Techniques for Reducing Overhead of Shared-Memory Multiprocessing

Alain Kägi, Nagi M. Aboulenein,
Douglas C. Burger and James R. Goodman

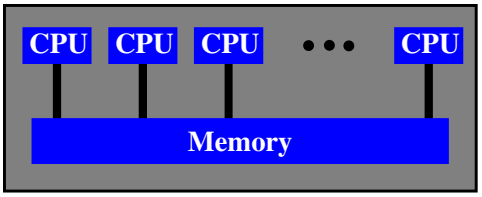
July 4, 1995

{alain, dburger, goodman}@cs.wisc.edu, nagi@cs.uno.edu
Computer Sciences Department
University of Wisconsin-Madison

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

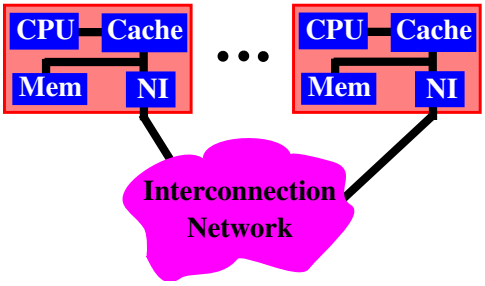
Shared-Memory Multiprocessing

Programming model



+ **uniform global address space**
+ **transparent communication**
+ **ease of programming**
+ **fine-grain communication**

Implementation



–but–

– **substantial overhead**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Goal & Outline

Three opportunities to reduce overhead

- **Synchronization**
- **Memory model**
- **Data sharing patterns**

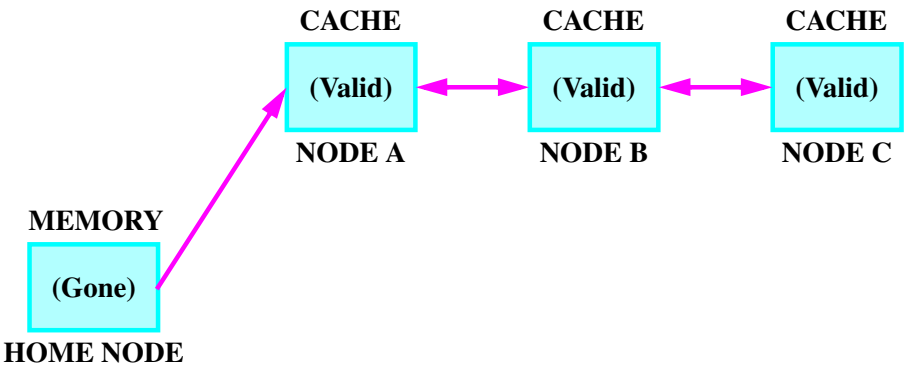
Outline

- **Scalable Coherent Interface**
- **Three classes of optimizations**
- **Experiments/methodology**
- **Evaluation**
- **Summary**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Scalable Coherent Interface (SCI)

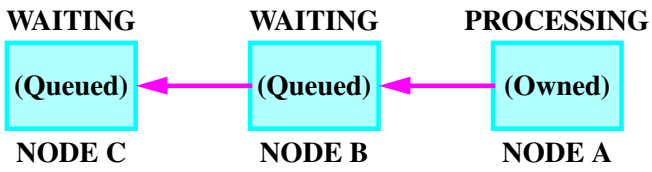
- **Shared memory in hardware**
- **Complete specification (IEEE standard)**
- **Distributed, directory-based cache-coherence protocol**



© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Synchronization Overhead

- **QOLB [ASPLOS III, 1989]**
  - **Hardware queue**
  - **+ reduced network traffic**
  - **+ reduced lock handoff latency**
  - **+ can couple lock & data**
  - **+ prefetch**
  - **– hardware implementation cost**



© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Synchronization Overhead (cont'd)

- **MCS [ASPLOS IV, 1991]**
  - **QOLB-inspired software solution**
  - **+ no hardware required**
  - **+ reduced network traffic**
  - **– can't couple lock & data**
  - **– can't prefetch**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Memory Consistency Overhead

- Sequential consistency
  - + intuitive
  - – overly restrictive
- Release consistency
  - + operations complete out of order
  - – programming complexity
  - – hardware support
- We study
  - Sequential consistency (seq)
  - 2 implementations of release consistency (r1, r2)

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Pairwise Sharing

- Optimize specific sharing patterns
  - No protocol is perfect
  - Extend for common patterns
  - 2-node sharing can be common
  - Pairwise sharing
    - Extend SCI for 2-node sharing

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Methodology

- **Wisconsin Wind Tunnel (WWT)**
  - **Discrete-event simulator on CM-5**
  - **Directly execute parallel shared-memory programs**
- **Benchmarks**
  - **Barnes, Mp3d, Ocean, Pthor, Water**
  - **Optimized for each experiment**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Methodology (cont'd)

- **Simulation parameters:**
  - **Target system**

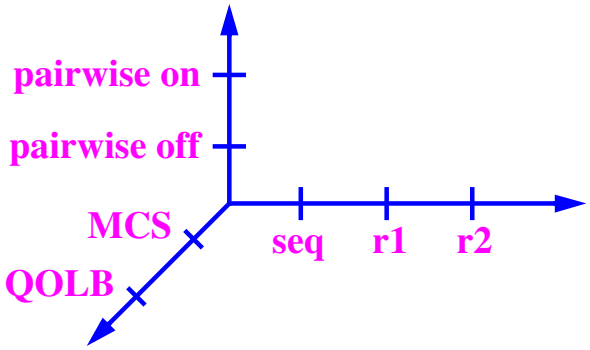
    32 processors, 1MB caches,

    4-way assoc., 64 byte cache line
  - **Non-blocking stores (write queue)**
  - **Validated constant network latency**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Experiments

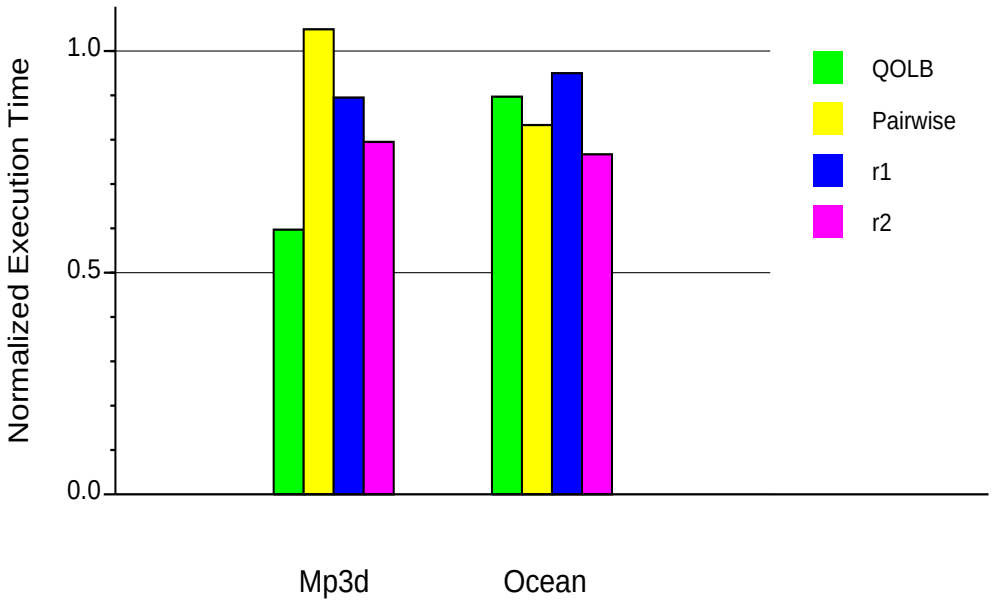
- **Full cross product**



- **Base case: MCS, seq, pairwise off**
- **Future technology effects**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Experimental Results for Current Technology



© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Optimization Interaction

Interaction	Benchmark	
	Mp3d	Ocean
Q+P	104%	106%
Q+r1	78%	93%
Q+r2	59%	88%
P+r1	97%	95%
P+r2	99%	57%

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Summary

- **Synchronization**
  - **QOLB very effective**
- **Memory model**
  - **r1 moderately effective and cheap**
  - **r2 very effective but expensive**
- **Pairwise sharing**
  - **Extremely good if 2-node sharing**
- **Future technology**
  - **Same trends, more dramatic speedups**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Future Work

- **Breakdown of QOLB**
- **Scalability**
- **Percentage of overhead**
- **QOLB prefetch**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

■ 9th ACM International Conference on Supercomputing ■

Cache Size Issue

- **Problem**
  - **Benchmarks don’t strain large caches**
  - **Does this qualitatively affect results?**
- **Solution**
  - **Simulate very small (8K) cache**
  - **Compare optimization effects**
  - **Only pairwise sharing changed**

© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison
Techniques for Reducing Overhead of Shared-Memory Multiprocessing
Backup

Optimization Interaction

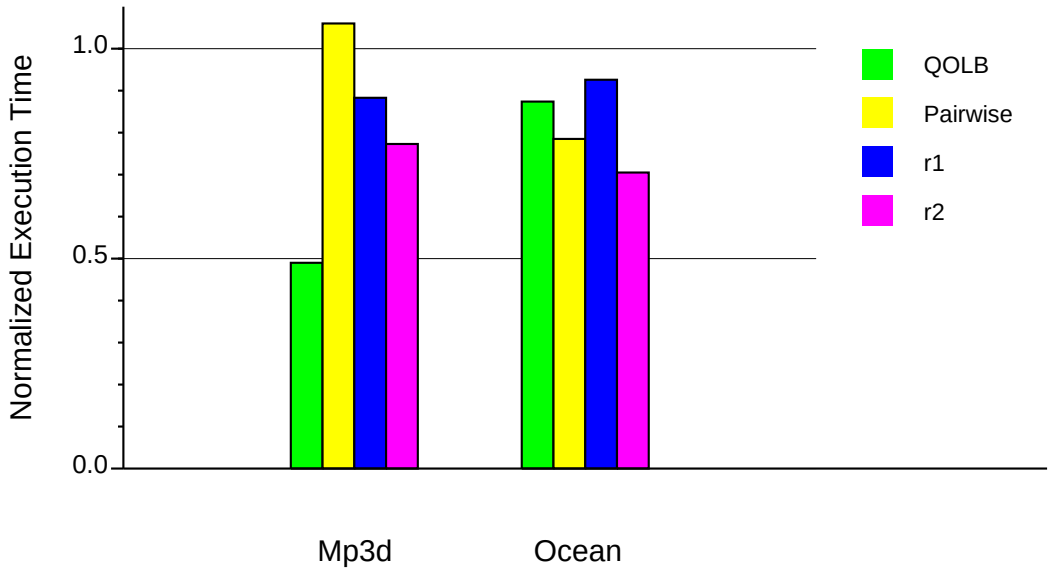
Interaction	Benchmark	
	Mp3d	Ocean
Q+P	104%	109%
Q+r1	16%	86%
Q+r2	19%	74%
P+r1	97%	82%
P+r2	99%	40%

0.0% := no interaction

100.0% := additive

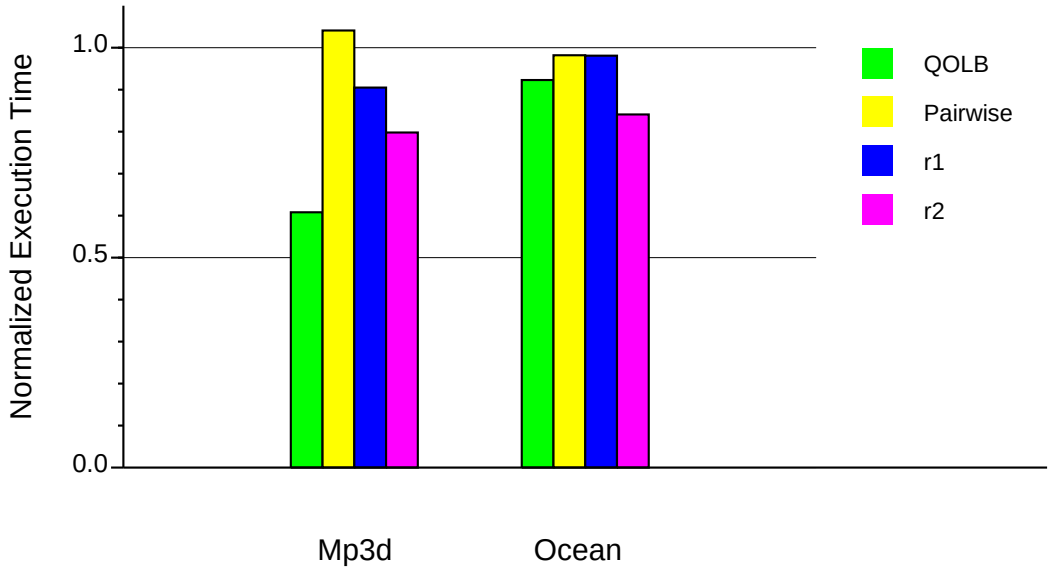
© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Experimental Results for Future Technology



© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison

Experimental Results for Small Cache



© 1995 by Alain Kägi
Computer Sciences Department
University of Wisconsin-Madison