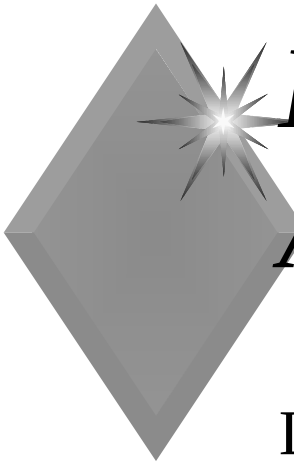
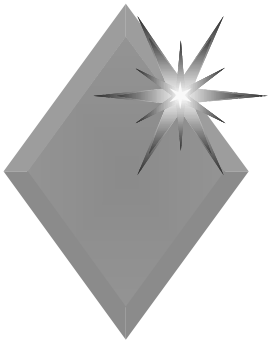
digital™



VM-Based Shared Memory on Low-Latency, Remote-Memory-Access Networks

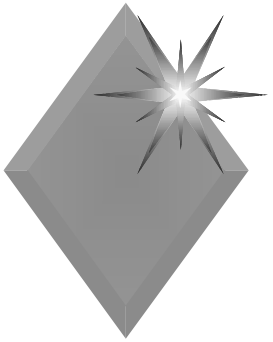
Leonidas Kontothanassis, Galen Hunt, Robert Stets, Nikolaos Hardavellas, Michal Cierniak, Srinivasan Parthasarathy, Wagner Meira Jr., Sandhya Dwarakadas, and Michael Scott.

University of Rochester
Computer Science



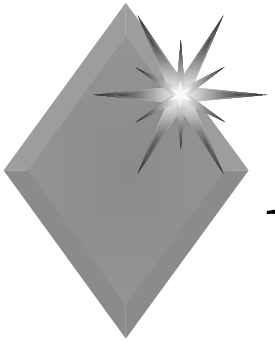
Goal

- Transparent Shared Memory on Clusters of SMPs.
- How to best exploit the “special” abilities of a remote-memory-access network?



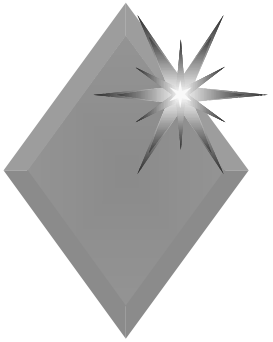
Outline

- Memory Channel API.
- Cashmere and TreadMarks implementations on Memory Channel.
- Methodology.
- Memory Channel microbenchmarks.
- Performance results.
- Future Work and Conclusions.



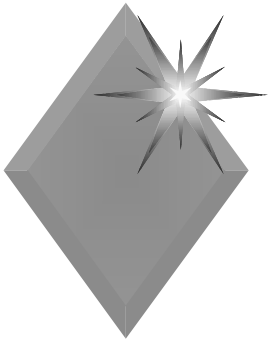
Memory Channel API

- Create transmit and receive segments.
- When writing into a transmit segment the write appears on all receive segments with the same segment identifier.
- Total ordering of writes
  - Allows for implementation of synchronization primitives



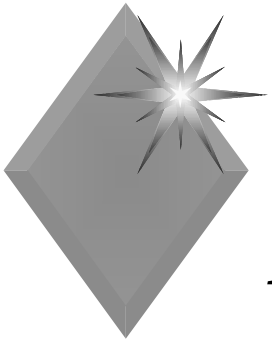
Cashmere Implementation

- Release-consistent multi-writer protocol.
- Uses directories to maintain sharing information.
- Uses write-through via write-doubling to collect writes from multiple writers.
- Coherence granularity is a VM-page.
- Invalidation notices propagated at release and processed at acquire sync. points.



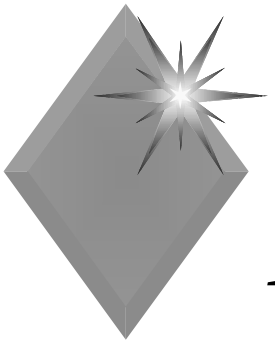
TreadMarks Implementation

- Release-consistent multi-writer protocol.
- Uses vector timestamps and sync. chains to maintain memory coherence.
- Uses “twins” and “diffs” to collect writes from multiple writers.
- Coherence granularity is a VM-page.
- Invalidation notices requested and processed at acquire sync. points.



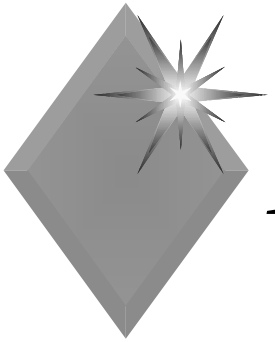
Cashmere: Pros and Cons

- \+ Write notices sent only when data is shared.
- \+ Merging via write-through allows processor to get new version of data in one operation.
- \+ Write-through may be overlapped with computation.
- – Write-through increases traffic.
- – May cause more invalidations since it doesn’t track happens-before.



TreadMarks: Pros and Cons

- \+ Lazier Implementation may cause less invalidations.
- \+ Diffs and Twins generate less traffic.
- – May require multiple requests to update a page.
- – May send unnecessary invalidation notices.

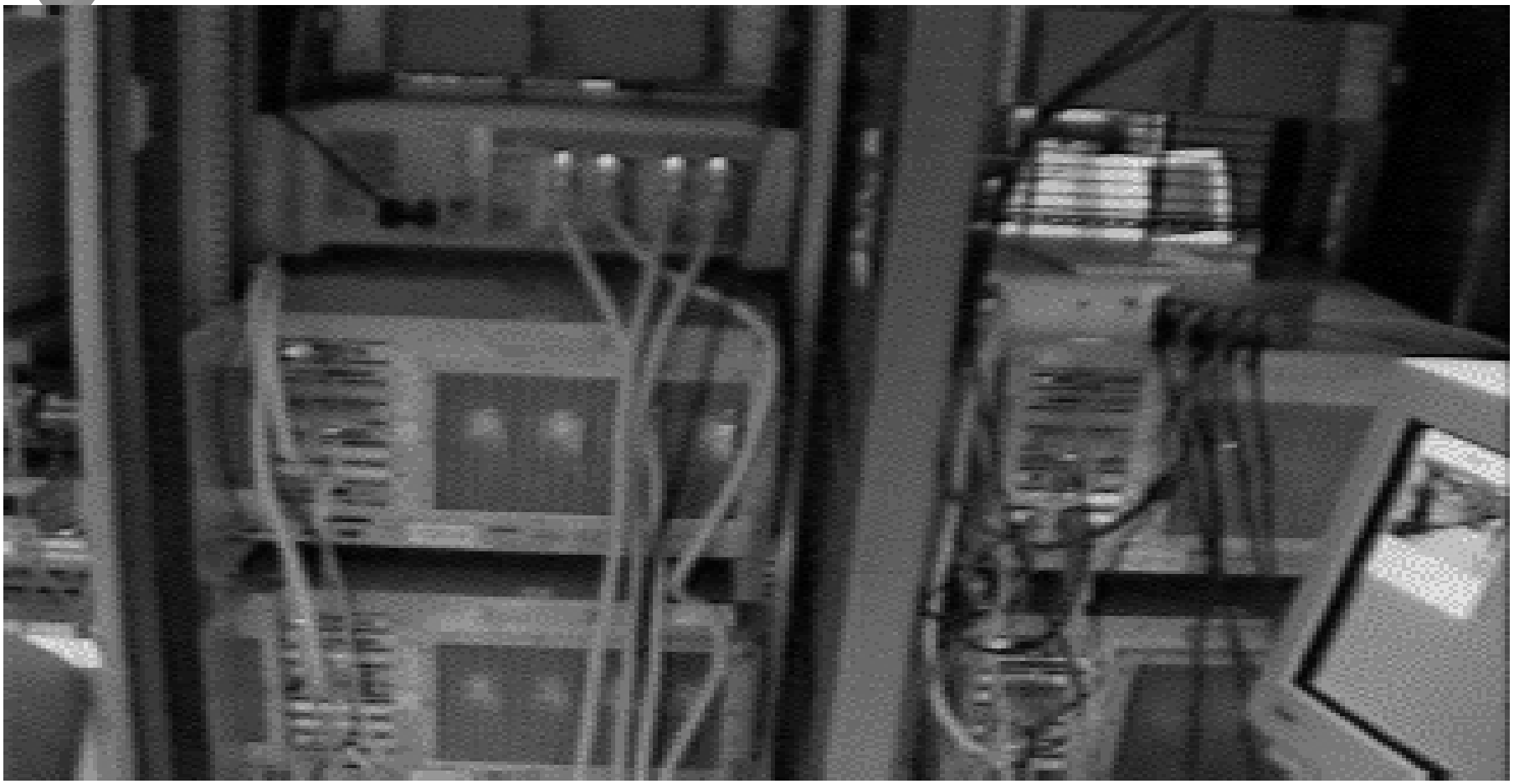


Methodology

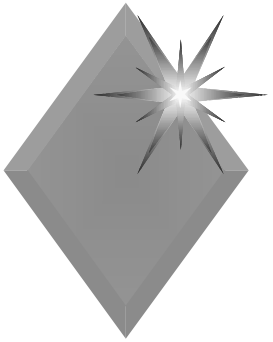
- Eight DEC AlphaServer 2100 4/233 SMPs connected with Memory Channel.
- DEC Alpha 21064A processors at 233Mhz with 16K I- and 16K D-cache on chip and 1Mbyte B-cache.
- Point to point bandwidth is 30Mbytes/sec.
- Aggregate bandwidth is 32Mbytes/sec.

digital™

Methodology

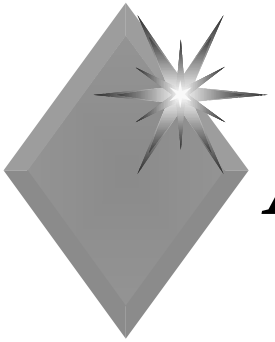


University of Rochester
Computer Science



MicroBenchmarks

OP	*CSM-INT*	*CSM-POL*	*TMK-INT*	*TMK-POL*
lock	11usec	11usec	976usec	79usec
barrier	208usec	205usec	5432usec	1213usec
pagefetch	1960usec	742usec	1962usec	784usec
fault	89usec	89usec	89usec	89usec
twin	N/A	N/A	362usec	362usec
diff	N/A	N/A	289-533us	289-533us

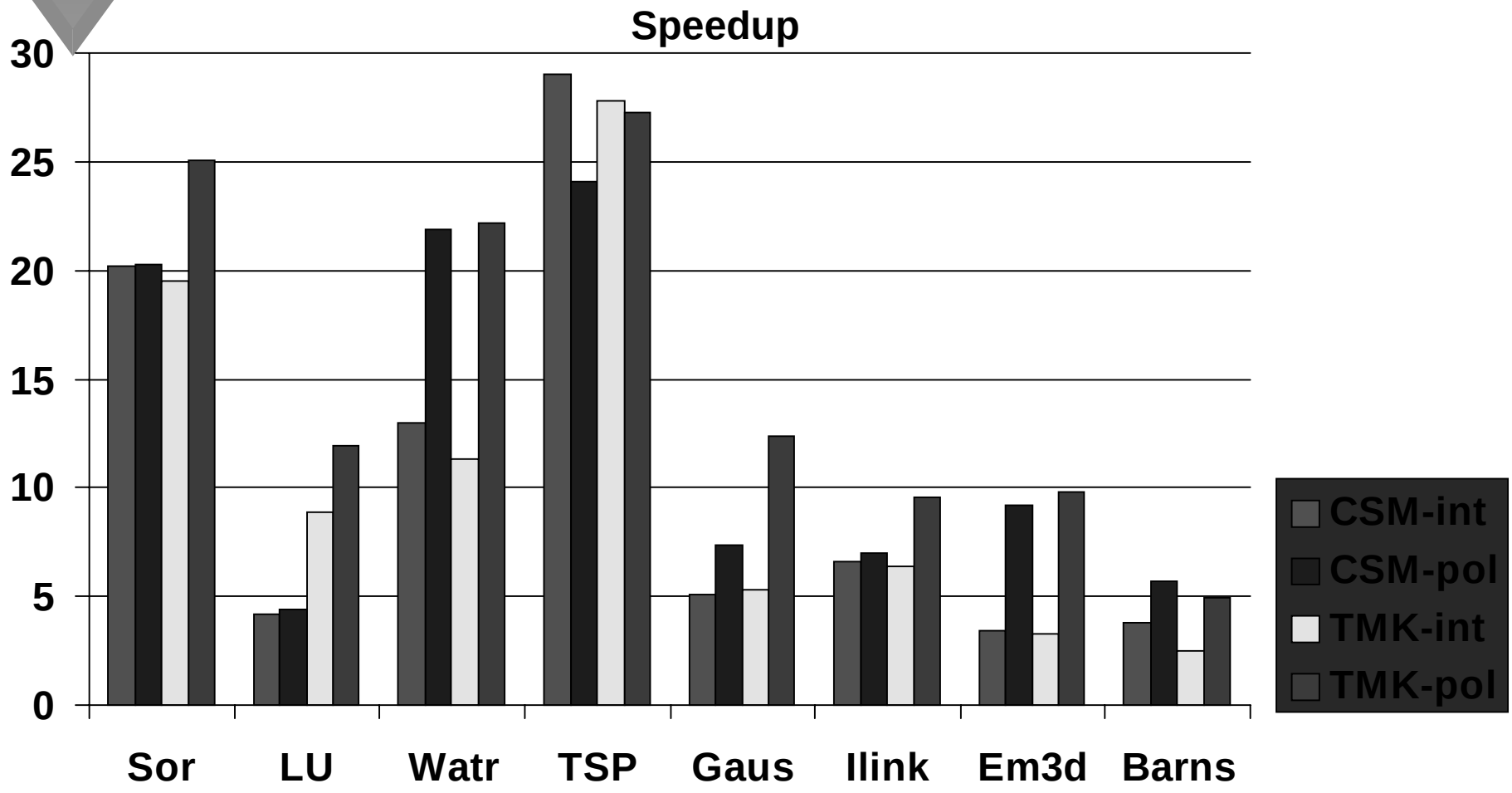


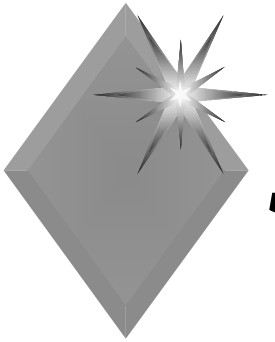
Applications

Program	***Problem Size***	***Time (sec)***
SOR	3072X4096 (50M)	194.96
LU	2046X2046 (33M)	254.77
Water	4096 mols (4M)	1847.56
TSP	17cities (1M)	4028.95
Gauss	2046X2046 (33M)	953.71
Ilink	CLP (15M)	898.97
Em3d	60106 nodes (49M)	161.43
Barnes	128K bodies (26M)	469.43

digital™

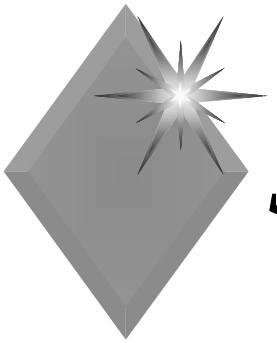
Results





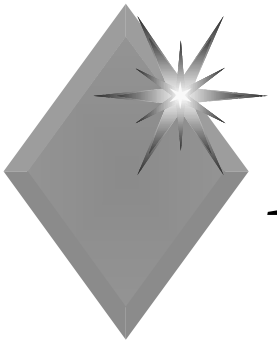
Sources of inefficiency

- Cache interference due to doubling of writes.
  - Solutions: (Use twins/diffs). Twins/diffs have been adopted for both the second generation of the 1-level and a future 2-level protocol.
- High cost of locks on directory accesses.
  - Solutions: Redesign directory so that no locking is necessary (more memory intensive).

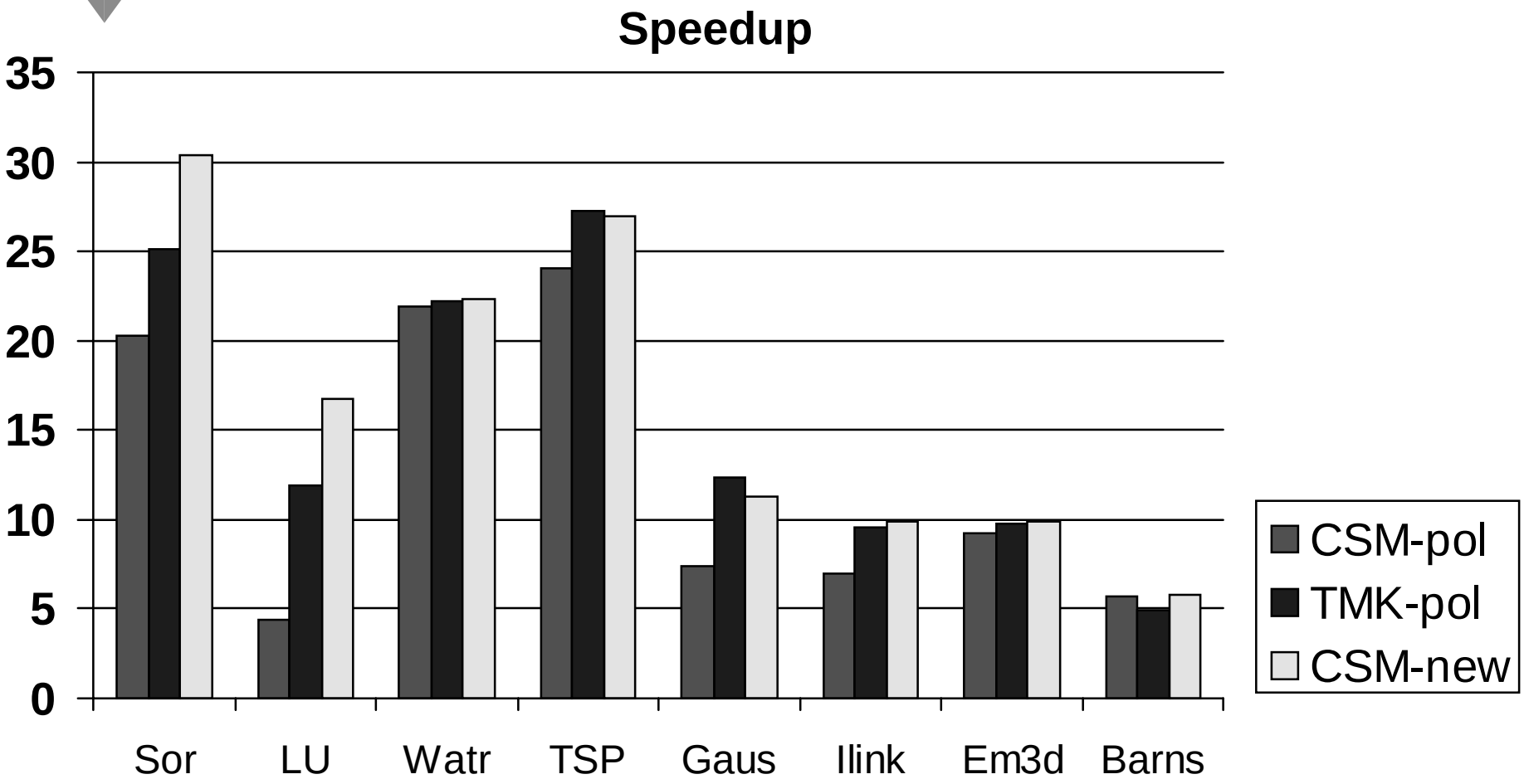


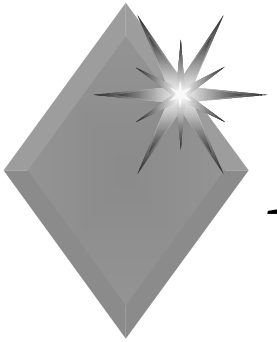
Sources of inefficiency

- Unnecessary coherence transactions on essentially private data (I.e. internal rows in Sor, partial results on pivots for Gauss).
  - Solution: Introduce new exclusive state into the protocol.



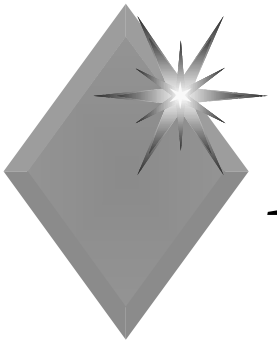
Results


Speedup
35
30
25
20
15
10
5
0
Sor
LU
Watr
TSP
Gaus
Ilink
Em3d
Barns
CSM-pol
TMK-pol
CSM-new



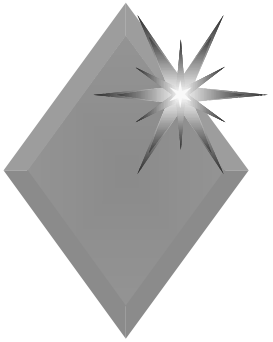
Future work

- Two-level protocol.
  - Exploit intra-node hardware cache coherence.
  - Minimize page transfers and exploit sharing between processors within a node.
- What to place in MC space?
  - Nothing (only used for message passing).
  - Just metadata.
  - Data and metadata (current version of CSM)



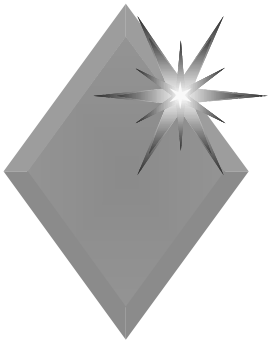
Future Work

- Very Large Memory DSM.
  - Scale memory with size of cluster.
  - Allow dynamic number of processes in DSM
  - Support pthreads within a DSM process.



Conclusions

- Write-doubling in software is a bad idea.
- Low latency networks make directories a viable alternative for DSM.
- Software cache coherence does work for scientific apps.
  - Single System Image and tools still a serious limitation for wider acceptance of software DSM.



Questions

University of Rochester
Computer Science