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


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

University of Rochester
Computer Science
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.

University of Rochester
Computer Science
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.
Methodology

University of Rochester
Computer Science
# MicroBenchmarks

<table>
<thead>
<tr>
<th>OP</th>
<th>CSM-INT</th>
<th>CSM-POL</th>
<th>TMK-INT</th>
<th>TMK-POL</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock</td>
<td>11usec</td>
<td>11usec</td>
<td>976usec</td>
<td>79usec</td>
</tr>
<tr>
<td>barrier</td>
<td>208usec</td>
<td>205usec</td>
<td>5432usec</td>
<td>1213usec</td>
</tr>
<tr>
<td>pagefetch</td>
<td>1960usec</td>
<td>742usec</td>
<td>1962usec</td>
<td>784usec</td>
</tr>
<tr>
<td>fault</td>
<td>89usec</td>
<td>89usec</td>
<td>89usec</td>
<td>89usec</td>
</tr>
<tr>
<td>twin</td>
<td>N/A</td>
<td>N/A</td>
<td>362usec</td>
<td>362usec</td>
</tr>
<tr>
<td>diff</td>
<td>N/A</td>
<td>N/A</td>
<td>289-533us</td>
<td>289-533us</td>
</tr>
</tbody>
</table>
## Applications

<table>
<thead>
<tr>
<th>Program</th>
<th>Problem Size</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>3072x4096 (50M)</td>
<td>194.96</td>
</tr>
<tr>
<td>LU</td>
<td>2046x2046 (33M)</td>
<td>254.77</td>
</tr>
<tr>
<td>Water</td>
<td>4096 mols (4M)</td>
<td>1847.56</td>
</tr>
<tr>
<td>TSP</td>
<td>17cities (1M)</td>
<td>4028.95</td>
</tr>
<tr>
<td>Gauss</td>
<td>2046x2046 (33M)</td>
<td>953.71</td>
</tr>
<tr>
<td>Ilink</td>
<td>CLP (15M)</td>
<td>898.97</td>
</tr>
<tr>
<td>Em3d</td>
<td>60106 nodes (49M)</td>
<td>161.43</td>
</tr>
<tr>
<td>Barnes</td>
<td>128K bodies (26M)</td>
<td>469.43</td>
</tr>
</tbody>
</table>
Results

Speedup

University of Rochester
Computer Science
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

University of Rochester
Computer Science

Legend:
- 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