Hyperthreading and “Almost Amdahl”

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

Each of the Multiple Cores keeps its own State

1 core, 1 state

2 cores, 2 states

4 cores, 4 states

State

Core

Cache

- Registers
- Program Counter
- Stack Pointer
So, if that’s what Multicore is about, what is \textit{Hyperthreading}?

<table>
<thead>
<tr>
<th>Core States</th>
<th>1 core, 1 state</th>
<th>1 core, 2 states, with Hyperthreading</th>
<th>2 cores, 2 states</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core States</td>
<td>2 cores, 4 states, with Hyperthreading</td>
<td>4 cores, 4 states</td>
<td></td>
</tr>
</tbody>
</table>

\textbf{What is Hyperthreading and what can it Do?}

\textbf{Hyperthreading} is when a CPU chip has more states than cores.

In this case, if one thread of execution blocks (waiting for a memory fetch, for instance), then the other thread can resume execution with its state.

If we let $H$ be the fraction of a CPU’s capacity that one hyperthread can keep busy, then the remaining unused capacity is $(1-H)$. If another hyperthread can keep $H\%$ of that capacity busy, then that leaves $(1-H)(1-H)$ remaining unused capacity and so on.

If we have $n$ hyperthreads, then the final remaining unused capacity is $(1-H)^n$. The capacity actually in use would then be $1-(1-H)^n$. If one thread can only keep the CPU $H\%$ busy, then the speed-up is potentially:

$$SU = \frac{1-(1-H)^n}{H}$$
What is Hyperthreading and what can it Do?

If we have \( n \) hyperthreads, then the final remaining unused capacity is \((1-H)^n\). The capacity actually in use would then be \(1-(1-H)^n\). If one thread can only keep the CPU \( H\% \) busy, then the speed-up is potentially:

\[
SU = \frac{1-(1-H)^n}{H}
\]

A Lidar Application:
Four Cores with Two Hyperthreads per Core

Note that this is upside-down from our usual convention. Sorry. I got this from someone else.