OS Support for Virtualizing Hardware Transactional Memory


University of Wisconsin—Madison
The Virtualization Problem of HTM

What should happen to a transaction when the Operating System virtualizes a processor or memory?

1. Abort the transaction
2. Go non-speculative
3. Switch to software
4. Virtualize the transaction
Why virtualize?

1. Not virtualizing requires undoing the transaction
   - Aborts take time
   - Blocks other threads until complete
   - Example: aborting a 2048-page TX takes 1,000 µs

2. Invoking OS services may require blocking
   - Locks, reversible I/O

3. Limits generality of TM as compared to locks
Importance of Virtualization

Profile results from Sun T1000 32 thread machine, 256 MB/process

<table>
<thead>
<tr>
<th>Application</th>
<th>Locked Context Switches</th>
<th>Page Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>Voluntary</td>
</tr>
<tr>
<td>BIND 9.0</td>
<td>QueryPerf</td>
<td>1494.0</td>
</tr>
<tr>
<td>Apache 2.0</td>
<td>SpecWeb99</td>
<td>555.0</td>
</tr>
<tr>
<td>AOLServer</td>
<td>ApacheBench</td>
<td>0.1</td>
</tr>
<tr>
<td>Firefox</td>
<td>Browsing</td>
<td>12.0</td>
</tr>
<tr>
<td>OpenOffice</td>
<td>Editing</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Virtualizing HTM

• We implemented TVM, an OpenSolaris kernel module that:
  – Virtualizes a variant of LogTM-SE
  – Supports context switching and paging
  – Hooks the kernel in 9 places
  – Comprises 1120 lines of code
  – Adds less than 2% runtime overhead
Outline

• Introduction
• **Hardware Overview**
• OS support for virtualization
• Evaluation
• Conclusion
LogTM-VSE Hardware Overview

• Version management:
  – Update in place, old values logged in VM
LogTM-VSE Regular Operation

CPU 0
Thread 1
write 0x10

CPU 1
Thread 2

Memory

0x10:32
Log

Log
LogTM-VSE Hardware Overview

- **Version management:**
  - Update in place, old values logged in VM

- **Conflict detection:**
  - Read/write addresses hashed into signature
  - Coherence reqs. check signature for conflicts
LogTM-VSE Regular Operation

CPU 0
Thread 1
write 0x10
P:1000

CPU 1
Thread 2
write 0x10
P:0000

Memory

Log

0x10:32
Log
LogTM-VSE Hardware Overview

• Version management:
  – Update in place, old values logged in VM

• Conflict detection:
  – Read/write addresses hashed into signature
  – Coherence reqs. check signature for conflicts

• Virtualization
  – Adds virtual signatures for paging
  – Summary signature for conflict detection on suspended transactions
Virtualizing a Transaction

CPU 0
Thread 3
- P:0000
- V:0000
- S:1011

CPU 1
Thread 2
- P:0000
- V:0000
- S:1011

Memory

Log
0x10:32

WRONG

Thread 1
- P:0000
- V:0000
- S:1011

Log
0x10:32

V:1011
Outline

• Introduction
• Hardware Overview
• **OS support for virtualization**
  – Context switching
  – Paging
• Evaluation
• Conclusion
Virtualizing TX with TVM

- Transaction Virtualization Manager (TVM) enforces isolation of virtualized TX
  - Hooks OpenSolaris in 9 places
  - Computes & distributes summary signatures
Context Switching

• TVM hooks the OpenSolaris kernel in 4 places
  – `savectx()`: save virtual signature, distribute new summary
  – `restorectx()`: restore virtual signature
  – Complete virtual transaction: distribute new summary
  – Summary conflict: forward to contention manager
Optimizations

• Why distribute new summaries synchronously?
  – **Lazy completion** defers update on transaction complete
  – **Lazy summary update** defers update on thread suspend to reduce latency
Lazy Summary Update

CPU 0
- Thread 3
  - LSU filter
  - P: 0000
  - V: 000
  - S: 1011

CPU 1
- Thread 2
  - write 0x20
  - P: 0000
  - V: 000
  - S: 1011

Memory
- V: 1011

Log
- 0x10:32

Thread 1
- P: 1110
- V: 1011
- S: 0000

Thread 2
- P: 0000
- V: 000
- S: 0000

CPU 0
- write 0x20
- LSU filter
- P: 0000
- V: 000
- S: 1011

CPU 1
- LOG
- OK

Thread 1
- Log
- 0x10:32

Thread 3
- Log
- 0x10:32

LSU filter
- 0 1 2 3

OK

NACK

Thread 1
- V: 1011

Thread 2
- P: 0000
- V: 000
- S: 1011

Thread 3
- P: 0000
- V: 000
- S: 1011
Paging Transactional Data

- What happens when the kernel changes the address mapping for a page?
  - On paging
  - On copy-on-write

Virtualize Transactions!

- Details in paper
TVM Summary

• TVM manages summary signatures to virtualize transactions

• TVM is implemented as an OpenSolaris kernel module
  – 1120 lines of code
    • Context Switch: 325 lines
    • Paging: 265 lines
    • Common: 530 lines
  – TVM invoked from only 9 locations
Outline

- Introduction
- Hardware Overview
- OS support for virtualization
- Extending virtualization to a VMM
- Evaluation
- Conclusion
Evaluation

• Profile lock-based critical sections
  – Sun T1000 platform - Niagara (8-core 4-way SMT)

• Microbenchmark LogTM-VSE/TVM in simulation
  – GEMS/LogTM-VSE full-system simulation
    • 32 in-order SPARC cores
    • Memory latencies match T1000
    • 2048-byte signatures

• Predict overhead
  – Profiled event count × simulator measured time
Virtualization Overhead

Percent Overhead

- Bind 9.0: < 0.1%
- AOLserver: 1.0%
- Apache 2.0: 2.0%
- Firefox 2.0: 0.5%
- OpenOffice: 0.5%

Legend:
- Context Switch
- Paging
Summary

• We implemented TVM, an OpenSolaris kernel module that:
  – Supports context switching and paging
  – Hooks the kernel in 9 places
  – Comprises 1120 lines of code
  – Adds less than 2% runtime overhead

• Our design supports execution in a VMM and virtualization by a VMM
Questions?

For more information:
swift@cs.wisc.edu
www.cs.wisc.edu/multifacet/logtm