Admin

- Test script will be provided over the weekend.
- Sign up for demo slots on the ‘signup’ link on the calendar.
Consistency vs Coherence

• Coherence is micro-scale
  • Defines the behavior of a single data value.
  • All actors should have the same view of state.

• Consistency is macro-scale
  • Defines visible concurrency between data.
  • Contract for how shared resources behave.
x86 Cache

Processor Chip

- **Regs.**
- **L1 Data**
  - 1 cycle latency
  - 16 KB
  - 4-way assoc
  - Write-through
  - 32B lines
- **L1 Instruction**
  - 16 KB, 4-way
  - 32B lines

**L2 Unified**
- 128KB–2 MB
- 4-way assoc
- Write-back
- Write allocate
- 32B lines

**Main Memory**
- Up to 4GB

---

Thursday, April 18, 13
Coherence Strategies
(For write-back caches)

- Invalidation / Directory
- Snooping
Directory Coherence

- Centralized manager for cache states.
- Manager/Directory sends invalidations on behalf of client.
- SGI Origin (mid 90s) had a directory with 1024 cores, Intel i7’s use ‘Home Snoop’, a directory protocol.
There are many ways of sizing and organizing the directory, and for now we assume the simplest model: for each block in memory, there is a corresponding directory entry. In Section 8.6, we examine and compare more practical directory organization options. We also assume a monolithic LLC with a single directory controller; in Section 8.7, we explain how to distribute this functionality across multiple banks of an LLC and multiple directory controllers.

8.2.2 High-Level Protocol Specification

The baseline directory protocol has only three stable states: MSI. A block is owned by the directory controller unless the block is in a cache in state M. The directory state for each block includes the stable coherence state, the identity of the owner (if the block is in state M), and the identities of the

- FIGURE 8.1: Directory system model.
- FIGURE 8.2: Directory entry for a block in a system with N nodes.
Snooping Coherence

- When one cache is the ‘owner’ in a write-back scenario, other caches can read by snooping from the owner.

- Also allows with minimal effort an ‘exclusive-read’ state, where data is not invalidated, but

- Implemented with a hardware bus by Sun Starfire (1991), IBM POWER5 (2005)
FIGURE 7.1: MSI: Transitions between stable states at cache controller.

FIGURE 7.2: MSI: Transitions between stable states at memory controller.

FIGURE 7.3: Simple snooping system mode.
<table>
<thead>
<tr>
<th>Cycle</th>
<th>Core 1</th>
<th>Core 2</th>
<th>Memory</th>
<th>Request Bus</th>
<th>Data Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load Miss, req S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>C1 req S</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Store Miss</td>
<td></td>
<td>C1 response</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>data from mem</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Perform Load</td>
<td>req M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>C2 req M</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>C2 response</td>
<td></td>
<td>data from mem</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>data from mem</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Perform store</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Load miss, req S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>C1 req S</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>C1&amp;mem response</td>
<td></td>
<td></td>
<td>data from C2</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Perform load</td>
<td></td>
<td>store data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dangers

• Violate single-writer-multiple-reader
• Deadlock
Consistency

- Eventual Consistency
- Total Store Order
- Sequential Consistency
- Linearizable
Consistency

\[ x = 0, y = 0 \]

<table>
<thead>
<tr>
<th>Core 1</th>
<th>Core 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = 1 )</td>
<td>( y = 1 )</td>
</tr>
<tr>
<td>( r1 = y )</td>
<td>( r2 = x )</td>
</tr>
</tbody>
</table>

**Sequential:**
\( (r1, r2) = (0, 1) \) or \((1, 0)\) or \((1, 1)\)

**TSO:**
\( (r1, r2) = (0, 0) \) or \((1, 0)\) or \((0, 1)\) or \((1, 1)\)