faster than Y"

$$n = \frac{Perf\ orman \boldsymbol{e}_x}{Perf\ orman \boldsymbol{e}_y} = \frac{ExecutionTime_y}{ExecutionTime_x}$$

• Example: Machine *Orange* and *Grape* run a program
Orange takes 5 seconds, Grape takes 10 seconds

Orange is _____ times faster than Grape

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
- Example: Unix "time" command

```
fpga.olin.edu> time javac CircuitViewer.java
3.370u 0.570s 0:12.44 31.6%
```

- Our focus: user CPU time
 - time spent executing the lines of code that are "in" our program

CPU Time

CPU execution time = CPU clock cycles * 1 Clock rate

Application example:

A program takes 10 seconds on computer *Orange*, with a 400MHz clock. Our design team is developing a machine *Grape* with a much higher clock rate, but it will require 1.2 times as many clock cycles. If we want to be able to run the program in 6 second, how fast must the clock rate be?

CPI

• How do the # of instructions in a program relate to the execution time?

CPI Example

- Suppose we have two implementations of the same instruction set (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?

CPO Clock (yclas = N x 2.0

B = N x 1.2

(PU Time A = N x 2.0 x 10 ns =
$$20x$$
 This

B = N x 1.2 x 20 us = $24x$ This

Computing CPI

- Different types of instructions can take very different amounts of cycles
- Memory accesses, integer math, floating point, control flow

$$CPI = \sum_{types} (Cycles_{type} * Frequency_{type})$$

Instruction Type	Type Cycles	Type Frequency	Cycles * Freq
ALU	1	50%	0,5
Load	5	20%	/1 ()
Store	3	10%	0.3
Branch	2	20%	0,4
		CPI:	22

CPI & Processor Tradeoffs

Instruction Type	Type Cycles	Type Frequency
ALU	1	50%
Load	5	20%
Store	3	10%
Branch	2	20%

How much faster would the machine be if:

1. A data cache reduced the average load time to 2 cycles?

$$\frac{2.2}{1.6} = 1.3$$

2. Branch prediction shaved a cycle off the branch time?

Brand
$$0.4 > 0.2$$
 $\frac{2.2}{2.0} = \left[1.1 \times \right]$

3. Two ALU instructions could be executed at once?

$$ALU 1 \rightarrow 0.5 \quad 0.5 \rightarrow 0.25 \quad \frac{2.2}{1.95} = \sqrt{1.2} \times$$

Warning 1: Amdahl's Law

The impact of a performance improvement is limited by what is NOT improved:

Execution time after improvement = Execution time to fundificated to fundificated to fundificated to fundificate to fundificat

• Example: Assume a program runs in 100 seconds on a machine, with multiply responsible for 80 seconds of this time. How much do we have to speed up multiply to make the program run 4 times faster?



$$255 = 205 + 805 \times \frac{1}{N}$$
 $N = \frac{80}{5} = \frac{16}{5}$

• 5 times faster?



Warning 2: MIPs, MHz ≠ Performance

• Higher MHz (clock rate) doesn't always mean better CPU Orange computer: 1000 MHz, CPI: 2.5, 1 billion instruction program

Grape computer: 500MHz, CPI: 1.1, 1 billion instruction program

 Higher MIPs (million instructions per second) doesn't always mean better CPU

1 MHz machine, with two different compilers

Compiler A on program X: 10M ALU, 1M Load

Compiler B on program X: 5M ALU, 1M Load

Execution Time: A _____ B ____

MIPS: A B

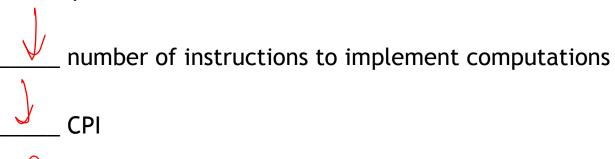
Instruction Type	Type Cycles	
ALU	1	
Load	5	
Store	3	
Branch	2	

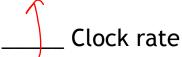
Processor Performance Summary

• Machine performance:

```
CPU execution time for a program * CPI * Clock rate
```

• Better performance:





• Improving performance must balance each constraint Example: RISC vs. CISC