Last Class …

- Loop fusion
- Loop unrolling
- Gcc compiler optimizations
- List scheduling algorithm

Performance Metrics

- Throughput
- Execution time
- Instruction count
- Instruction rate (MIPS/GIPS)
- Cycles per instruction (CPI)
- Clock cycle time
- Clock rate – inverse of clock cycle time

Metrics of Performance

- Application
- Programming Language
- Compiler
  - ISA
  - Datapath
    - Control
    - Functional Units
      - Transistors
      - Wires
      - Pins

- Answers per month
- Operations per second
- Execution time
- Instructions or floating point
- Operations per second (MIPS/GIPS)
- Cycles per instruction (CPI)
- Megabytes per second
- Clock rate (cycles per second)

Relating Performance Metrics

- CPU Time = Instructions X Average CPI X Cycle time

<table>
<thead>
<tr>
<th></th>
<th>Instr. Count</th>
<th>Avg. CPI</th>
<th>Clock rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compiler</td>
<td>x</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ISA</td>
<td>x</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
How To Evaluate Performance

- Benchmarking
  - Evaluate different systems or changes to a single system
  - Represent a large class of important programs so as to provide a target to improve performance
  - Choice of benchmarks important

Programs to Evaluate Processor Performance

- Microbenchmarks – stress test different aspects of the system
- Synthetic benchmarks – attempt to match average instruction mixes of real workloads, e.g., Whetston, Dhrystone
- Kernels – time critical excerpts of programs, e.g., Livermore loops
- Real programs – e.g., gcc, spice
  - See [www.spec.org](http://www.spec.org) (Standard Performance Evaluation Corporation)
    - Emphasizes speedup, throughput, reproducibility of results
    - Constantly updated (SPEC89, SPEC92, SPEC95, SPEC2000, SPEC2006, …)

Averaging/Summarizing Performance

- Arithmetic (or weighted arithmetic) mean of time-based metric tracks execution time
- Harmonic mean (or weighted harmonic mean) of rates (e.g., instructions per second) tracks execution time
- Normalized execution time is handy for scaling performance (e.g., time on reference machine/time on measured machine)
  - But, use geometric mean to average normalized execution time

Amdahl’s Law

- Speedup limited by the part of the program that has not been optimized/enhanced
- Suppose enhancement E accelerates a fraction F of the task by a factor S and the remainder of the task is unaffected
  - $\text{ExTime}(\text{with E}) = ((1-F)+F/S)\times\text{ExTime}(\text{without E})$
  - $\text{Speedup}(\text{with E}) = \frac{\text{ExTime}(\text{without E})}{\text{ExTime}(\text{with E})}$