Performance Overview

- Execution time is the best measure of performance: simple, intuitive, straightforward.
- Benchmarks
- Metrics for summarizing performance data
- Pitfalls
- Two important quantitative methods:
  - Amdahl’s Law and Speedup
  - CPI - cycles per instruction

Time-based Metrics

<table>
<thead>
<tr>
<th>Plane</th>
<th>DC to Paris</th>
<th>Speed</th>
<th>Passengers</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing 747</td>
<td>6.5 hours</td>
<td>610 mph</td>
<td>470</td>
<td>286,700</td>
</tr>
<tr>
<td>Concorde</td>
<td>3 hours</td>
<td>1350 mph</td>
<td>132</td>
<td>178,200</td>
</tr>
</tbody>
</table>

- Elapsed time to run the task
  - Execution time, response time, latency
- Rate: tasks completed per day, hour, week, sec, ns
  - Throughput, bandwidth
Performance comparisons

"X is n times faster than Y" means

\[
\frac{\text{ExTime}(Y)}{\text{ExTime}(X)} = \frac{\text{Performance}(X)}{\text{Performance}(Y)} = n
\]

Execution time metric

wall clock time = response time = elapsed time

CPU time = user CPU time + system CPU time

We will measure CPU performance using user CPU time on an unloaded system

Example: Unix time command

90.7u 12.9s 2:39 65%
Programs to evaluate performance

- Real programs
- Kernels
- Toy benchmarks
- Synthetic benchmarks

SPEC Benchmark Suite
(Standard Performance Evaluation Cooperative
http://www.spec.org)

- **First Round 1989**
  - 10 programs yielding a single number ("SPECmarks")

- **Second Round 1992**
  - SPECInt92 (6 integer programs) and SPECfp92 (14 floating point programs).
  - Compiler Flags unlimited
  - Normalized to VAX-11/780
**SPEC Benchmark Suite - continued**

- **Third Round 1995** “benchmarks useful for 3 years”
  - SPECint95 (8 integer programs) and SPECfp95 (10 floating point)
  - Single flag setting for all programs: SPECint_base95, SPECfp_base95

- **Fourth Round 2000**
  - CINT2000 (12 integer programs) and CFP2000 (14 floating point)
  - usable across Unix and Windows NT
  - Normalized to Sun Ultra5_10 workstation with 300-MHz SPARC and 256 MB memory
  - base and optimized versions

**Newer desktop (PC) benchmarks**

- **Winbench**
  - scripts that test CPU, video, and disk performance

- **Business Winstone**
  - netscape, office suite applications

- **CC Winstone**
  - content creation applications such as Photoshop
Specialized Benchmarks

- **Graphics benchmarks**
  SPECviewperf, SPECapc

- **Embedded systems benchmarks**
  EEMBC - automotive, consumer, networking, office automation, telecommunications

- **Server benchmarks**
  SPEC benchmarks for CPU, files system, web server, transaction servers (www.tpc.org)

How to Summarize Performance

Two metrics for summarizing execution time

- **Arithmetic mean** (weighted arithmetic mean) tracks execution time: \( \Sigma(T_i)/n \) or \( \Sigma(W_i*T_i) \)

- **Harmonic mean** (weighted harmonic mean) of rates (e.g., MFLOPS) tracks execution time: \( n/\Sigma(1/R_i) \) or \( n/\Sigma(W_i/R_i) \)
How to Summarize Performance

- Normalized execution time for scaling performance (e.g., X times faster than SPARCstation 10)
  - Arithmetic mean impacted by choice of reference machine

- Geometric mean for comparison of normalized execution times
  \[ \prod (T_i)^{1/n} \]
  - Independent of chosen machine
  - but not good metric for total execution time

Fallacies and Pitfalls

- Fallacy: MIPS is an accurate measure of comparative performance.
  
  \[ \text{MIPS} = \frac{\text{instruction count}}{\text{execution time} \times 10^6} = \frac{\text{clock rate}}{\text{CPI} \times 10^6} \]

- Fallacy: MFLOPS is a consistent and useful measure of performance.
  
  \[ \text{MFLOPS} = \frac{\text{floating point operations}}{\text{execution time} \times 10^6} \]
More Fallacies and Pitfalls

- **Fallacy**: Synthetic benchmarks predict performance for real programs.
- **Fallacy**: Benchmarks remain valid indefinitely.
- **Fallacy**: Peak performance tracks observed performance.
- **Fallacy**: Your performance in CIS 429 depends on how much you eat in class.