The Cashmere Project

Michael Scott,
Sandhya Dwarkadas,
and students

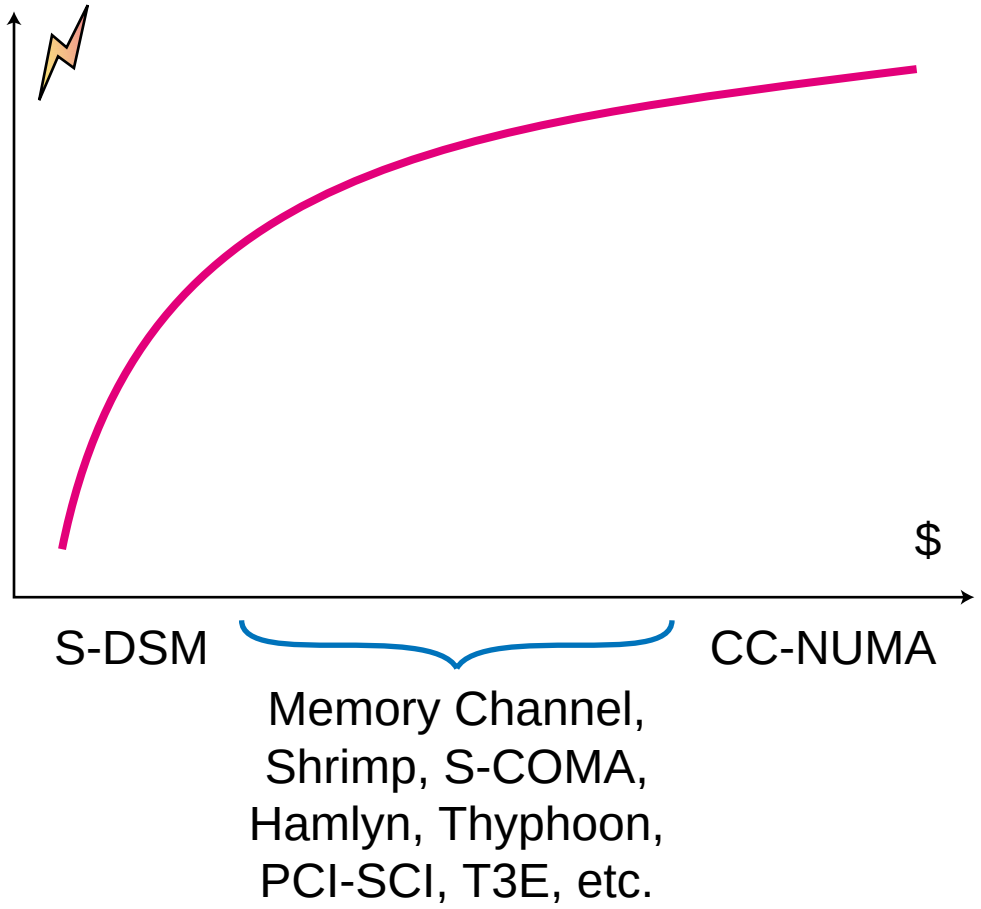
Department of Computer Science
University of Rochester

- Background and motivation
- Status and results
- Current and future work

Background and Motivation

- Shared memory a highly attractive programming model
- Wide hardware spectrum
  - » DSM/SVM
  - » Full hardware coherence
  - » Options in-between
- Hardware is faster, but software
  - » Is cheaper
  - » Can be built faster (sooner to market, faster processors)
  - » Can use more complex protocols
  - » Is easier to tune/enhance
  - » Is easier to customize

The Price-Performance Curve



Q: How should coherence work given very low latency user-level messages?

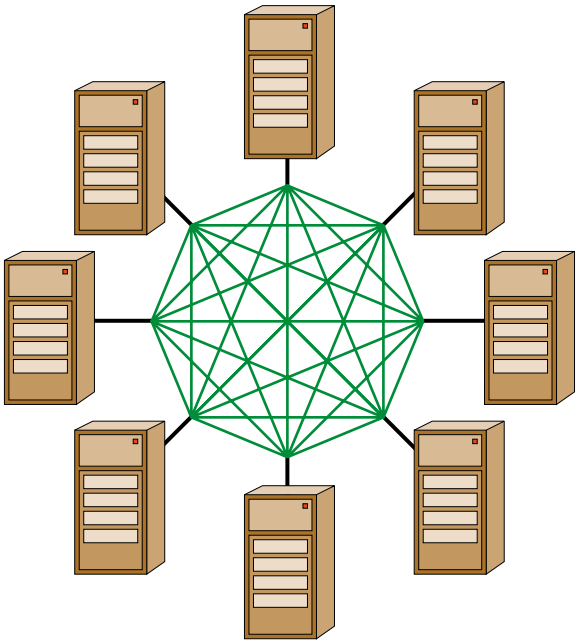
Project Summary

- PREMISE: with appropriate software, a cluster of server-class SMPs with a very low-latency network can provide supercomputer-class performance for shared-memory applications.
- GOAL: verify this premise on a commercial platform

- Funding from NSF ESS and IIP (RI) programs
- Major hardware support from Digital Equipment Corp (Compaq)
- Compiler integration funded through Dwarkadas CAREER award

Hardware platform

- 8-node cluster of 4-way SMPs -- 32 processors total



- Past work: 233 MHz EV45s, 2 GB total memory; 5us remote latency; 30 MB/s per-link bandwidth; 60 MB/s total bandwidth
- Recent acquisition: 600 MHz EV56s, 16 GB total memory; 3us remote latency; 70 MB/s per-link bandwidth; >500 MB/s total bandwidth

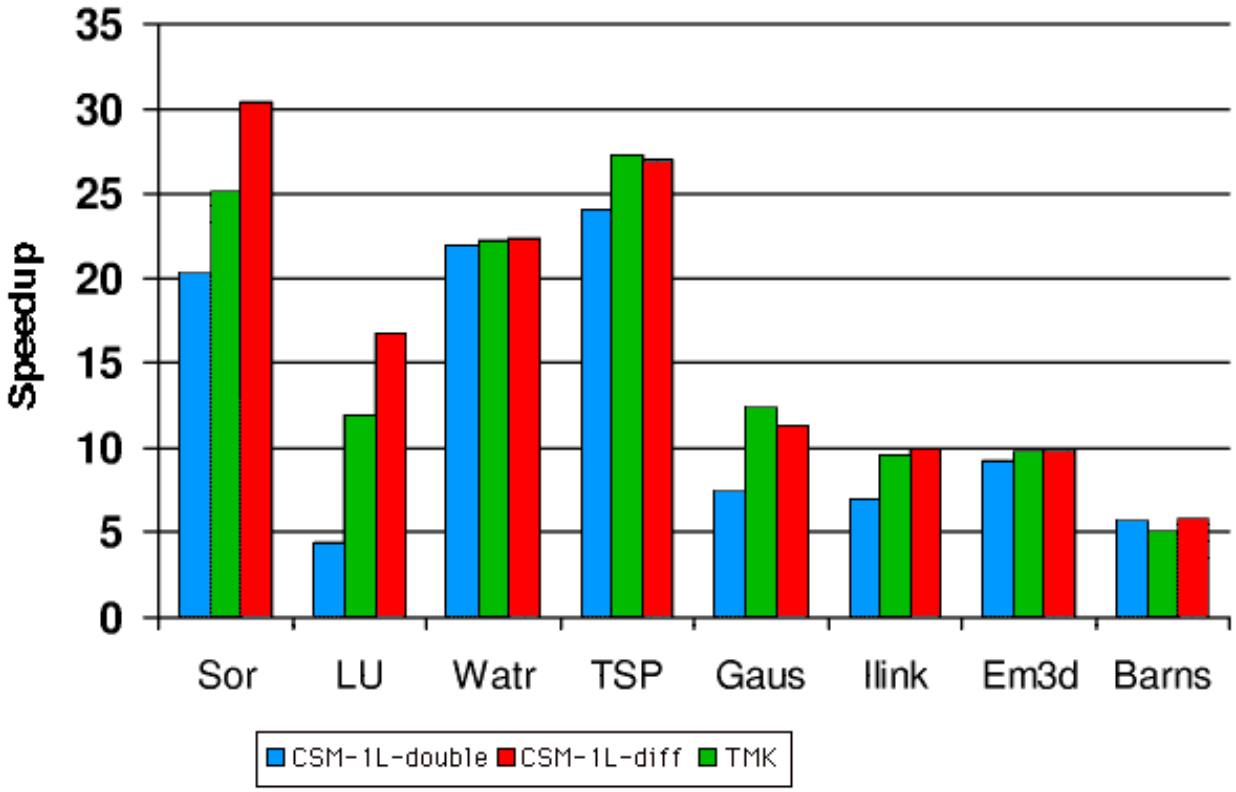
Core Cashmere Features

- Page-size coherence blocks
- Data-race-free programming model
- “Moderately lazy” protocol; multiple concurrent writers to a page
- Master copy of data at home node
- Distributed directory

Early “1-level” Protocol

- Each processor a separate “node”
- All shared data in Memory Channel space
- Merging of changes via
  - » On-the-fly write doubling
  - » Twins and diffs
- Comparison to TreadMarks; results reported at ISCA ‘97

The Feasibility of Remote Writes



Cashmere-1L v. Treadmarks

8-way (32-processor) 2100/4-233 system

Cashmere-2L

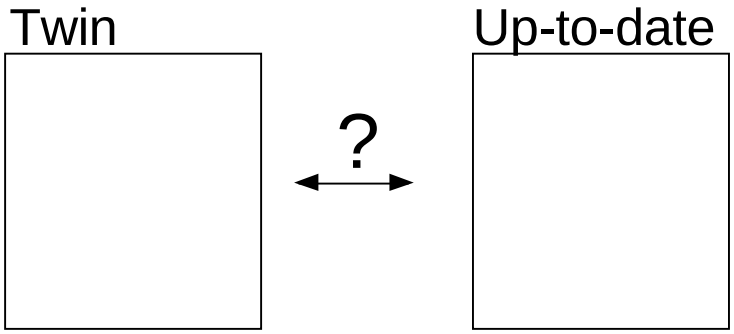
- HW coherence within nodes, SW coherence among nodes
- Twins and *two-way* diffs
- No shootdown
- Largely asynchronous protocol (lock-free meta-data)
- Results reported at SOSP ‘97

Protocol Operations

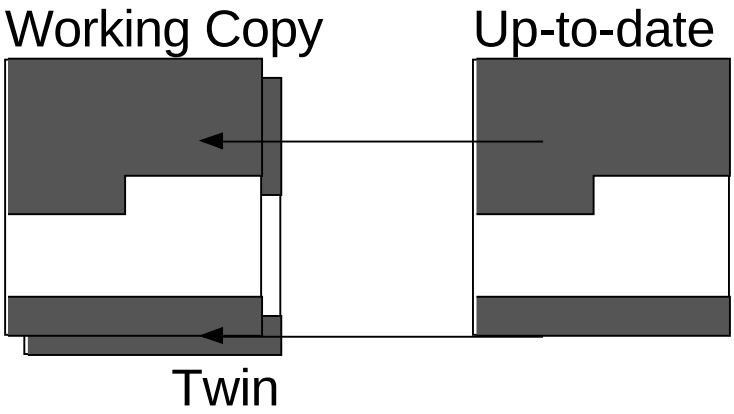
- Page Fault
  - » Update global page state information
  - » *Page Update*: Obtain up-to-date page data (incoming diff)
- Release
  - » Send modifications to the home node, via twins and diffs [Munin, Home-based LRC]
  - » Send *write notices*
- Acquire
  - » Invalidate all pages named by write notices

Incoming Diffs

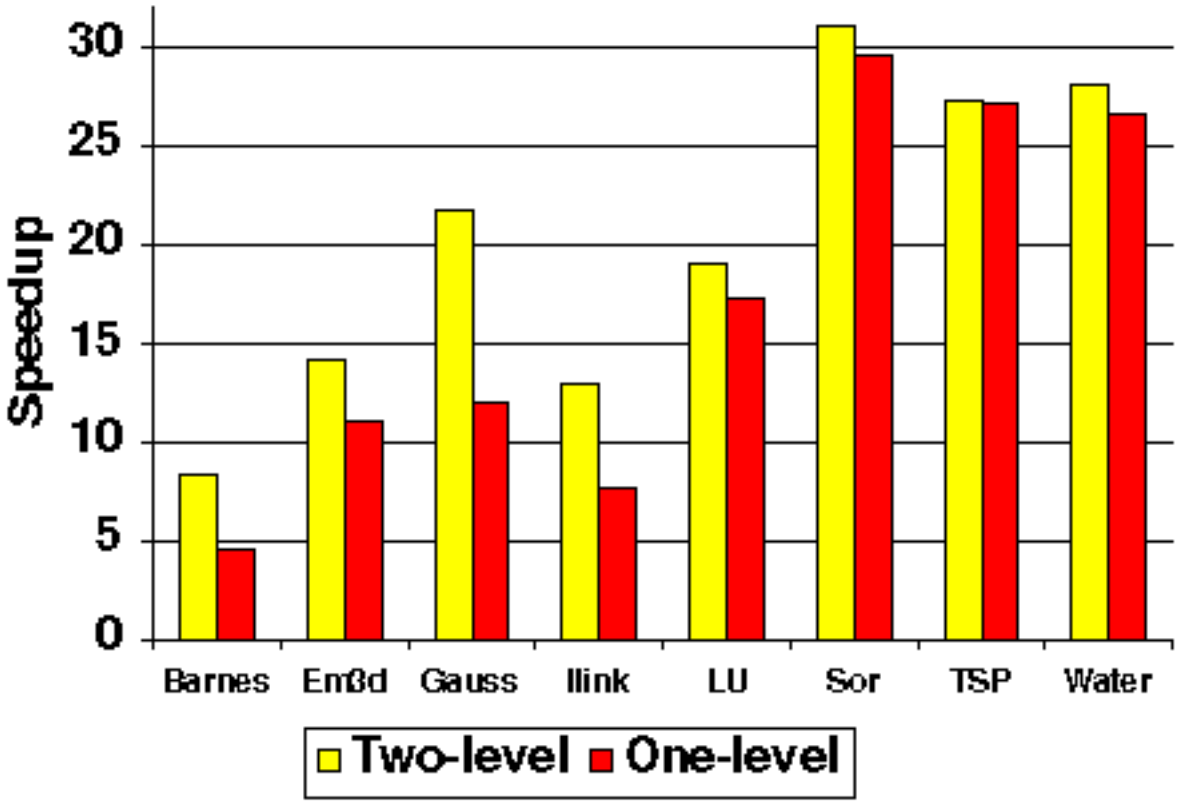
Compare up-to-date data to the twin.



Copy differences to the working copy and the twin.



The Value of Clustering



8-way (32-processor) 2100/4-233 system

Recent Changes

- Very large (out-of-core) data sets [SSMM ‘98]; meta-data only in Memory Channel space
- Migrating home nodes

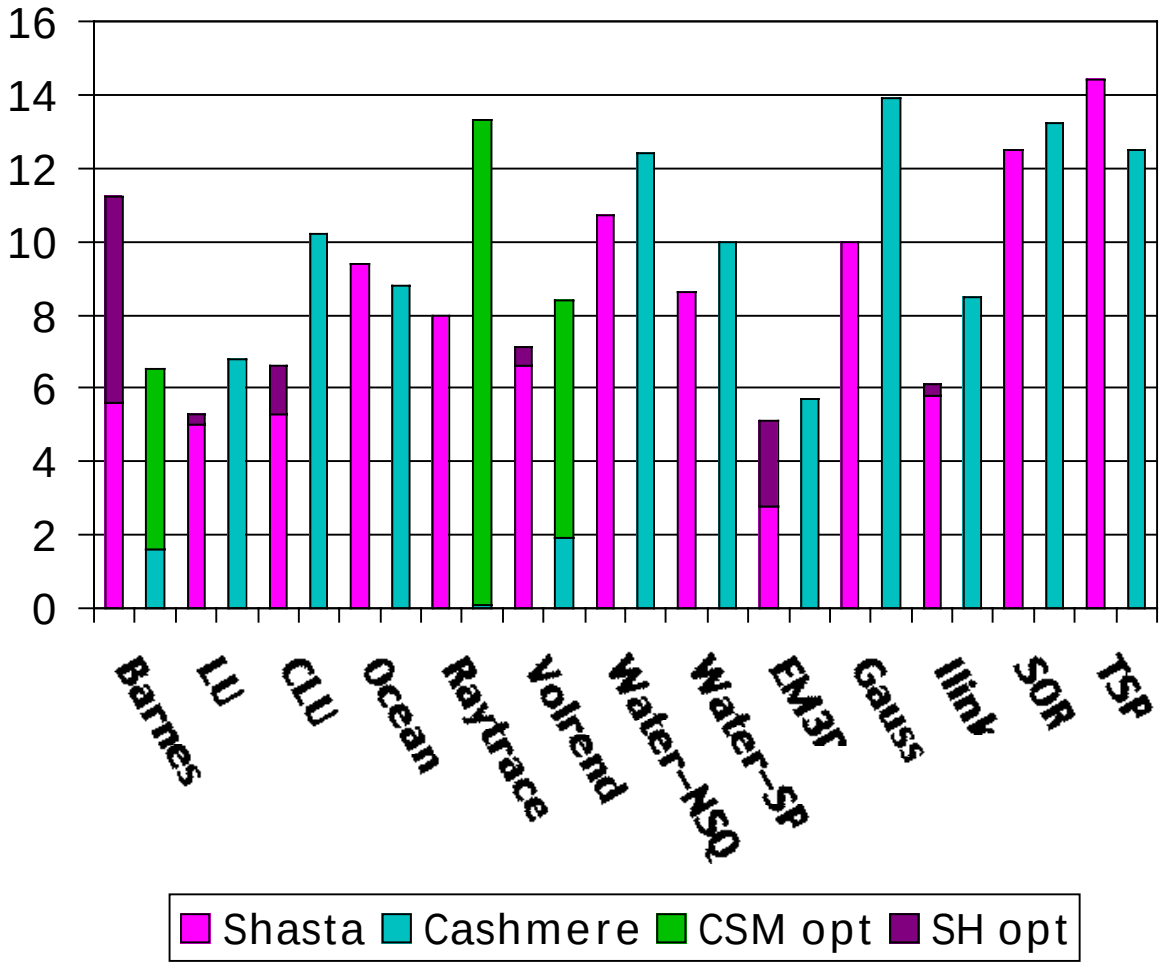
⇨ Minimal net effect on performance

- Comparison to Shasta [SSMM ‘98; HPCA ‘99]
- Intelligent paging of out-of-core data sets [IPPS ‘99]
- Elimination of MC dependences [TR]
  - » remote writes
  - » total message ordering, broadcast

Comparison to Shasta

- Shasta [Gharachorloo and Scales, 1997] runs on the same hardware as Cashmere
- Unlike Cashmere, Shasta has:
  - » Arbitrary (e.g. small) coherence granularity
  - » In-line protocol operations
  - » No dependence on MC broadcast, ordering
  - » No need for application source
- The downside:
  - » Overhead on every load and store
  - » Smaller transfer granularity

Performance of Cashmere and Shasta



- 16-processor 400 MHz cluster at DEC WRL

Fast Messages (Only) Protocol

- Non-replicated directory
- Message-based synchronization
- Acknowledgments for write notices
- Slowdowns from zero (CLU, Ilink, TSP) to 37% (Volrend). 7-10% typical. (Problem in Volrend is task stealing)

⇨ Results suggest that remote writes, broadcast, and total order matter, but latency matters more.

Conclusions

- S-DSM can yield good speedups, at least on modest numbers of nodes
- HW and SW coherence can work well together
- Low-latency messages, home node migration, and HW coherence within nodes are all crucial
- Remote write, broadcast, and total order are secondarily useful

- Open question: how do things scale to very large nodes or very large numbers of nodes?

Future Plans

- Protocol tweaks, VIA port
- Pseudo-single system image
- “Three-level” coherence protocol (InterAct); heterogeneity
- Java support
- Compiler integration (ARCH)
- Fault tolerance
- Application studies
  - “Cone beam” CAT-scan reconstruction
  - N-body simulation
  - Volumetric reconstruction
  - Object recognition
  - Neural simulation

Technology Transfer

- Papers
- Students (Kontothanassis, Hardavellas, Hunt, Zaki; Parthasarathy, Stets)
- Patent application on two-way diffing (joint with Digital)
- Modifications to Digital (OSF) Unix
  - » user-specified VA for MC regions
  - » mprotect on MC regions
  - » fast interrupts
- Field test bug reports