Roadmap

- Performance Equation
- Quantifying Power
- Amdahl’s Law
Performance Equation

- Basic performance equation

$$\text{CPU Execution Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle}$$

$$\text{CPU Execution Time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- These factors separate three factors that affect performance
  - Can measure CPU execution time (run the program)
  - Clock rate is given
  - Can measure the overall instruction count using profilers/simulators
  - CPI varies by the instruction type and implementation details

The Power Wall

FIGURE 1.15 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip.

Copyright © 2009 Elsevier, Inc. All rights reserved.
### Power Dissipation Challenge

![Power Dissipation Chart](chart.png)

**Power delivery and dissipation will be prohibitive**

---

### Future Microprocessor Design Trends

![Future Microprocessor Design Trends Chart](chart2.png)

**Power not a Consideration**

**Optimize Performance then Optimize Power**

**Set Power Envelope then Optimize Performance**

---

*Courtesy, Intel®*
Quantify Power

- For CMOS chips, traditional dominant energy consumption has been in switching transistors, called dynamic power.

$$\text{Power}_{\text{Dynamic}} = \text{Capacitive Load} \times \text{Voltage}^2 \times \text{Frequency}$$

$$\text{Energy}_{\text{Dynamic}} = \text{Capacitive Load} \times \text{Voltage}^2$$

- For fixed task, slowing clock rate (frequency switched) reduces power, but not energy.

---

Quantify Power

- Capacitive load is a function of the number of transistors connected to the output and technology.
  - Determines capacitance of wires and transistors.
- Dropping voltage helps both, so went from 5 V to 1 V.
- To save energy and dynamic power, most CPUs now turn off the clock of inactive modules (E.g.: Floating Point Unit).

- See the handout on “IBM PowerPC7.”
Multi-Cores, Clock Rate and Power

- Sea Change: From uniprocessor designs to multicores!

<table>
<thead>
<tr>
<th>Product</th>
<th>AMD Opteron X4 (Bar celon a)</th>
<th>Intel Neha lem</th>
<th>IBM Power 6</th>
<th>Sun Ultra SPARC T2 (Ni aga ra 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores per chip</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Clock rate</td>
<td>2.5 GHz</td>
<td>~2.5 GHz</td>
<td>4.7 GHz</td>
<td>1.4 GHz</td>
</tr>
<tr>
<td>Microprocessor power</td>
<td>120 W</td>
<td>~100 W</td>
<td>~100 W</td>
<td>94 W</td>
</tr>
</tbody>
</table>

Power at the Enterprise Level

- Power consumption is a significant problem at the enterprise/server level
- Mismatch between rated power and actual utilization
- See “Power Provisioning for a Warehouse-sized Computer” by X. Fan et.al in ISCA 2007

<table>
<thead>
<tr>
<th>Component</th>
<th>Peak Power</th>
<th>Count</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>40 W</td>
<td>2</td>
<td>80 W</td>
</tr>
<tr>
<td>Memory</td>
<td>9 W</td>
<td>4</td>
<td>36 W</td>
</tr>
<tr>
<td>Disk</td>
<td>12 W</td>
<td>1</td>
<td>12 W</td>
</tr>
<tr>
<td>PCI Slots</td>
<td>25 W</td>
<td>2</td>
<td>50 W</td>
</tr>
<tr>
<td>Motherboard</td>
<td>25 W</td>
<td>1</td>
<td>25 W</td>
</tr>
<tr>
<td>Fan</td>
<td>10 W</td>
<td>1</td>
<td>10 W</td>
</tr>
<tr>
<td>System Total</td>
<td></td>
<td></td>
<td>213 W</td>
</tr>
</tbody>
</table>

- Several techniques have been proposed – energy proportional computing, power napping, DVFS (dynamic voltage and frequency scaling)
Amdahl's Law

- This 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.
- The law defines the performance factor as the speedup: it tells us how much faster a task will run using the machine with the enhancement as opposed to the original machine.
- It gives us the overall performance behavior of a computer with two different modes and speeds of operation.
- A serial machine (one CPU) vs a parallel version, multiple CPUs, etc.

Speedup due to enhancement E:

\[
\text{Speedup}(E) = \frac{\text{Performance w/o E}}{\text{Performance w/ E}} = \frac{\text{ExTime w/o E}}{\text{ExTime w/ E}}
\]

Suppose that enhancement E accelerates a fraction F of the task by a factor S, and the remainder of the task is unaffected.
Amdahl's Law

- It depends on two factors: \( \text{Fraction}_{\text{enhanced}} \) and \( \text{Speedup}_{\text{enhanced}} \)
- \( \text{Fraction}_{\text{enhanced}} \): fraction of the computation time in the original program that can be converted to take advantage of enhancement.
- \( \text{Speedup}_{\text{enhanced}} \): the amount of improvement.

\[
\text{ExTime}_{\text{new}} = \frac{\text{ExTime}_{\text{old}} \times (1 - \text{Fraction}_{\text{enhanced}})}{\text{Speedup}_{\text{enhanced}}} + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}
\]

\[
\text{Speedup}_{\text{overall}} = \frac{\text{ExTime}_{\text{old}}}{\text{ExTime}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}
\]

Amdahl’s Law: Example 1

- Floating point instructions improved to run 2X; but only 10% of actual instructions are FP

\[
\text{ExTime}_{\text{new}} =
\]

\[
\text{Speedup}_{\text{overall}} =
\]
Amdahl’s Law: Example 1

- Floating point instructions improved to run 2X; but only 10% of actual instructions are FP

\[ \text{ExTime}_{\text{new}} = \text{ExTime}_{\text{old}} \times (0.9 + 0.1/2) = 0.95 \times \text{ExTime}_{\text{old}} \]

\[ \text{Speedup}_{\text{overall}} = \frac{1}{0.95} = 1.053 \]

Amdahl’s Law: Example 2

- An enhancement that runs 10 times faster than the original machine but only for 40% of the time:

\[ \text{Fraction}_{\text{enhanced}} = 0.4 \]
\[ \text{Speedup}_{\text{enhanced}} = 10 \]
\[ \text{Speedup}_{\text{overall}} = 1/ (0.6 + 0.4/10) = 1/0.64 = 1.65 \]

Amdahl’s law is applicable for parallel processors
Amdahl’s Law summary

- If an enhancement is only suitable for a fraction of a task, we can’t speed up the task by more than the reciprocal of 1 minus the fraction.
- This of course assumes fixed-load problems: the problem size does not change with improvements.
- Gustafson’s Law deals with fixed-time problems and scalable workloads.

MIPS (Million Instructions Per Second)

\[
\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}
\]

- MIPS rating misleading
- Execution time is the only valid and unimpeachable measure of performance.

\[
\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{clock cycle}
\]

\[
\text{seconds program} = \frac{\text{Instructions program}}{\text{Instruction program}} \times \frac{\text{Clock cycle program}}{\text{clock cycle}} \times \text{seconds program}
\]