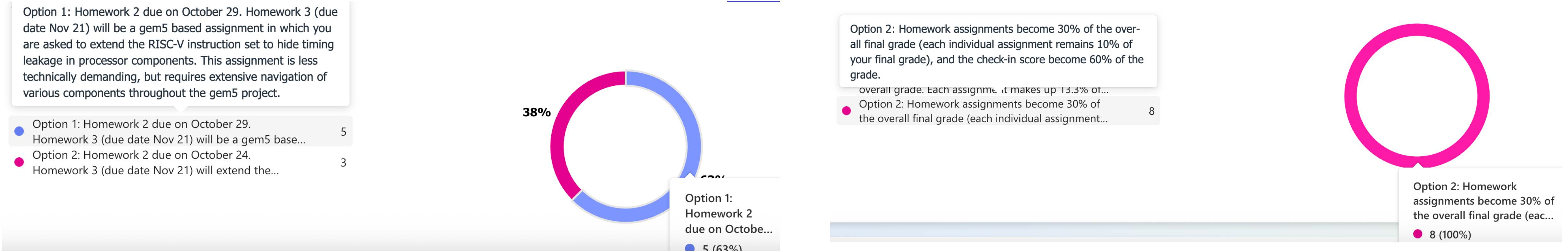


More Cache Coherence!

Check-In 4 in class;
Colloquium today! 11am
at Seaver North

Syllabus Re-Write Results



Black-box Concurrent Data Structures for NUMA Architectures

Irina Calciu
VMware Research

Siddhartha Sen
Microsoft Research

Mahesh Balakrishnan
Yale University

Marcos K. Aguilera
VMware Research

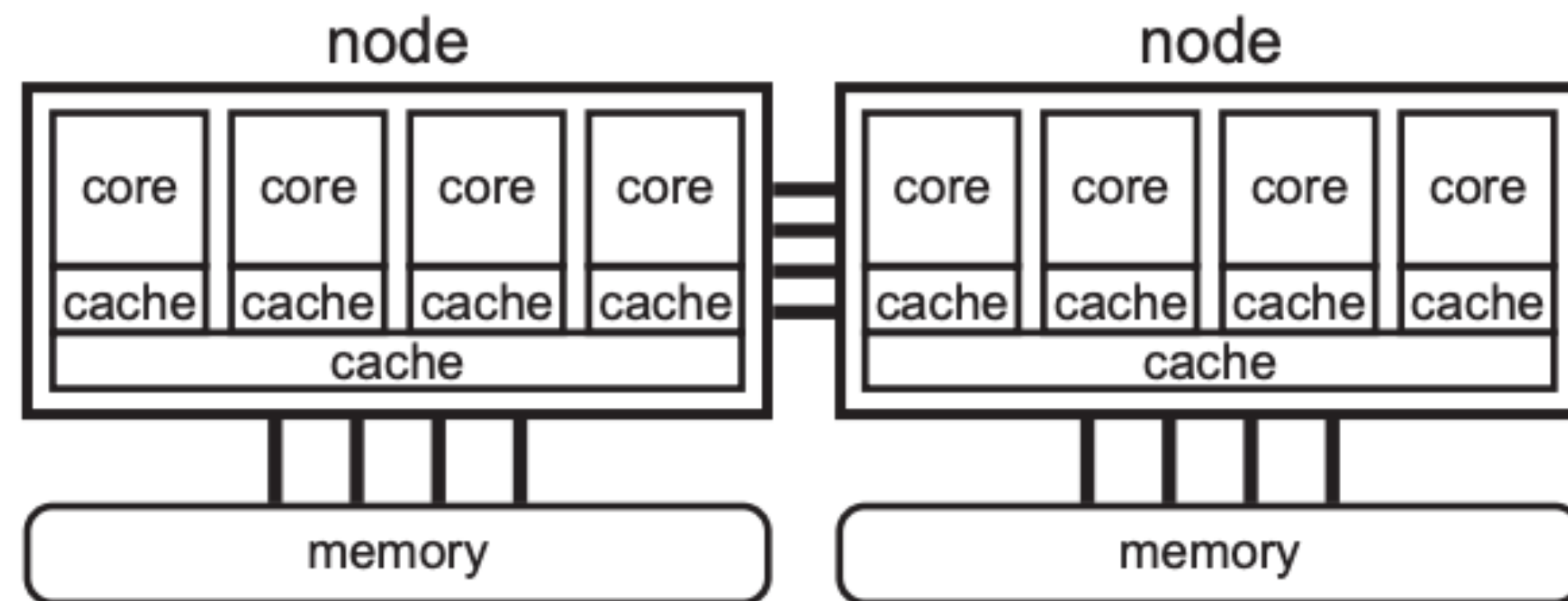


Figure 1. NUMA architecture: cores are grouped into nodes, each with its local memory. Nodes are connected by an interconnect, so that cores in one node can access the remote memory of another node, but these accesses come at a higher cost. Typically, cores have local caches, and cores on a node share a last level cache.

Figure Credit: <https://dl.acm.org/doi/pdf/10.1145/3093336.3037721>

CAPULET: Cache Pooling Metadata Caches in Secure Disaggregated Memory Systems

Abstract

Disaggregated memory systems, such as CXL, NVLink, and UALink, have emerged as pivotal technologies to address the performance limitations of memory-intensive workloads.

Given these attacks, cloud providers must deploy secure hardware so developers can reason about the safety of their deployed applications. In particular, the aforementioned memory vulnerabilities provide strong motivation for the develop-

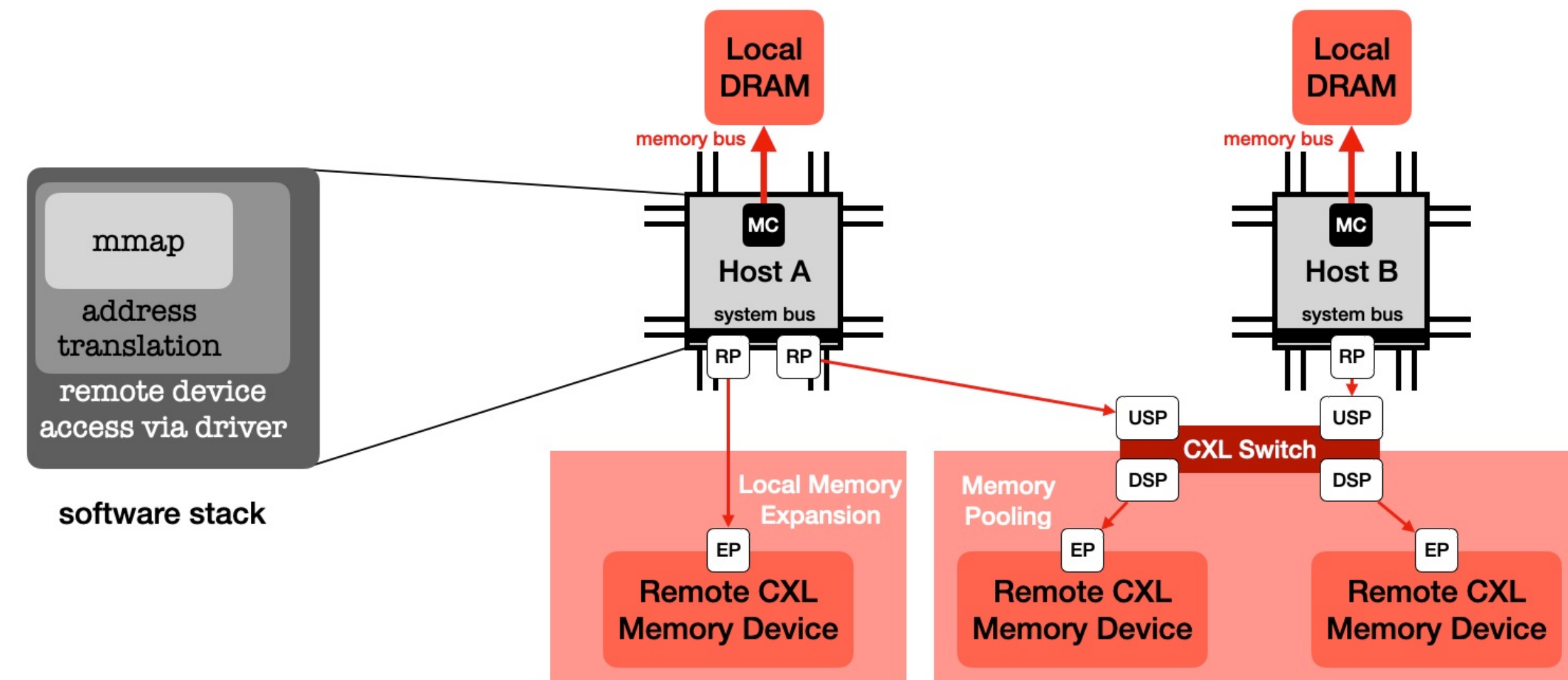


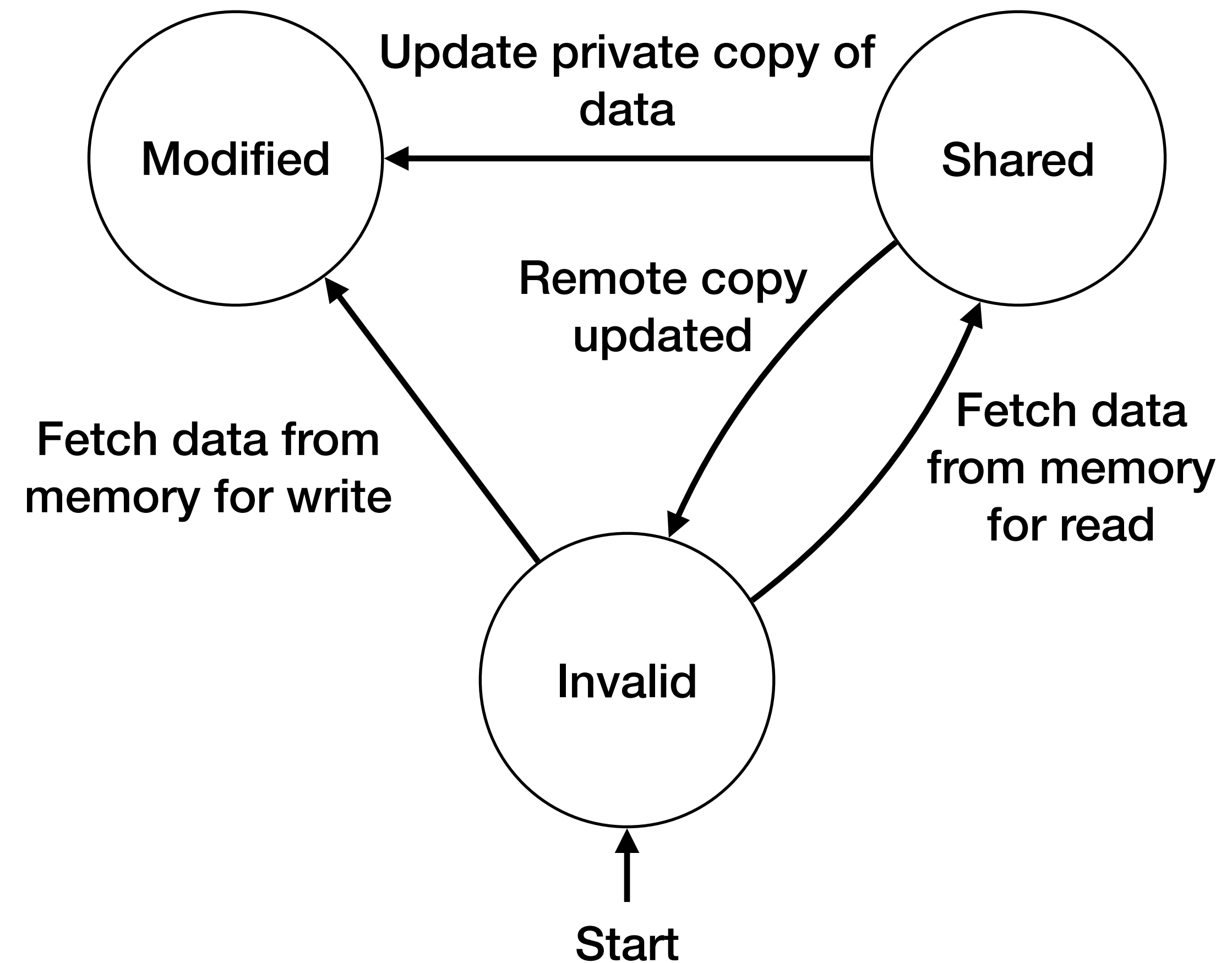
Figure Credit: Prof. Thomas' (under submission) research

Outline

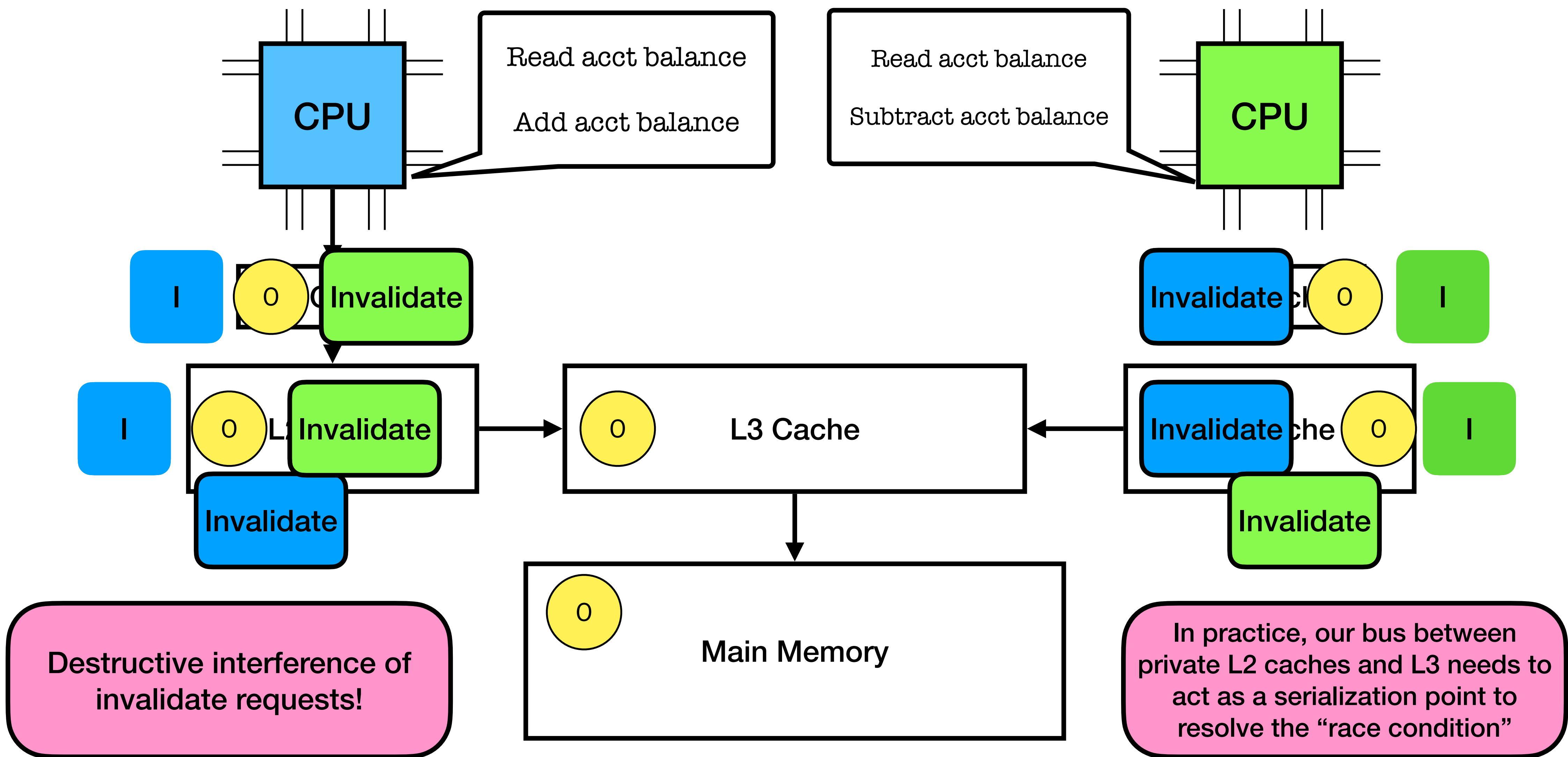
- Continuing discussion of MSI coherence
- Introducing MESI coherence, MOESI coherence
- Check-In 4

Implementing MSI Cache Coherence

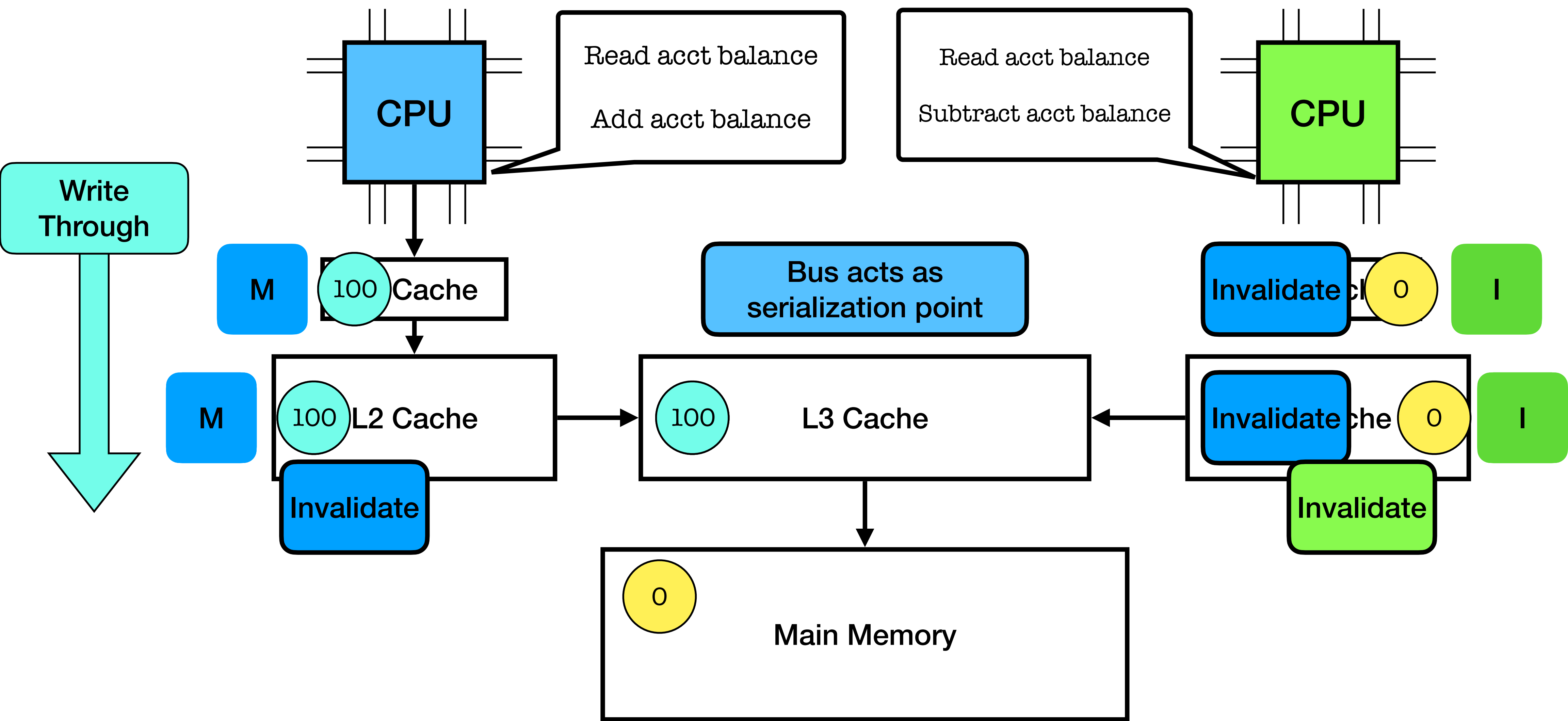
- MSI coherence: at most one processor can own a cache block in *modified* state
- To update a block, first send an *invalidate* to other caches in the memory system for this address
- Once the invalidate has responded, the updating cache can safely set the cache block state from *shared* to *modified*



MSI Coherence



MSI Coherence



MESI Coherence (the Illinois Coherence Protocol, used in Intel i7)

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 6, NO. 8, AUGUST 1995

A New Approach for the Verification of Cache Coherence Protocols

Fong Pong, *Member, IEEE*, and Michel Dubois, *Senior Member, IEEE Computer Society*

- Insight: if all readable states are assumed to be shared, then any updates to unmodified states require invalidation broadcasts ✗
- The Illinois Protocol adds an *exclusive* state to the MSI protocol to indicate whether or not the data is privately maintained ➡ transition from *exclusive* to *modified* does not require broadcast
- All cache misses require broadcasting a snoop to private caches to move cached state from *exclusive* to *shared*
- If misses are infrequent, then this will not be a significant penalty!

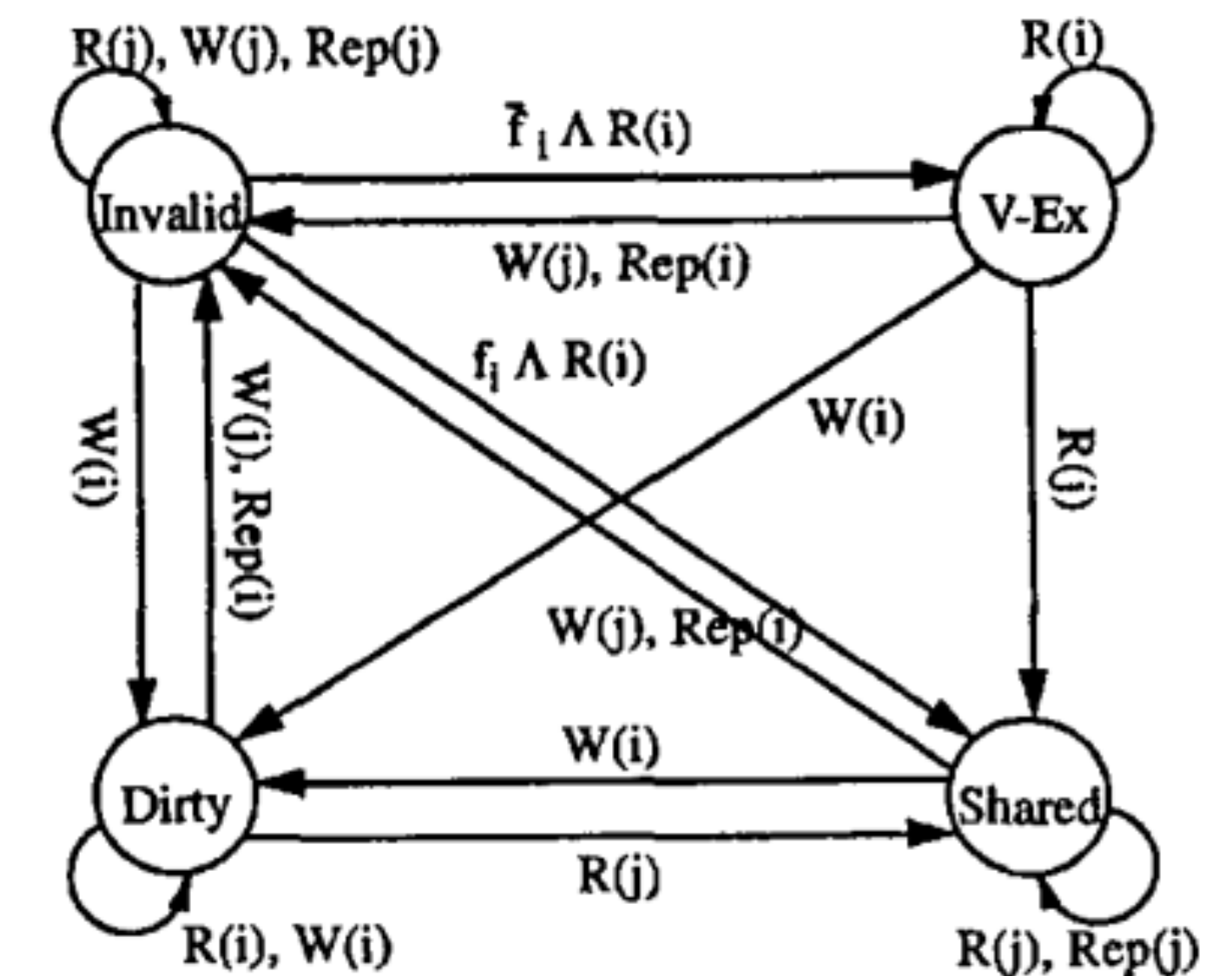


Fig. 1. The Illinois protocol transition diagram from the perspective of cache C_i .

Image Credit: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=406955>

MOESI Coherence (Used in AMD64!)

- Insight: write through caches are expensive! Even in an MESI cache, data updates need to be propagated downwards through the memory hierarchy to ensure that subsequent reads get the right value
- The *MOESI* coherence protocol extends MESI to also include an *owned* state where cache blocks may be updated without writing through
- On a cache miss, any cache that “owns” the missed packet must respond to the broadcasted snoop with the data ➡ it *cannot* ignore or drop the snoops even if the cache is under high strain!
- If the miss needs to be broadcast anyways to transition from *exclusive* to *shared* in MESI, then use the broadcasted snoop to also communicate the data!

Coherence Takeaways

- We use coherence to reason about the consistency of data across the memory system when multiple processors share data
- Two notable disadvantages of MSI are:
 - **1** every update requires notifying the rest of the memory hierarchy by sending an invalidate request... this is a lot of traffic!
 - **2** all data is pessimistically assumed to be potentially shared across the memory system
- More context about what else is happening in the cache hierarchy means that more operations can safely be performed on caches across the memory system