CENG3420

Lecture 04 Review

Bei Yu

byu@cse.cuhk.edu.hk

2017 Spring

香港中文大學
The Chinese University of Hong Kong

Throughput v.s. Response Time

Response time (execution time)

- The time between the start and the completion of a task.
- Important to individual users

Throughput (bandwidth)

- The total amount of work done in a given time
- Important to data center managers

Throughput v.s. Response Time

Response time (execution time)

- The time between the start and the completion of a task.
- Important to individual users

Throughput (bandwidth)

- The total amount of work done in a given time
- Important to data center managers

Will need different performance metrics as well as a different set of applications to benchmark embedded and desktop computers, which are more focused on response time, versus servers, which are more focused on throughput

Defining (Speed) Performance

- To maximize performance, need to minimize execution time

$$\text{performance}_X = \frac{1}{\text{execution\_time}_X}$$

- If X is n times faster than Y, then

$$\frac{\text{performance}_X}{\text{performance}_Y} = \frac{\text{execution\_time}_Y}{\text{execution\_time}_X} = n$$

- Decreasing response time almost always improves throughput.

EX-1

If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

Solution:

EX-1

If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

Solution:

The performance ratio is $\frac{15}{10} = 1.5$, so A is 1.5 times faster than B.

Performance Factors

- CPU execution time (CPU time): time the CPU spends working on a task
- Does not include time waiting for I/O or running other programs

$$
\begin{aligned}
\text{CPU execution time} &= \#\ \text{CPU clock cycles} \times \text{clock cycle time} \\
&= \frac{\#\ \text{CPU clock cycles}}{\text{clock rate}}
\end{aligned}
$$

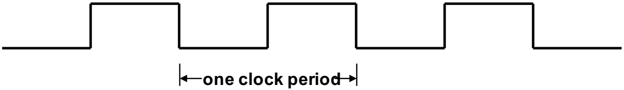
Can improve performance by reducing

- Length of the clock cycle
- Number of clock cycles required for a program

Review: Machine Clock Rate

Clock rate (clock cycles per second in MHz or GHz) is inverse of clock cycle time (clock period)

$$CC = \frac{1}{CR}$$



10 nsec clock cycle => 100 MHz clock rate

5 nsec clock cycle => 200 MHz clock rate

2 nsec clock cycle => 500 MHz clock rate

1 nsec (10^{-9}) clock cycle => 1 GHz (10^{9}) clock rate

500 psec clock cycle => 2 GHz clock rate

250 psec clock cycle => 4 GHz clock rate

200 psec clock cycle => 5 GHz clock rate

EX-2: Improving Performance Example

A program runs on computer A with a 2 GHz clock in 10 seconds. What clock rate must a computer B run at to run this program in 6 seconds? Unfortunately, to accomplish this, computer B will require 1.2 times as many clock cycles as computer A to run the program.

Solution:

- For computer A: cycle # = $10 \times 2 \times 10^9 = 20 \times 10^9$.
- For computer B: exe_time = $1.2 \times 20 \times 10^9$ / clock_rate. Therefore, clock_rate = 4 GHz.

Clock Cycles per Instruction

- Not all instructions take the same amount of time to execute
- One way to think about execution time is that it equals the number of instructions executed multiplied by the average time per instruction

CPU clock cycles = # instruction $\times$ clock cycle per instruction

Clock cycles per instruction (CPI)

- The average number of clock cycles each instruction takes to execute
- A way to compare two different implementations of the same ISA

Effective (Average) CPI

$$\sum_{i=1}^{n} CPI_i \times IC_i$$

IC_i: percentage of the number of instructions of class i executed

CPI_i: (average) number of clock cycles per instruction for that instruction class

n: number of instruction classes

- Computing the overall effective CPI is done by looking at the different types of instructions and their individual cycle counts and averaging
- The overall effective CPI varies by instruction mix
- A measure of the dynamic frequency of instructions across one or many programs

EX-3: Using the Performance Equation

Computers A and B implement the same ISA. Computer A has a clock cycle time of 250 ps and an effective CPI of 2.0 for some program and computer B has a clock cycle time of 500 ps and an effective CPI of 1.2 for the same program. Which computer is faster and by how much?

Solution: Assume each computer executes I instructions, so

$$\text{CPU time}_A = I \times 2.0 \times 250 = 500 \times I \text{ ps}$$
$$\text{CPU time}_B = I \times 1.2 \times 500 = 600 \times I \text{ ps}$$

A is faster by the ratio of execution times:

$$\frac{\text{performance}_A}{\text{performance}_B} = \frac{\text{execution\_time}_B}{\text{execution\_time}_A} = \frac{600 \times I}{500 \times I} = 1.2$$

Basic Performance Equation

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{clock cycle}$$

$$\text{CPU time} = \frac{\text{Instruction count} \times \text{CPI}}{\text{clock rate}}$$

Three key factors that affect performance

- Can measure the CPU execution time by running the program The clock rate is usually given Can measure overall instruction count by using profilers/ simulators without knowing all of the implementation details

CPI varies by instruction type and ISA implementation for which we must know the implementation details

Determinates of CPU Performance

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{clock cycle}$$

	Instruction_ count	CPI	clock_cycle
Algorithm			
Programming language			
Compiler			
ISA			
Core organization			
Technology			

Determinates of CPU Performance

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{clock cycle}$$

	Instruction_ count	CPI	clock_cycle
Algorithm	X	X	
Programming language	X	X	
Compiler	X	X	
ISA	X	X	X
Core organization		X	X
Technology			X

EX-4

Op	Freq	CPI_i	Freq x CPI_i
ALU	50%	1	
Load	20%	5	
Store	10%	3	
Branch	20%	2	
			$\sum =$

- How much faster would the machine be if a better data cache reduced the average load time to 2 cycles?
- How does this compare with using branch prediction to shave a cycle off the branch time?
- What if two ALU instructions could be executed at once?

Solution:

- CPU time new = 1.6 x IC x CC so 2.2/1.6 means 37.5% faster
- CPU time new = 2.0 x IC x CC so 2.2/2.0 means 10% faster
- CPU time new = 1.95 x IC x CC so 2.2/1.95 means 12.8% faster