Cashmere-2L: Software Coherent Shared Memory on a Clustered Remote-Write Network

Robert Stets, Sandhya Dwarkadas, Nikolaos Hardavellas, Galen Hunt, Leonidas Kontothanassi, Srinivasan Parthasarathy, and Michael Scott

University of Rochester Computer Science

¹Digital Equipment Corporation Cambridge Research Laboratory
Cost Effective Parallel Systems

Remote-Write Network

SMP Nodes

- Excellent platform for software-based distributed shared memory (SDSM) protocols.
Cashmere-2L

- Protocol description
  - Principal operations
  - Performance advantages
- Performance results
  - One level vs. two level
  - Effectiveness of reduced protocol synchrony
- Conclusion and future work
Basic Protocol Design Principles

- Virtual Memory Faults (Page-based)
- Home-node based
- Directory-based
- Multiple Writer
Principal Protocol Operations

- **Page Faults**
  - Update global page state information
  - *Page Update*: Obtain up-to-date page data

- **Release**
  - Send modifications to the home node, via twins/diffs [Munin, Home-based LRC]
  - Send *write notices*

- **Acquire**
  - Invalidate all pages named by write notices
Key Performance Advantages

- Each processor in a node shares same page frame.
- Exploit remote-write network capabilities
  - Broadcast of directory modifications
  - Remote delivery of diff and write notices
  - Polling-based messaging
  - Fast application locks and barriers
Protocol Levels: Synergy

- Hardware coherence effectively performs coherence operations for the entire node.
- Redundant operations are avoided.
- Per-node logical clocks are used to timestamp key events, e.g.
  - Last write notice received (per-page)
  - Last Update (per-page)
Avoiding Redundant Updates

Node 2

P1: Acq Read Rel

P2: Acq Read

- Logical Clock: 5 → 6 → 6 → 7 → 7 → 7
- Last Update: 3 → 3 → 3 → 7 → 7 → 7
- Write Notice: 2 → 6 → 6 → 6 → 6

Page Update
NoPage Update

Write Notice
Protocol Levels: Compatibility

- Page Update operation should respect the modifications of local concurrent writers.
- Established technique
  - Shootdown all concurrent writers in the node.
- Cashmere-2L technique
  - Incoming Diffs
Compare up-to-date data to the twin.

Copy differences to the working copy and the twin.
Hardware Platform

- Thirty-two 233MHz 21064A processors (Eight AlphaServer 2100 4/233 SMPs)
  - 16K icache, 16K dcache on-chip caches
  - 1M board-level caches
- DEC Memory Channel I Network
  - One-way latency: 5.2 μs
  - Bandwidth
    » 29 MBytes/s per-link
    » 60MBytes/s aggregate
Performance: Two- vs. One-level

32 Processors

Speedup

Barnes  Em3d  Gauss  Ilink  LU  Sor  TSP  Water

[Bar chart showing speedup comparison between two-level and one-level for various applications: Barnes, Em3d, Gauss, Ilink, LU, Sor, TSP, Water. Two-level speedup is represented in blue, one-level speedup in green.]
Execution Breakdown: 2L vs. 1L

32 Processors

Normalized Execution Time (%)
Incoming Diffs vs. Shootdown

- *No performance difference!*
  - Seems to contradict SoftFlash results

- *Important protocol design decisions*
  - Concurrent writers?
    » SoftFlash: single-writer
    » Cashmere-2L: multiple-writer
  - Page Tables
    » SoftFlash: shared
    » Cashmere-2L: separate
Two-level design provides significant performance improvements.

Remote-write network handles directories well.

Multiple-writer protocol and independent page tables reduces need for shootdown operations.
Future Work

- Continue improving overall performance.
  - Migrating home nodes.
    - Adaptive invalidate/update mechanism.
- Support new classes of applications.
  - Very large-scale resident data sets.
  - Out-of-core data sets.
- Examine impact of variable coherence granularities. (e.g. Shasta)