A Primer on Memory Consistency and Cache Coherence (Chapter 1-2)

by Yunqi Zhang
04/12/2013

1. Consistency
    a. define shared memory correctness
2. Coherence
    a. visible to software
    b. ensures that the caches never enable new or different functional behavior
3. Coherence Basics
    a. baseline system model: all memory shared
    b. incoherence: each core has its own caches
4. SWMR model
    a. *single-writer-multiple-reader* invariant: For any given memory location, at any given moment in time, there is either a single core that may write, or some number of cores that may read it.
5. Granularity of Coherence
    a. cache block in practice
    b. finer and coarser granularities in some protocols
6. Other definitions
    a. Sequential consistency
        i. The system must appear to execute all threads' loads and stores to all memory locations in a total order that respects the program order of each thread.
    b. K. Gharachorloo's
        i. Every store is eventually made visible to all cores
        ii. Writes to the same memory location are serialized
    c. Hennessy and Patterson
        i. A load to memory location A by a core obtains the value of the previous store to A by that core, unless another core has stored to A in between
        ii. A load to A obtains the value of a store S to A by another core if S and the load "are sufficiently separated in time" and if no other store occurred between S and the load
        iii. Stores to the same memory location are serialized