ACMP: An Architecture to Handle Amdahl’s Law

M. Aater Suleman
Advisor: Yale Patt
HPS Research Group
Acknowledgements

Eric Sprangle, Intel
Anwar Rohillah, Intel
Anwar Ghuloum, Intel
Doug Carmean, Intel
• Single-thread performance is power constrained
• To leverage CMPs for a single application, it must be parallelized
• Many kernels cannot be parallelized completely
• Applications likely include both serial and parallel portions
• Amdahl’s law is more applicable now than ever
Serial Bottlenecks

• Inherently serial kernels
  \[ \text{For } I = 1 \text{ to } N \]

• Parallelization requires effort
CMP Architectures

• Tile small cores e.g. Sun Niagara, Intel Larrabee
  – High throughput on the parallel part
  – Low serial thread performance
  – Highest performance for completely parallelized applications

• Tile large cores e.g. Intel Core2Duo, AMD Barcelona, and IBM Power 5.
  – High serial thread performance
  – Lower throughput than Niagara
ACMP

<table>
<thead>
<tr>
<th>Niagara-like core</th>
<th>Niagara-like core</th>
<th>Niagara-like core</th>
<th>Niagara-like core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
</tr>
<tr>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
</tr>
<tr>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
<td>Niagara-like core</td>
</tr>
</tbody>
</table>

“Niagara” Approach

- Run serial thread on the large core to extract ILP
- Run parallel threads on small cores
• Run serial thread on the large core to extract ILP
• Run parallel threads on small cores
**ACMP**

- Run serial thread on the large core to extract ILP
- Run parallel threads on small cores
Performance vs. Parallelism

![Graph showing speedup vs. degree of parallelism for different processors: ACMP, Niagara, P6-Tile. The graph plots speedup on the y-axis and degree of parallelism on the x-axis. The lines show how each processor's speedup increases as the degree of parallelism increases.]
At low parallelism, ACMP and P6-Tile outperform Niagara.
At high parallelism, Niagara outperforms ACMP.
At medium parallelism, ACMP wins.
Performance vs. Parallelism

The cut-off point moves to the right in the future.
Experimental Methodology

- Large core: Out-of-order (similar to P6)
- Small Core: 2-wide, In-order
- Configuration:
  - Niagara: 16 small cores
  - P6-Tile: 4 large cores
  - ACMP: 1 Large core, 12 small cores
- Single ISA, shared memory, private L1 and L2 caches, bi-directional ring interconnect

- Simulated existing multi-threaded applications without modification
- ACMP Thread Scheduling
  - Master thread → large core
  - All additional threads → small cores
Performance Results

Speedup vs. Niagara

- Low Parallelism
  - mcf
  - is_js
  - fft_sp
  - cg_nasp

- Medium Parallelism
  - ep_nasp
  - arc_mn
  - mg_usp
  - fmm_sp
  - cholesky

- High Parallelism
  - page
  - cowq
Summary

• ACMP trades peak parallel performance for serial performance
• Improves performance for a wide range of applications
• Performance is less dependent on length of serial portion
• Improves programmer efficiency
  – Programmers can only parallelize easier-to-parallelize kernels
Future Work

• Enhanced ACMP scheduling
  – Accelerate execution of finer-grain serial portions (critical sections) using the large core
  – Requires compiler support and minimal hardware

• Improved threading decision based on run-time feedback
Thank you