Shared Memory SMP and Cache Coherence (cont)

Larger MPs
- Separate Memory per Processor
- Local or Remote access via memory controller
- Cache Coherency solution: non-cached pages
- Alternative: directory per cache that tracks state of every block in every cache
  - Which caches have a copies of block, dirty vs. clean, ...
  - Info per memory block vs. per cache block?
  - PLUS: In memory => simpler protocol (centralized/one location)
  - MINUS: In memory => directory is \( f(\text{memory size}) \) vs. \( f(\text{cache size}) \)
- Prevent directory as bottleneck?
  - distribute directory entries with memory, each keeping track of which Procs have copies of their blocks

Directory Protocol
- Similar to Snoopy Protocol: Three states
  - Shared: \( \geq \) 1 processor has data, memory up-to-date
  - Uncached (no processor has it; not valid in any cache)
  - Exclusive: 1 processor (owner) has data; memory out-of-date
- In addition to cache state, must track which processors have data when in the shared state (usually bit vector, 1 if processor has copy)
- Keep it simple(s):
  - Writes to non-exclusive data => write miss
  - Processor blocks until access completes
  - Assume messages received and acted upon in order sent

Review: Snoopy Cache Protocol
- Write Invalidate Protocol:
  - Multiple readers, single writer
  - Write to shared data: an invalidate is sent to all caches which snoop and invalidate any copies
  - Read Miss:
    - Write-through: memory is always up-to-date
    - Write-back: snoop in caches to find most recent copy
- Write Broadcast Protocol (typically write through):
  - Write serialization: bus serializes requests!
  - Bus is single point of arbitration
  - Good for a small number of processors; how about 16 or more?

Distributed Directory MPs

Directory Protocol
- No bus and don’t want to broadcast:
  - interconnect no longer single arbitration point
  - all messages have explicit responses
- Terms: typically 3 processors involved
  - Local node where a request originates
  - Home node where the memory location of an address resides
  - Remote node has a copy of a cache block, whether exclusive or shared
- Example messages on next slide:
  - \( P = \) processor number, \( A = \) address
**Directory Protocol Messages**

<table>
<thead>
<tr>
<th>Message type</th>
<th>Source</th>
<th>Destination</th>
<th>Msg Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read miss</td>
<td>Local cache</td>
<td>Home directory</td>
<td>P, A</td>
</tr>
<tr>
<td>Write miss</td>
<td>Local cache</td>
<td>Home directory</td>
<td>P, A</td>
</tr>
<tr>
<td>Invalidate</td>
<td>Home directory</td>
<td>Remote caches</td>
<td>A</td>
</tr>
<tr>
<td>Fetch</td>
<td>Home directory</td>
<td>Remote cache</td>
<td>A</td>
</tr>
<tr>
<td>Fetch/Invalidate</td>
<td>Home directory</td>
<td>Remote cache</td>
<td>A</td>
</tr>
<tr>
<td>Data value reply</td>
<td>Home directory</td>
<td>Local cache</td>
<td>Data</td>
</tr>
<tr>
<td>Data write-back</td>
<td>Remote cache</td>
<td>Home directory</td>
<td>A, Data</td>
</tr>
</tbody>
</table>

**State Transition Diagram for an Individual Cache Block in a Directory Based System**

- States identical to snoopy case; transactions very similar.
- Transitions caused by read misses, write misses, invalidates, data fetch requests.
- Generates read miss & write miss msg to home directory.
- Write misses that were broadcast on the bus for snooping => explicit invalidate & data fetch requests.
- Note: on a write, a cache block is bigger, so need to read the full cache block.

**CPU–Cache State Machine**

- State machine for CPU requests for each memory block.
- Invalid state if in memory.
- Fetch/Invalidate: send Data Write Back message to home directory.
- CPU read hit: CPU Write: Send Write Miss message to home directory.
- CPU write hit: send Data Write Back message and Write Miss to home directory.

**State Transition Diagram for the Directory**

- Same states & structure as the transition diagram for an individual cache.
- 2 actions: update of directory state & send msgs to satisfy requests.
- Tracks all copies of memory block.
- Also indicates an action that updates the sharing set, Sharers, as well as sending a message.