Synchronization Mechanisms for Shared-Memory Multiprocessors

Alain Kägi, Doug Bürger, James R. Göödman

24th International Symposium on Computer Architecture

Tuesday, June 3rd, 1997

galileo@cs.wisc.edu

Computer Sciences Department

University of Wisconsin-Madison

Efficient Synchronization: Let Them Eat QOLB

Alain Kägi, Doug Bürger, James Göödman

24th International Symposium on Computer Architecture

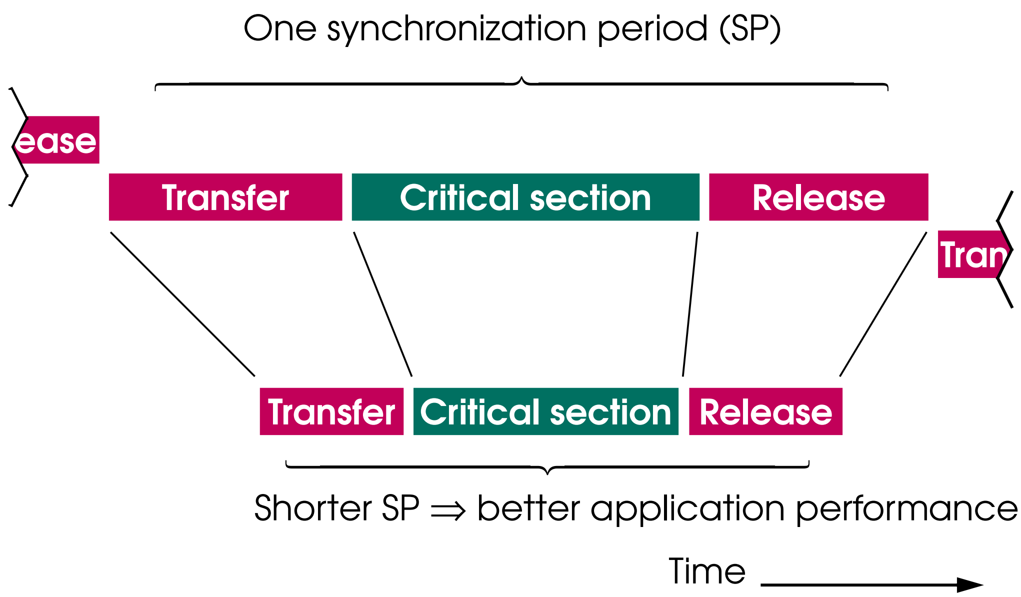
Tuesday, June 3rd, 1997

galileo@cs.wisc.edu

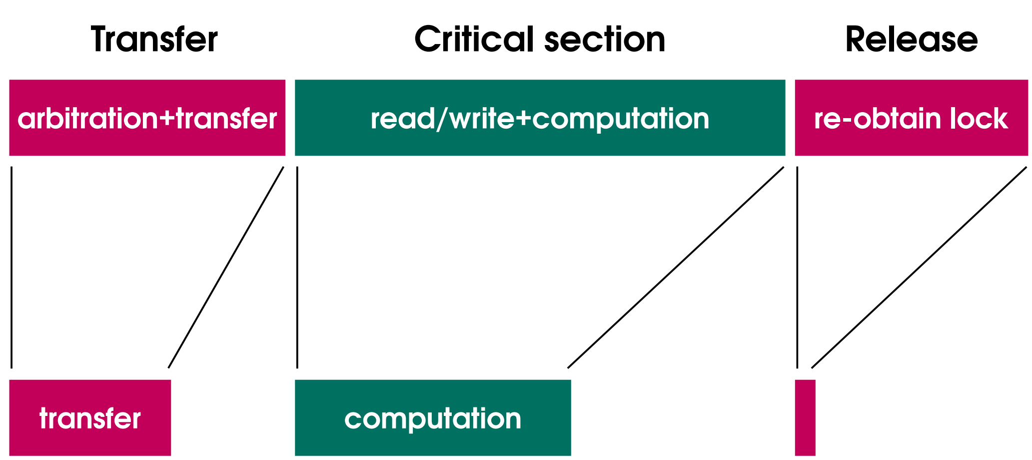
Computer Sciences Department

University of Wisconsin-Madison

Motivation



Synchronization period



Synopsis

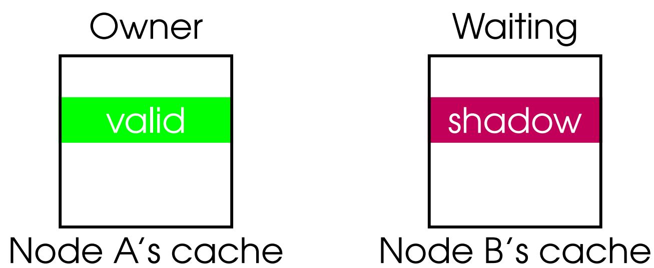
- 4 synchronization mechanisms
  - Local spinning
  - Queue-based locking
  - Collocation
  - Synchronous prefetch
- Categorizing synchronization primitives
- QOLB (Goodman, Vernon & Woest 1989)
- MCS (Mellor-Crummey & Scott 1991)
- QOLB
  - <u>47%</u> faster than MCS
  - Outperforms <u>all</u> other primitives <u>all</u> the time

Outline

- Motivation & synopsis
- Synchronization mechanisms
- Synchronization primitives
- Performance analysis
  - Methodology
  - Microbenchmark
  - Macrobenchmarks
- Conclusions

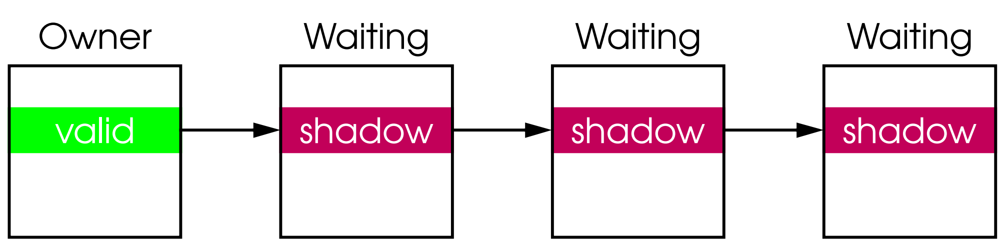
Local spinning

- What
  - Wait for synchronization *w/o* generating messages
- How
  - Read-only copy of the lock (Test&Test&Set)
  - Different addresses (MCS)
  - Shadow line (QOLB)



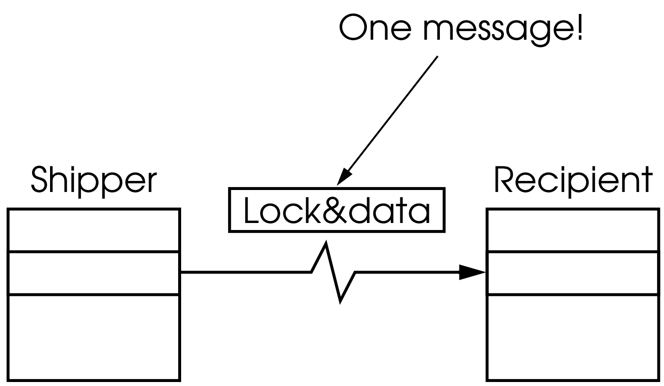
Queue-based locking

- What
  - Arbitration off critical path
  - Fixed number of messages per synchronization
  - Direct transfer of lock
- How
  - Queue in software (MCS)
  - Queue in hardware (QOLB)



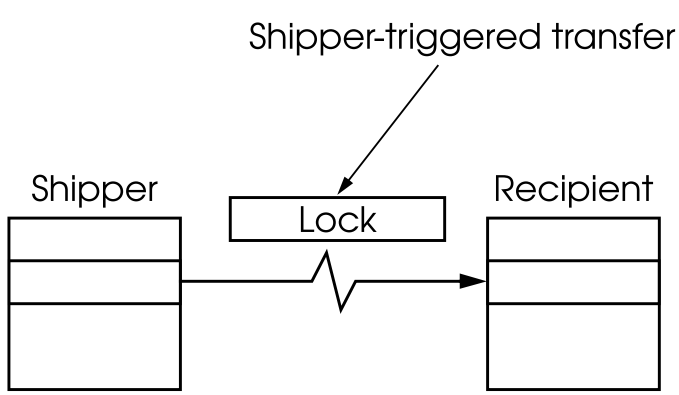
Collocation

- What
  - Reduce read/write misses
- How
  - Lock & data in *same* cache line

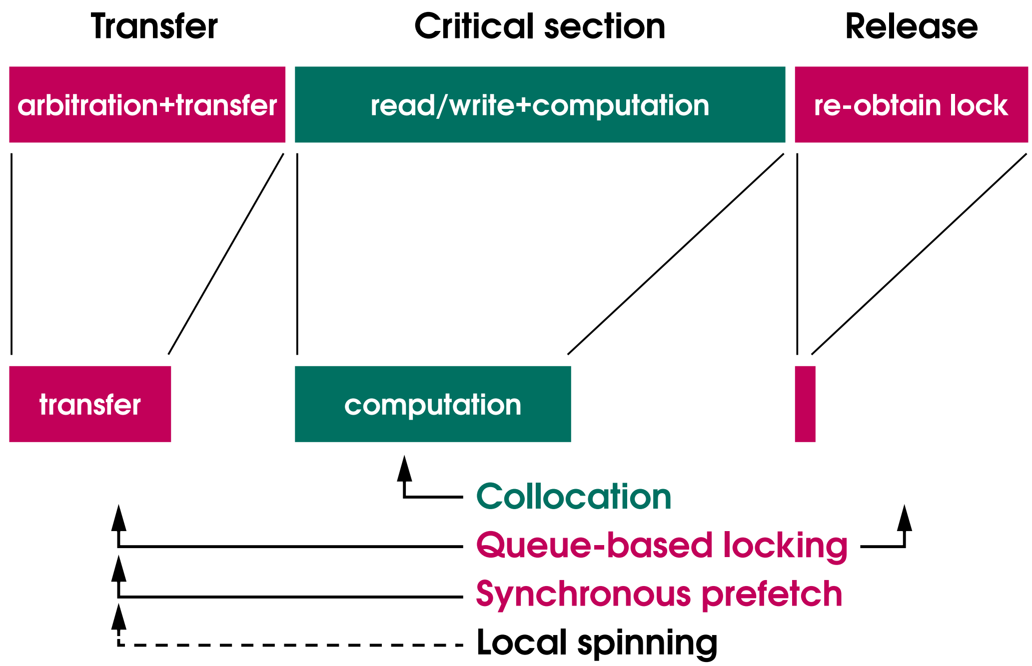


Synchronous prefetch (cooperative prefetch)

- What
  - Overlap transfer w/ computation
- How
  - Request synchronization ahead of time
  - Transfer *just* in time



Impact on overhead



Outline

- Motivation & synopsis
- Synchronization mechanisms
- Synchronization primitives
- Performance analysis
  - Methodology
  - Microbenchmark
  - Macrobenchmarks
- Conclusions

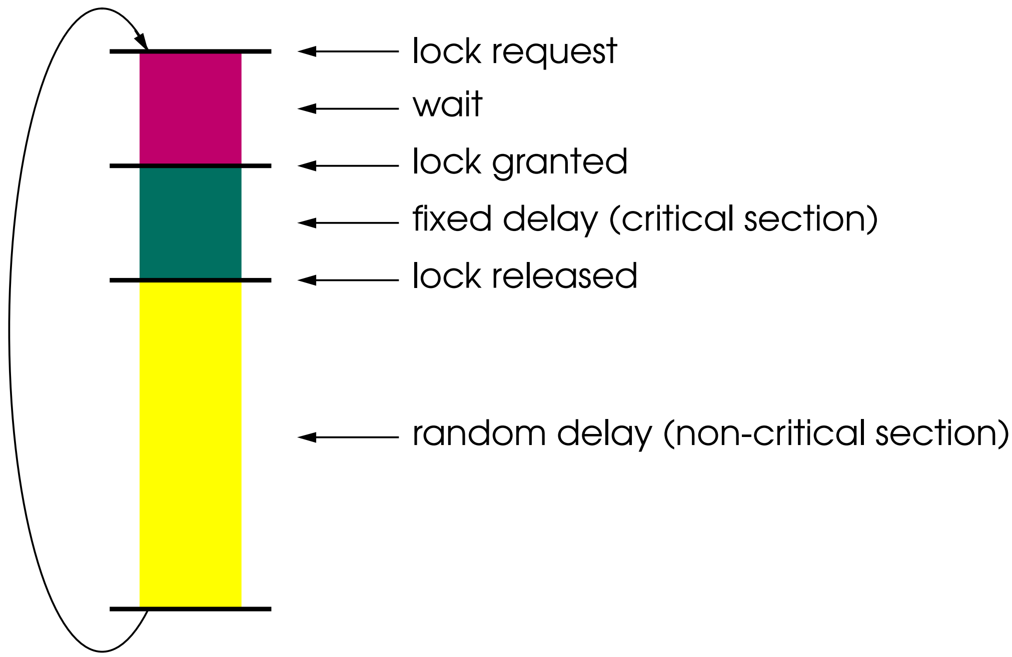
Synchronization primitives

Synchronization primitive	Synchronization mechanisms			
	Local spinning	Queue-based locking	Collocation	Synchronous prefetch
Test&Set (TS)	no	no	optional	no
Test&Test&Set (TTS)	yes	no	optional	no
MCS	yes	yes	no	no
QOLB	yes	yes	yes	yes

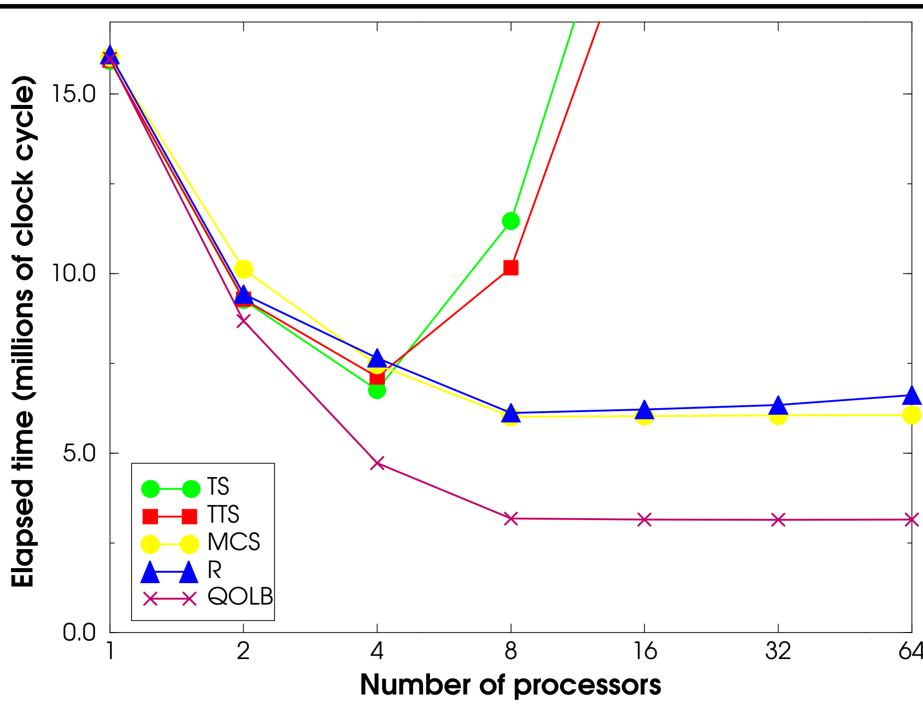
Evaluation method

- Simulated benchmarks on Wisconsin Wind Tunnel
  - 6 base primitives:
    TS, TTS, MCS, LH, M, QOLB
  - Vary mechanisms/policies:
    Collocation (+C)
    Reactive synchronization (R)
  - Microbenchmark + 5 applications
- Assumptions
  - 32 nodes
  - Processor: in-order issue
  - Cache: 1Mb 4-way assoc., 64b line, blocking
  - Constant network latency (~100 cycles)
  - Sequential consistency

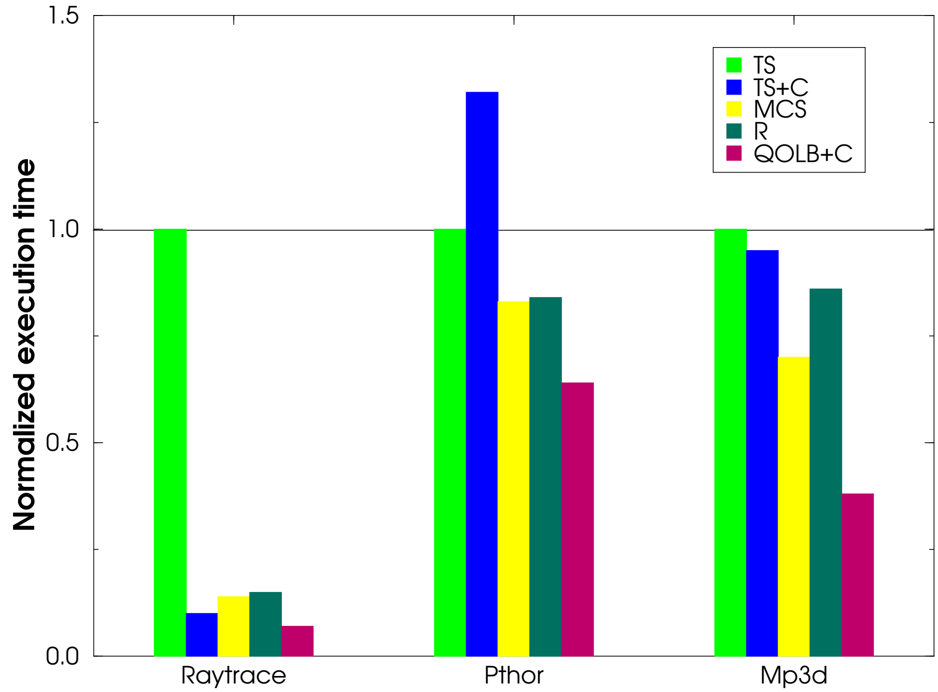
Microbenchmark



Results — microbenchmarks



Results — macrobenchmarks



Conclusions & other contributions

- Conclusions
  - 4 mechanisms
  - QOLB implements all 4
  - QOLB
    - Consistently outperforms all other primitives
    - 9% - 108% faster than MCS (avg. 47%)
    - 13% - 121% faster than Reactive (avg. 59%)

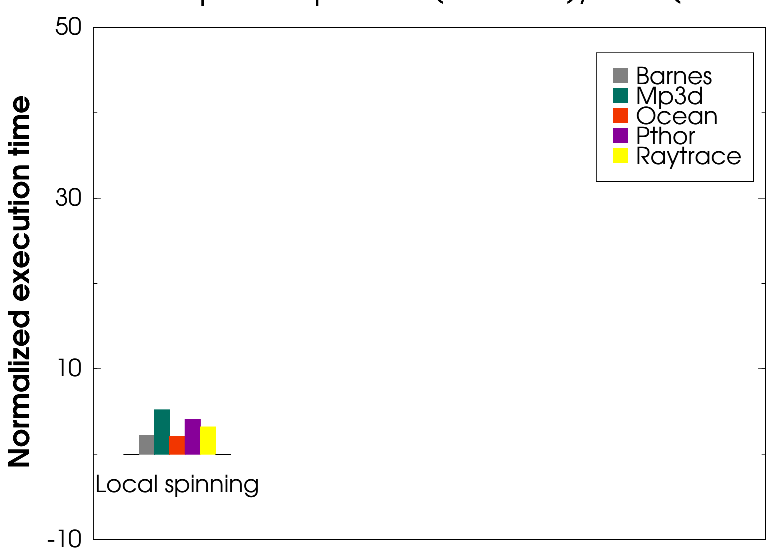
- Other contributions
  - QOLB-light
  - Isolating mechanisms

Microbenchmark

```
LOOP
    LOCK(X);
    DELAY(FIXED);
    UNLOCK(X);
    DELAY(RANDOM);
END LOOP;
```


Results — breakdown: local spinning

- Isolate local spinning
  - 2 runs: Test&Test&Set and Test&Set
  - Measure speedup: time(T&S run)/time(T&T&S run)



... and the others