

What is a consistency model?

- Describes which possible re-orderings (from a single processor) and interleavings (across processors) of memory operations are valid from the perspective of each processor.

Why do you care?

- Provide guarantees to the programmer.
- Some determinism in execution
- Hard to write lock free data-structures w/o solid understanding of consistency model
- Lock / Unlock, primitives when correctly coded, use fence operations correctly
- It was thought that less strict consistency leads to higher performance

What is sequential consistency and why do you want it or not?

- Most intuitive model to reason about (for some) least re-orderings
 - All memory operations are observed to be in the same order from all processors
 - The global memory order is an interleaving of the individual thread orderings
 - A load *always* gets the value from the most recent store in the *global* ordering

What is processor consistency?

- A processor will observe the stores from another processor in the order they occurred.

Wr A, 1
Wr B 2

Wr A, 3
Wr B 4

Re A
Re B

Re A
Re B

Wr A, 1
Wr A, 3
Re A
Wr B 2
Wr B 4
Re B

Wr A, 1
Wr B 2
Wr A, 3
Wr B 4
Re A
Re B

Wr A, 1
Re A
Wr B 2
Wr A, 3
Wr B 4
Re B

Where does reality bite consistency?

- Store, @2, 0x12345678 =>
 - store @2, 0x1234
 - store @4, 0x5678
- Store, @2, 0xdeadbeef
 - store @2, 0xdead
 - store @4, 0xbeef
- LOCK MOV @2, 0x12345678