The Stanford FLASH Multiprocessor


Henry Cook CS258 4/2/2008

Goals

• Support both cache-coherent shared memory and message passing
  – Not just either/or, but both at the same time

• Design a custom node controller
  – Build actual hardware

• 256 node target, scaling to thousands
  – (Actually built 64)

Big Idea

• Principal difference between CCSM and MP is protocol for transferring data
  – Overall machine structure is the same
  – Functions performed by node controller are also the same

• By making the controller a special purpose protocol processor, can leverage flexibility of software while reducing overheads

System Architecture

• Each processor is a single MIPS chip

• MAGIC has a protocol processor

• Different messages types are processed by different software handlers
Protocols - CCSM

- Directory-based, with dynamic pointer allocation structure
- Similar to DASH protocol
- Separate request-reply networks to eliminate cycles
- Handlers must yield if they cannot run to completion

Protocols - MP

- Long messages vs short messages
  - Block transfer vs synchronization
- User-level parameters passed to transfer handler running on MAGIC
- When all user message components have arrived at dest., a reception handler is invoked

Protocols - Extensions

- Just have to change the handlers
- Emulate COMA attraction memories
- Implement synchronization primitives as MAGIC handlers
- Short message support similar to active messages
  - Not user-level active messages

MAGIC Architecture

- Separation and specialization
  - Use hardwired data movement logic for speed
  - Use control logic that runs software protocols for flexibility
MAGIC Architecture

- Must operate quickly enough to avoid being the bottleneck
- Hardware based speculative message dispatch to PP
- Separate hardware for message sends

Protocol Processor

- Implements subset of DLX ISA with extensions for common protocol ops
- Statically-scheduled dual-issue superscalar processor
- No interrupts, exceptions, address translation or interlocks

Performance

- How do latencies look for local read misses?
  - Derived from Verilog model
  - Assuming PP cache hits

Performance

- PP occupancies are longest sub-operation that must be performed
- Block transfer bandwidth of 300-400MB/s
<table>
<thead>
<tr>
<th>Testbed</th>
<th>Retrospective</th>
</tr>
</thead>
<tbody>
<tr>
<td>• System-level simulator in C++</td>
<td>• Flexibility of software handlers is key</td>
</tr>
<tr>
<td>• Protocol verifier, SPLASH benchmarks</td>
<td>– Supporting multiple protocols</td>
</tr>
<tr>
<td>• Verilog description of MAGIC</td>
<td>– Scaling to arch to multiple machine sizes</td>
</tr>
<tr>
<td>• N-1 simulated nodes and one Verilog node</td>
<td>– Debugging protocol operation</td>
</tr>
<tr>
<td></td>
<td>• Building hardware in an academic environment is</td>
</tr>
<tr>
<td></td>
<td>difficult and time consuming</td>
</tr>
</tbody>
</table>