Concepts Introduced in Chapter 2

- basic computer performance measures
- only valid measures of computer performance are based on time
- examine other popular measures of performance
Comparing the performance between systems involves the performance of:
- architectures
- implementations of the same architecture
- compilers for the same architecture and implementation
- operating systems
Performance Terms

- response time (execution time) - time between the start and completion of an event
- throughput - the total amount of work done in a given time
- CPU time - does not include time for I/O or running other programs
  - user CPU time - only CPU time spent in that program (executable)
  - system CPU time - CPU time spent in the O.S. performing tasks requested by the program
Performance Equations

- Performance has an inverse relationship to time.

\[ \text{Performance} = \frac{1}{\text{Execution Time}} \]

- Comparing the performance of two machines can be accomplished by comparing execution times.

\[ \text{Performance}_x > \text{Performance}_y \]

\[ \frac{1}{\text{Execution Time}_x} > \frac{1}{\text{Execution Time}_y} \]

\[ \text{Execution Time}_y > \text{Execution Time}_x \]
N Times Faster

- One often may state that one machine is \( n \) times faster than another machine. What does this mean?

\[
\frac{Performance_X}{Performance_Y} = n = \frac{Execution\_Time_Y}{Execution\_Time_X}
\]
Measures of Clock Speed

- **Clock Periods**
  - millisecond (ms) - one thousandth of a second ($10^{-3}$)
  - microsecond (μs) - one millionth of a second ($10^{-6}$)
  - nanosecond (ns) - one billionth of a second ($10^{-9}$)
  - picosecond (ps) - one trillionth of a second ($10^{-12}$)

- **Clock Rates**
  - hertz (Hz) - one cycle per second
  - kilohertz (kHz) - one thousand cycles per second ($10^3$)
  - megahertz (MHz) - one million cycles per second ($10^6$)
  - gigahertz (GHz) - one billion cycles per second ($10^9$)
  - terahertz (THz) - one trillion cycles per second ($10^{12}$)
Measures of Size of Data

- Bit - one binary digit
- Nibble - four binary digits
- Byte - eight bits
- Word - typically four bytes
- Kilobyte (Kb) - $2^{10}$ (1024) bytes (cache)
- Megabyte (Mb) - $2^{20}$ bytes (memory)
- Gigabyte (Gb) - $2^{30}$ bytes (disk)
- Terabyte (Tb) - $2^{40}$ bytes (tape libraries)
CPU Time

CPU Time

- CPU time ignores I/O and time for executing other processes.

\[
\text{CPU} \_\text{Time} = \frac{\text{CPU} \_\text{Clock} \_\text{Cycles} \times \text{Clock} \_\text{Cycle} \_\text{Time}}{\text{Clock} \_\text{Rate}}
\]

- We can look up the clock cycle time in a manual. How can we calculate the number of clock cycles? (Note CPI stands for clocks per instruction.)

\[
\frac{\text{CPU} \_\text{Clock} \_\text{Cycles}}{\text{Inst} \_\text{Count} \times \text{CPI}} = \sum_{i=1}^{i=n} (\text{CPI}_i \times C_i)
\]

\[
\text{CPU} \_\text{Time} = \text{Inst} \_\text{Count} \times \text{CPI} \times \text{Clock} \_\text{Cycle} \_\text{Time}
\]
Amdahl's Law

- Amdahl's Law states that the performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used.

- Amdahl's Law depends on two factors:
  - The fraction of the time that the enhancement can be exploited.
  - The improvement gained by the enhancement mode, as if the enhanced mode were used for the entire execution.

- New execution time can be calculated as:

\[
old\text{-}\text{execution\ time} \times (1 - \frac{Fraction\_enhanced}{Speedup\_enhanced})
\]

- Speedup overall can be calculated as:

\[
\frac{1}{1 - \frac{Fraction\_enhanced}{Speedup\_enhanced}}
\]
Other Popular Measures

- **MIPS** - millions of instructions per second
  \[
  MIPS = \frac{Inst\_Count}{Execution\_Time \times 10^6}
  \]

- **MFLOPS** - millions of floating-point operations per second
  \[
  MFLOPS = \frac{FP\_Operations}{Execution\_Time \times 10^6}
  \]

- **Problems**
  - Instruction sets differ between machines. Some architectures have instructions that do more work than instructions on other machines.
  - MIPS and MFLOPS ratings vary between programs on the same machine.
  - These measures can vary inversely with performance.
Benchmarks

- Benchmarks are programs chosen to measure the performance of a computer system
  - Synthetic benchmarks are artificial programs constructed to try to match the characteristics of a large set of programs.
    - Dhrystone
    - Whetstone
  - Kernel benchmarks are small, time-intensive pieces from real programs that are extracted and then used as benchmarks.
    - Livermore Loops
    - Linpack
  - SPEC (System Performance Evaluation Cooperative) Benchmarks are more realistic.
  - The applications that will actually be used would be the best benchmarks.
## Figure 2.6: The SPEC95 CPU benchmarks.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>AI; plays the game of Go</td>
</tr>
<tr>
<td>m88kswim</td>
<td>Motorola 88K chip simulator</td>
</tr>
<tr>
<td>gcc</td>
<td>GNU C compiler; generating SPARC code</td>
</tr>
<tr>
<td>compress</td>
<td>Compresses and decompresses file in memory</td>
</tr>
<tr>
<td>lj</td>
<td>Lisp interpreter</td>
</tr>
<tr>
<td>jpeg</td>
<td>Graphic compression and decompression</td>
</tr>
<tr>
<td>perl</td>
<td>Manipulates strings and prime numbers in Perl</td>
</tr>
<tr>
<td>vortex</td>
<td>A database program</td>
</tr>
<tr>
<td>tomcat</td>
<td>A mesh generation program</td>
</tr>
<tr>
<td>swim</td>
<td>Shallow water model with 513 x 513 grid</td>
</tr>
<tr>
<td>ss2cor</td>
<td>Quantum physics; Monte Carlo simulation</td>
</tr>
<tr>
<td>hydco2d</td>
<td>Astrophysics; Hydrodynamic Navier-Stokes equations</td>
</tr>
<tr>
<td>mgrid</td>
<td>Multigrid solver in 3-D potential field</td>
</tr>
<tr>
<td>applc</td>
<td>Parabolic/elliptic partial differential equations</td>
</tr>
<tr>
<td>web3d</td>
<td>Simulates isotropic, homogeneous turbulence in a cube</td>
</tr>
<tr>
<td>aaps</td>
<td>Temperature, wind velocity, and distribution of pollutants</td>
</tr>
<tr>
<td>fpppp</td>
<td>Quantum chemistry</td>
</tr>
<tr>
<td>wave5</td>
<td>Plasma physics; electromagnetic particle simulation</td>
</tr>
</tbody>
</table>
Fallacies and Pitfalls

- Pitfall: Expecting the improvement of one aspect of a machine to increase performance by an amount proportional to the size of the improvement.
- Fallacy: Hardware-independent metrics predict performance.
- Pitfall: Comparing computers using only one or two of three performance metrics: clock rate, CPI, and instruction count.
- Pitfall: Using peak performance to compare machines.