Timestamp Snooping:
An Approach for Extending SMPs

Milo M. K. Martin, Daniel J. Sorin, Anastassia Ailamaki,
Alaa R. Alameldeen, Ross M. Dickson, Carl J. Mauer,
Kevin E. Moore, Manoj Plakal, Mark D. Hill, David A. Wood

University of Wisconsin-Madison

http://www.cs.wisc.edu/multifacet/

Ninth International Conference on Architectural Support for
Programming Languages and Operating Systems (ASPLOS-IX)

November 2000
Overview

• Problem: **multiprocessors for commercial workloads**
• **Snooping (SMPs)**
  + Finds data directly - no indirection
  - Constrains interconnect
• **Goal:** Free snooping from interconnect constraints
• **Timestamps provide logical global order**
• **Evaluation vs directory protocol (CC-NUMA)**
  • Commercial workloads on 16 processors
  • 6-23% faster
  • Directories use 17-37% less bandwidth

**EXTENDING SMPs TO GENERAL INTERCONNECTS**
Outline

• Overview
• Commercial Workloads
• Traditional Coherence
• Timestamp Snooping
  • Interconnect
  • Protocol
• Evaluation
• Conclusion
Commercial Workloads

- Dominant use of multiprocessors

- Moderate processor count
  2-8, then 16-64, but not 1024

- Many cache-to-cache transfers (3-hop or dirty misses)
  - 55-62% for OLTP [Barroso et al. ISCA '98]
  - 40-60% for our commercial workloads

Design multiprocessors for commercial workloads
Traditional Snooping (SMPs)

- **Operation**
  - Requests sent on physical bus
  - Processors & memory *snoop* requests
  - Snoop responses
  - Owner responds

- **Advantages**
  + Fast cache-to-cache transfers

- **Disadvantages**
  - Bus bottleneck
  - Signaling limitations

- **Agarwal et al. (1988) predicted the demise of SMPs**
Directory Protocols (CC-NUMA)

- Add a level of indirection (for some requests)
  - Send requests to a directory
  - Directory redirects request

- Advantages
  + Avoids broadcast → scalable
  + Few interconnect restrictions

- Disadvantage
  - Directory state
  - Slow cache-to-cache transfers (3-hops)

- Example: Alpha 21364 - directory protocol with 2D torus

GAINS SCALABILITY AT THE COST OF SLOW 3-HOP TRANSACTIONS
Modern SMPs

- Many enhancements
  - Multiple buses
  - Pipelined broadcast tree with point-to-point links
- Commercially successful, few academic papers
- Challenges
  - ‘Logical bus’ → synchronous broadcast
  - Global snoop responses
  - Arbitration & flow control
- Example: Sun E10000 - 64 processors
  130 ASICs for interconnect

SMPs impose interconnect restrictions
Extending Snooping

• Key requirements
  • **Total order**
  • Broadcast

• Relax other requirements
  • No synchronous interconnect
  • Arbitrary topology (direct or indirect)
  • No snoop responses
  • No global arbitration

**Provide total order with fewer interconnect restrictions**
• **Goal:** Create a logical total order
Goal: Create a logical total order

![Diagram showing the goal of timestamp snooping]
• **Goal:** Create a logical total order
• **Goal:** Create a logical total order
• **Goal:** Create a logical total order

- **Logical Time:**
  - $P_1$: Logical Time = 3
  - $P_2$: Logical Time = 4

- **Timestamps:**
  - $R_1$: Timestamp = 5
  - $R_2$: Timestamp = 4

- **Processors:**
  - $P_a$:
  - $P_b$:
Timestamp Snooping

- **Goal:** Create a logical total order

![Diagram showing logical and timestamp ordering]

- Logical Time = 3
  - $P_a$ (Timestamp = 4)
  - $P_b$ (Timestamp = 5)

- Logical Time = 5
  - $P_a$ (Timestamp = 5)
  - $P_b$ (Timestamp = 4)
• Goal: Create a logical total order
• Goal: Create a logical total order
• Goal: Create a logical total order

Logical Time = 6

Timestamp = 5

Timestamp = 4
Logical Time

• Ordering Time (OT)
  • Arrival timestamp of request
  • Assign at source
  • Broadcast without regard to order
  • Re-order at the end-points

• Guarantee Time (GT)
  • Logical time base
  • Recursively maintained at switches

• Invariant
  • Messages delivered while $OT_{request} \geq GT_{destination}$
Uncontended Example

Single unicast request

2D torus node

source

destination

Logical Time
Uncontended Example

Assign \( OT_{\text{request}} \) at source

\[ OT_{\text{request}} = GT_{\text{source}} + \text{Distance} = 5 \]

Ordering

Time \( = 5 \)

Guarantee

Time \( = 1 \)
Uncontended Example

Ordering
Time = 5

Guarantee
Time = 2
Uncontended Example

Ordering
Time = 5

Guarantee
Time = 3
Uncontended Example

Ordering
Time = 5

Guarantee
Time = 4
Uncontended Example

Ordering
Time = 5

Guarantee
Time = 5
Interconnect Contention

• Invariant
  • Requests delivered while $\text{OT}_{\text{request}} \geq \text{GT}_{\text{destination}}$

• No contention
  • GTs always advance

• Contention
  • Recursively delay GTs to ‘warp time’
  • Prevent requests from being \textit{late}
Contention Example
Contention Example
Contention Example

contended link
Contention Example

GT advance delayed
Contention Example

delay propagates
Contetion Example

both requests 'on time'
Contention Example
Contention Example
Contention Example
Conten tion Example
Contention Example
Adding Slack

• Contention
  • GTs delayed
  • Can delay processing of other requests
  • Recursively propagates

• Contention is common
  • Avoid delaying GTs in moderate contention
  • Add \textit{slack} to initial OTs
  • Slack: extra logical time to reach destination
Slack Example

Two requests with slack
Slack Example
Slack Example

contended link
Slack Example

delayed → loses slack
Slack Example
Slack Example

Avoids disruption in common cases

'on time'

'early'
Implementation: Tokens

- Token passing implementation
  - Encode delta OTs and GTs implicitly
  - Extra bit per link
  - Small field per request
  - Simple algorithm in switches
- Advantages
  + Total order
  + Asynchronous
  + Variable link delay
- Disadvantages
  - Switch complexity

Tokens encode logical time
Timestamp Snooping Protocol

- Conventional MSI write-invalidate protocol
- Track if memory is owner
  - 1 state bit per block in memory (0.2% overhead)
  - Old idea from Synapse [Frank, 1984]
  - Avoids snoop responses
- Does not require synchronous broadcast

Extends well-accepted snooping protocols
Outline

• Commercial Workloads
• Traditional Coherence
• Timestamp Snooping
• Evaluation
  • Workloads
  • Simulated System
  • Execution Time
  • Bandwidth
• Conclusion & Future Work
Workloads

- On-line transaction processing (OLTP)
  IBM’s DB2, TPC-C like, 400 MB in-memory DB

- Decision Support System (DSS)
  IBM’s DB2, Q12 from TPC-H, 100 MB in-memory DB

- Apache - web server
  8000 static files, 160 MB total

- Altavista - search engine
  500 MB index, 160,000 pages

- Barnes - scientific benchmark
  16K bodies
Simulated System

- Extended Virtutech’s Simics full-system simulator
- 16 processors
- SPARC/Solaris 7
- Processor can execute 4 billion instructions/second including L1 cache misses

Parameters
- 4 MB, 4-way set-associative blocking L2 caches
- 64 Byte blocks

Vary protocol
- Timestamp Snooping
- DirOpt: non-blocking directory protocol

Interconnect
- 2D Torus (4x4)
- Interconnect bandwidth unconstrained
# Latency Assumptions

- **Switch-to-switch**: 15 ns
- **Enter & exit network**: 4 ns
- **DRAM/directory access**: 80 ns
- **Cache SRAM access**: 25 ns

<table>
<thead>
<tr>
<th>Directory (CC-NUMA)</th>
<th>from Memory</th>
<th>from Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 hop + DRAM</td>
<td>148 ns</td>
<td>3 hop + directory + SRAM 207 ns</td>
</tr>
<tr>
<td>TS Snoop</td>
<td>2 hop + DRAM 148 ns</td>
<td>2 hop + SRAM 93 ns</td>
</tr>
</tbody>
</table>

- For **Directory** and **TS Snoop**, latency is the same.
- For **Directory** and **TS Snoop**, latency is 2x higher.
Execution Time Results

**Timestamp Snooping is 6-23% Faster than Directories**
Bandwidth Assumptions

- Back-of-the-envelope calculation
- Data at memory
- One request, one data response
- Dependent on number of processors

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Request</th>
<th>Data Response</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory (CC-NUMA)</td>
<td>8 Bytes</td>
<td>72 Bytes</td>
<td>= 160 B</td>
</tr>
<tr>
<td>TS Snoop</td>
<td>Broadcast 15 × 8 B</td>
<td>Unicast 2 × 72 B</td>
<td>= 264 B</td>
</tr>
</tbody>
</table>

Conservative Estimate: Directories 53% less bandwidth/miss
Bandwidth Results

Directories use 17-37% less bandwidth
Conclusion

- Comparison vs directory protocols
  - Efficient cache-to-cache transfers → performance advantage
  - Latency/bandwidth trade-off

- Comparison vs current SMPs
  - More interconnect choices
  - Less global communication

- Future work
  - Multicast snooping on Timestamp Snooping network
  - Bandwidth adaptive snooping hybrid

http://www.cs.wisc.edu/multifacet/