Parallel Computing and Performance Evaluation

-- Amdahl’s Law

Yinong Chen
Chapter 7 Roadmap: Evaluation in Design Process

1. Amdahl’s Law
2. Multi-Core and HyperThreading
3. Application of Amdahl’s Law
4. Limitation of scale up
Performance Evaluation Consideration: Amdahl's Law of Speedup

Named after computer architect Gene Amdahl, born on 11/16, 1922

- What counts the most: Make the common case fast;
- Where is bottleneck of speed? Improve the bottleneck first!
- Do not ignore the uncommon cases;
- The performance enhancement of an improvement is limited by how much the improved feature is used.
- Don’t expect an enhancement proportional to how much you enhanced something.
Amdahl’s Law (contd.)

Assume we have a design (e.g., a computer or a car) which does not quite meet the performance requirement (Think about the execution speed of a computer). We need to improve the design. Assume that we redesigned the system so that \( f \) fraction of time is enhanced by a factor of \( s \).

\[
\text{Speedup} = \frac{\text{old (execution) time}}{\text{new (execution) time}} = \frac{\text{old time}}{(f \times \text{old time})/s + (1 - f) \times \text{old time}} = \frac{1}{f/s + (1 - f)}
\]
Case Study: Enhancing Performance

Five Components of a Computer

- CPU
- ALU
- Memory
- Peripheral
- Bus

90% of performance

10% of performance
Single Core and HyperThreading Processor

Single Core Processor

HyperThreading Processor:
Allows to share the unused execution units

It appears to OS to have two processors, resulting two threads are assigned to the processor. Executing two mixed threads will reduce instruction dependency for pipelined processor!
Multi-Core and HyperThreading Processor

Multi-Core

Multi-Core with HyperThreading
## Intel Multi-Core HyperThreading Processors

<table>
<thead>
<tr>
<th>Processor</th>
<th>Cache</th>
<th>Clock Speed</th>
<th>No. of Cores</th>
<th>Hyper Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>i7-980X</td>
<td>12 MB</td>
<td>3.33 GHz</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>i7-970</td>
<td>12 MB</td>
<td>3.2 GHz</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>i7-950</td>
<td>8 MB</td>
<td>3.06 GHz</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>i7-870</td>
<td>8 MB</td>
<td>2.93 GHz</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>i7-860</td>
<td>8 MB</td>
<td>2.8 GHz</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>i5-760</td>
<td>8 MB</td>
<td>2.8 GHz</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>i5-750</td>
<td>8 MB</td>
<td>2.66 GHz</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>i5-680</td>
<td>4 MB</td>
<td>3.6 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i5-670</td>
<td>4 MB</td>
<td>3.46 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i5-660</td>
<td>4 MB</td>
<td>3.33 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i5-650</td>
<td>4 MB</td>
<td>3.2 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i3-550</td>
<td>4 MB</td>
<td>3.2 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i3-540</td>
<td>4 MB</td>
<td>3.06 GHz</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>i3-540</td>
<td>4 MB</td>
<td>2.93 GHz</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Improve CPU Performance

Bus

Multi-Core

ALU

control

Peripheral

input

output
The engineers are able to make $f = 90\%$ of the system much faster. If

$s = 10$ times faster,
$s = 100$ times faster, and
$s = 1000$ times faster

What is the overall speed up, respectively?

9 times, 90 times, 900 times, respectively?

What is your answer?
Revolutionary New Designs and Fallacy

\[
\text{Speedup } (f, s) = \frac{1}{f/s + (1 - f)}
\]

\[
\text{Speedup } (0.9, s) = \frac{1}{0.9/s + 0.1}
\]

\[
\text{Speedup } (0.9, s=10) = \frac{1}{0.9/10 + 0.1} = 5.3
\]

\[
\text{Speedup } (0.9, s = 100) = \frac{1}{0.9/100 + 0.1} = 9.1
\]

\[
\text{Speedup } (0.9, s = 1000) = 9.9
\]
Limitation of Speedup

\[ Speedup (f, s) = \frac{1}{f/s + (1 - f)} \]

When \( f \to 0 \) (trying to improve little or nothing of used fraction)

\[ Speedup \to 1 \] (no speedup at all)

When \( f \to 1 \) (improve the entire system 100%), \( Speedup \to s \)

When \( s \to 0 \) (no improvement), \( Speedup \to 1 \)

When \( s \to \infty \) (unlimited enhancement)

\[ \frac{1}{(1 - f)} \]

\[ Speedup \to (1 - f) \]

which is a constant, e.g., \( f = 0.9 \to Speedup = 10 \)
Why?

- What counts the most: Make the common case fast;
- Where is bottleneck of speed?
- Do not ignore the uncommon cases: They can become the common cases when the previous common case is addressed and a new bottleneck will occur;
Amdahl’s Speedup Law
Reviewed the Basic Architecture: Five Component Model
HyperThreading Architecture
Multi-Core Architecture
Multi-Core and HyperThreading Architecture
Intel core architecture series
Amdahl’s Speedup Law: Improvement is limited by different factors
Scale up and scale out