Quiz 3 Review

Ryan Lee

(Adapted from prior course offerings)

Quiz 3 logistics

- Time: 1pm EDT on Friday April 30
- Be ready to receive PDF via email 10 minutes before the start
- Zoom link: same as recitations
- Handout: released soon.

Topics

- Cache coherence
  - Snooping-based vs. Directory-based
  - VI, MSI, MESI, MOSI, …
  - Transient states
  - Synchronization primitives
- On-chip Networks
  - Topology
  - Routing
  - Flow control
  - Router micro-architecture
- Memory consistency model
  - Sequential consistency
  - Total Store Order (TSO)
  - Relaxed consistency

Cache coherence

- Simplify building shared memory systems

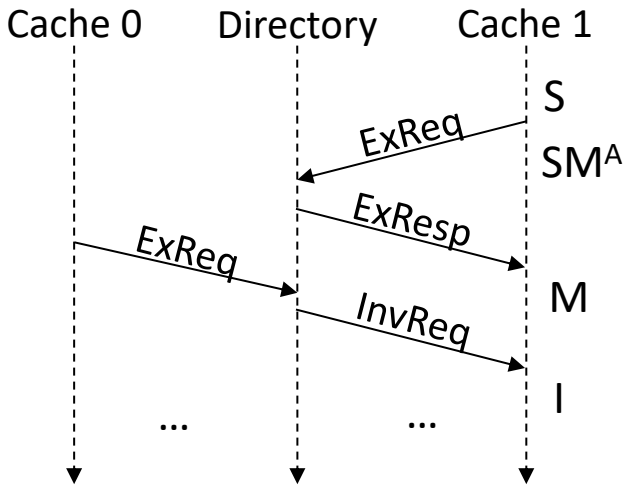
- Definition:
  - Write propagation — Liveness: do something good
    - Writes eventually become visible to all processors
  - Write serialization — Safety: don't do anything bad
    - Writes to the same location are serialized (all processors see them in the same order)

Cache coherence

- Transient states: required by lack of atomicity
  - Two types
    - Split states: to implement one transaction
      - E.g., S transitions to SM^A (instead of M), waiting for an ExResp ("A" denotes acknowledgement)
    - Race states: to handle overlaps of two transactions
      - Not all such overlaps require transient states
      - See the following examples

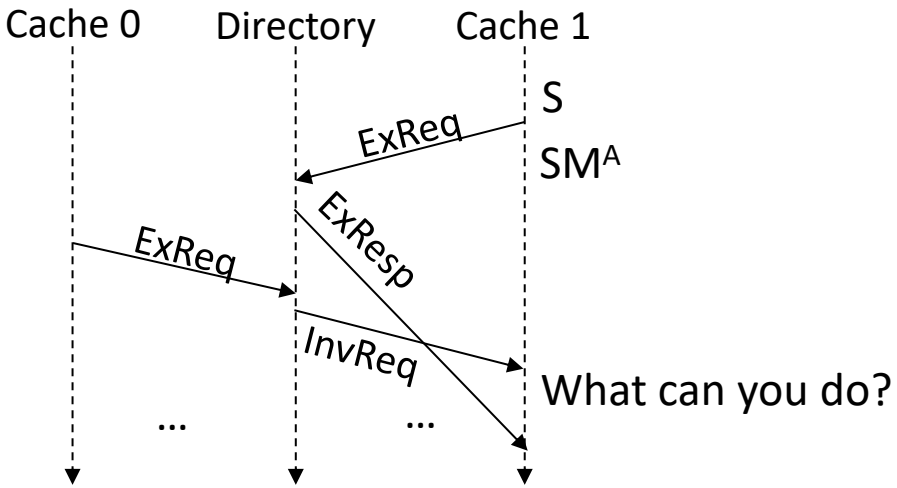
Cache coherence

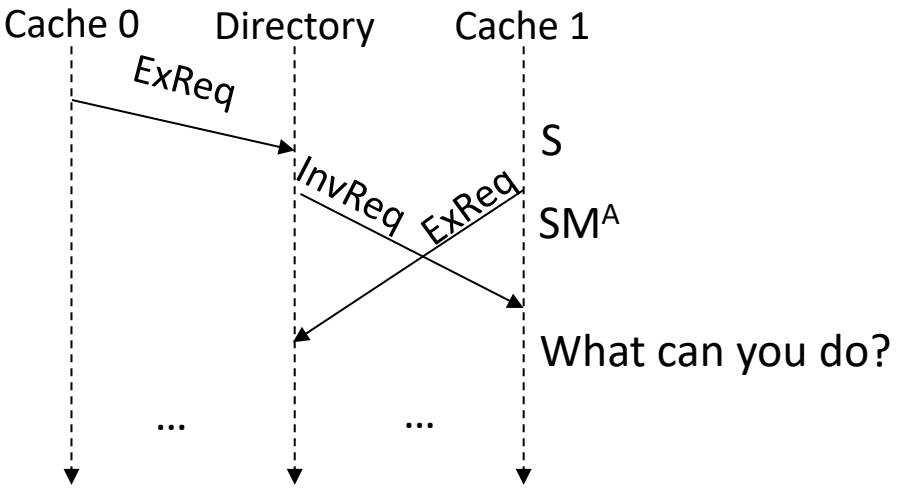
- Split example
  - SM^A



Cache coherence

- Race example





If the arriving message is from a younger transaction:

- Either defers processing it
- Or handles it immediately and transitions to a race state (e.g., SM^AI)

If the arriving message is from an elder transaction:

- Cannot defer processing it! Directory cannot accept the second ExReq because it conflicts with the first one!
- Handles it immediately and transitions to a race state (e.g., IM^D, "D" denotes data)

On-chip networks

- Allow sharing communication resource
- Topology
  - Metrics: routing distance, diameter, average distance, bisection bandwidth, …
- Routing
  - Properties: deterministic, adaptive, deadlock-free, …

On-chip networks

- Flow control
  - Bufferless
    - Circuit switching, dropping, misrouting, …
  - Buffered
    - Store-and-forward, virtual cut-through, wormhole, virtual channel
- Router architecture

Memory (consistency) model

- Concerns reads/writes to multiple memory locations
- Interacts with many parts and optimizations of the system
  - Probably more than what you would have imagined…
- Coherence is an useful (but not necessary) building block

Sequential consistency

- Definition
  - "The result of any execution is the same as if the operations of all the processors were executed in some sequential order, and the operations of each individual processor appear in the order specified by the program"
  - Arbitrary order-preserving interleaving of memory references of sequential programs

- Implementation
  - In-order instruction execution + atomic loads and stores

- Advantage: easy to understand
- Disadvantage: limits performance
  - Uniprocessor optimizations often violate them!
    - E.g., committed store buffers, non-blocking caches, speculative execution, memory address speculation, …

Total Store Order (TSO)

- Allows loads to go ahead of stores waiting in the store buffer

- Implementation
  - Sequential consistency implementation + per-core FIFO store buffer with store-load bypassing

Relaxed memory consistency

- Allows more reordering
  - Store-load
  - Store-store
  - Load-load
  - Load-store

- Re-ordering can be disabled by fences/barriers

Tips on consistency problems

- Keep definitions in mind
- Think systematically
  - E.g., For questions asking all allowed execution results: search invariants to minimize brute-force search
  - E.g., For questions asking to add minimal barriers/fences: find the precise reordering that violates the target model

Wish you all the best!