Chapter 01: Introduction

Lesson 06

Technological Trends, Measuring and Improving Performance

Objective

- **Understand Technology trends**
- **MIPS**
- **Benchmark Suites**
- **Amdahl's Law**

Technological Trends

- ICs improved in *density* (how many transistors and wires can be placed in a fixed area on a silicon chip)
- *speed* (how quickly basic logic gates and memory devices operate), and
- *area* (the physical size of the largest integrated circuit that can be fabricated)

Technological Trends

- Speed and Density improve *geometrically* rather than linearly — the increase in performance from one year to the next has been a relatively constant fraction of the previous year's performance, rather than constant absolute value

Technological Trends

- On average, the number of transistors that can be fabricated on a silicon chip increases by about 50 per cent per year, and transistor speed increases such that the delay of a basic logic gate (AND, OR, etc.) decreases by 13 per cent per year

Moore Law

- The observation that computer performance improves geometrically, not linearly, is often referred to as *Moore's Law* [Doubles in 18 months]

Example

- The amount of data that can be stored on a dynamic RAM (DRAM) memory chip has quadrupled every three years since the late 1970s, an annual growth rate of 60 per cent

Late 1990's

- Processor performance per unit energy dissipation has also improved *geometrically* rather than linearly
- This has made feasible PocketPCs and mobile devices with many features: for example, phone, text and multimedia messaging, gaming, e-mailing, commerce, and Internet access

Performance

- Performance— how quickly a given system can execute a program or programs
- Systems that execute programs in less time— said to have higher performance

Performance

- Best measure of computer performance— the execution time of the program or programs that the user wants to execute, but generally impractical to test all of the programs that will be run on a given system before deciding which computer to purchase or when making design decisions

MIPS

- Computer performance— the rate at which a given machine executed instructions
- *Millions of instructions per second* (MIPS)— Dividing the number of instructions executed in running a program by the time required to run the program

MIPS

- Fallen out-of-use
- Does not account for the fact that different systems often require different numbers of instructions to implement a given program.

CPI/IPC

- IPC= Number of instructions executed in running a program divided by the number of clock cycles required to execute the program,
- IPC = 1 /CPI
- These two metrics give the same information, and the choice of which one to use is generally made based on which of the values is greater than the number 1

CPI

- Number of clock cycles required to execute each instruction, known as *cycles per instruction*, or CPI
- The CPI of a given program on a given system calculated by dividing the number of clock cycles required to execute the program by the number of instructions executed in running the program

IPC

- For systems that can execute more than one instruction per cycle, the number of *instructions executed per cycle*, or IPC often used instead of CPI

Example

- A given program consists of a 100-instruction loop that is executed 42 times
- Calculation of system's CPI and IPC values for the program if it takes 16,000 cycles to execute the program on a given system

Calculation of system's CPI and IPC values

- The 100-instruction loop executed 42 times, so the total number of instructions executed is 100 x 42 = 4200
- 16,000 cycles to execute the program
- CPI = 16,000/4200 = 3.81
- To compute the IPC, we divide 4200 instructions by 16,000 cycles, getting an IPC of 0.26

Limitations of MIPS and CPI/IPC

- Both MIPS and CPI/IPC have significant limitations as measures of computer performance
- Benchmark suites— a third measure of computer performance
- Developed to address the limitations of MIPS and CPI/IPC

Benchmark Suite

- A set of programs that are believed to be typical of the programs that will be run on the system
- A system's score on the benchmark suite— how long it takes the system to execute all of the programs in the suite

Different Benchmark Suites

- One of the best-known benchmark suites— the SPEC suite, produced by the Standard Performance Evaluation Corporation
- A version of the SPEC suite— the SPEC CPU2000 benchmark, the third major, revision since the first SPEC benchmark suite published in 1989

Different Benchmark Suites

- First, their performance results based on total execution times, not rate of instruction execution
- Second, they average a system's performance across multiple programs to generate an estimate of its average speed

Different Benchmark Suites

- Makes a system's overall rating on a benchmark suite a better indicator of its overall performance than its MIPS rating on any one program

Different Benchmark Suites

- Also, many benchmarks require manufacturers to publish their systems' results on the individual programs within the benchmark, as well as the system's overall score on the benchmark suite
- Makes it possible to do a direct comparison of individual benchmark results if we know that a system will be used for a particular application

Geometric-Mean — Advantage and Disadvantage

- Inclusion of one extreme value in the series had a much greater effect on the arithmetic mean than on the geometric mean

Most important rule for designing high-performance computer systems

- ***Make the common case fast***
- The impact of a given performance improvement on overall performance dependent on both how much the improvement improves performance when it is in use and how often the improvement is in use

Amdahl’s Law

- Quantitatively, this rule has been expressed as *Amdahl s Law*

- Execution $Time_{new}$ = Execution $Time_{old}$ $\times$ [$Frac_{unused}$ + $Frac_{used}$ $\div$ $Speedup_{used}$]

Amdahl's Law

- Speedup $= \text{Execution Time}_{old} \div \text{Execution Time}_{new} = 1 \div [\ \text{Frac}_{unused} + \text{Frac}_{used} \div \text{Speedup}_{used}\ ]$

Summary

We learnt

- The observation that computer performance improves geometrically, not linearly, is often referred to as *Moore's Law* [Doubles in 18 months]
- *Millions of instructions per second* (MIPS)
- Number of clock cycles required to execute each instruction
- The number of *instructions executed per cycle*

We learnt

- Better measure of performance— Benchmark Suites
- Amdahl's Law
- Most important rule for designing high-performance computer systems— Make the common case fast

End of Lesson 06

Technological Trends, Measuring and Improving Performance