COMPUTER ORGANIZATION AND DESIGN
The Hardware/Software Interface
5th Edition

Chapter 1

Computer Performance

Transistors

- Fun facts about 45nm transistors:
  - 30 million can fit on the head of a pin.
  - 2,000 fit across the width of a human hair.
  - If car prices had fallen at the same rate as the price of a single transistor has since 1968, a new car today would cost about 1 cent.

Understanding Performance

- Algorithm
  - Determines the number of operations executed.
- Programming language, compiler, architecture
  - Determines the number of machine instructions executed per operation.
- Processor and memory system
  - Determines how fast instructions are executed.

Performance Metrics

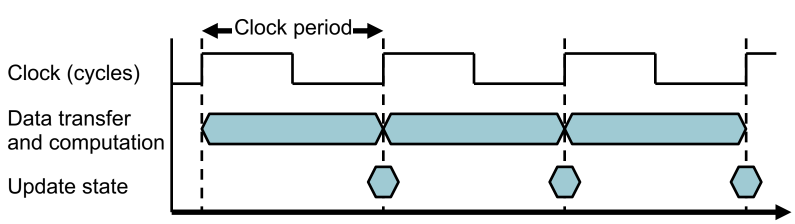
- Possible measures:
  - Response time – elapsed time between start and end of a program (important to individual users).
  - Throughput – amount of work done in a fixed amount of time (important to data centers).
- The two measures are usually linked:
  - A faster processor will improve both.
  - Near-future processors will likely only improve throughput.
  - Some architecture improvements will improve throughput and worsen response time, like pipelining.

Speedup and Improvement

- Example
  1. What is the speedup of System X over System Y if System X executes a program in 10 seconds and system Y executes the same program in 15 seconds? 5 seconds or 1.5 times
  2. What is the percentage reduction in execution time for the program of X compared to Y? (15-10) /15 = 33%
  3. What is the percentage increase in execution time for the program of Y compared to X? (15-10) /10 = 50%

CPU Clocking

- Operation of digital hardware is governed by a constant-rate clock:



- Clock frequency (rate) – cycles per second
  - e.g., 4.0GHz = 4000MHz = 4.0×10^{9}Hz
- Clock period – duration of a clock cycle
  - e.g., 250ps = 0.25ns = 250×10^{-12}s

Performance Equation #1

$$CPU\ execution\ time = (CPU\ clock\ cycles)(clock\ cycle\ time)$$

$$clock\ cycle\ time = \frac{1}{clock\ speed}$$

- Example #1:
  - If a program runs for 10 seconds on a 3 GHz processor, how many clock cycles did it run for? 30 billion
- Example #2:
  - If a program runs for 2 billion clock cycles on a 1.5 GHz processor, what is the execution time in seconds? 1.333

Performance Equation #2

- CPI = Clock Cycles Per Instruction.

$$cpu\ clock\ cycles = (number\ of\ instructions)(CPI)$$

- Substituting in the previous equation,

$$execution\ time = (clock\ cycle\ time)(number\ of\ instructions)(CPI)$$

- Example:
  - If a 2 GHz processor completes an instruction every third cycle, how many instructions are there in a program that runs for 10 seconds? 10(2E9)/3 = 6.667E9

Performance Equation Summary

- Our basic performance equation is then:

$$CPU\ time = (clock\ cycle\ time)(instruction\ count)(CPI)$$

or

$$CPU\ time = \frac{(instruction\ count)(CPI)}{clock\ rate}$$

- These equations separate the key factors that affect performance:
  - The CPU execution time is measured by running the program.
  - The clock rate is usually given.
  - The overall instruction count is measured by using profilers or simulators.
  - CPI varies by instruction type and the instruction set architecture.

Finding Average CPI

- Computing the overall effective CPI is done by looking at the different types of instructions and their individual cycle counts and averaging:

$$\text{Overall effective CPI} = \sum_{i=1}^{n} (CPI_i \times IC_i)$$

  - Where IC_i is the count (percentage) of the number of instructions of class i executed.
  - CPI_i is the (average) number of clock cycles per instruction for that instruction class.
  - n is the number of instruction classes.

Optimizing Example

Op	Freq	CPI_i	Freq x CPI_i			
ALU	50%	1	.5	.5	.5	.25
Load	20%	5	1.0	.4	1.0	1.0
Store	10%	3	.3	.3	.3	.3
Branch	20%	2	.4	.4	.2	.4
			$\sum =$ 2.2	1.6	2.0	1.95

- How much faster would the machine be if a better data cache reduced the average load time to 2 cycles?

  CPU time new = 1.6 x IC x CC so 2.2/1.6 means 37.5% faster
- How does this compare with using branch prediction to shave a cycle off the branch time?

  CPU time new = 2.0 x IC x CC so 2.2/2.0 means 10% faster
- What if two ALU instructions could be executed at once?

  CPU time new = 1.95 x IC x CC so 2.2/1.95 means 12.8% faster

SPEC Benchmarking

- SPEC – System Performance Evaluation Corporation, an industry consortium that creates a collection of relevant programs.
  - The 2006 version includes 12 integer and 17 floating-point applications.
  - The SPEC rating specifies how much faster a system is, compared to a baseline machine – a system with SPEC rating of 600 is 1.5 times faster than a system with SPEC rating of 400.
- Note that this rating incorporates the behavior of all 29 programs – this may not necessarily predict performance for your favorite program.

Benchmarking Performance

- Each vendor announces a SPEC rating for their system:
  - A measure of execution time for a fixed collection of programs.
  - It is a function of a specific CPU, memory system, IO system, operating system, compiler.
  - Enables easy comparison of different systems.
- The key is coming up with a collection of relevant programs.

CINT2006 for Intel Core i7 920

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

Deriving a Single Performance Number

- How is the performance of 29 different apps compressed into a single performance number?
- SPEC uses **Geometric Mean** (GM) – the execution time of each program is multiplied and the N^{th} root is derived.
- Another popular metric is **Arithmetic Mean** (AM) – the average of each program's execution time.
- Yet another is the **Weighted Arithmetic Mean** – the execution times of some programs are weighted to balance priorities.

Amdahl's Law

- Architecture design is very bottleneck-driven – make the common case fast, do not waste resources on a component that has little impact on overall performance/power.
- Amdahl's Law states that the performance improvement through an enhancement is limited by the fraction of time the enhancement comes into play:

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

Amdahl's Law Example

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

- In a certain program, multiply instructions account for 80 seconds of the 100 second execution time. How much improvement in multiply is needed to double performance?

$$50 = \frac{80}{n} + 20$$

n = 8/3

Common Principles for Computers

- Make the common case fast.
- Principle of locality
  - The same data/code will be used again (temporal locality).
  - Nearby data/code will be used next (spatial locality).
- Energy
  - Systems use energy even when idle.
- 90/10 rule – 10% of the program accounts for 90% of the execution time.
- Amdahl's Law.

Chapter 1 Recap

- Knowledge of hardware improves software quality – compilers, OS, threaded programs, memory management.
- Important trends to follow:
  - Transistor sizing.
  - Move to multi-core.
  - Slowing rate of performance improvement.
  - Power/thermal constraints.
  - Long memory/disk latencies.
- Reasoning about performance – clock speeds, CPI, benchmark suites, performance equations.
- Next class period – MIPS architecture.