DIMM

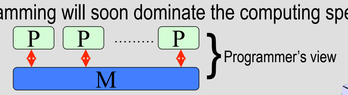
Architecture Support for Data Isolation & Memory Monitoring

Arrvindh Shriraman, Sandhya Dwarkadas, and Michael L. Scott

Department of Computer Science, University of Rochester

Motivation

- Multi-core processors based on shared memory programming will soon dominate the computing spectrum



- Coordinating and synchronizing data shared across multiple threads is hard!

- Tracking memory location accesses is difficult because of transparent coherence events
- Cannot issue speculative operations to memory because hardware protocol does not support undoing of writes

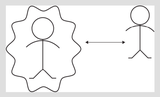
Shared Memory ++

- **Memory Monitoring (MM)**
  - ➡ provides read/write access summaries of code blocks
  - ➡ event-style notification of desired coherence events

Apps: Reliability, Security, Watchpoints, and Debugging

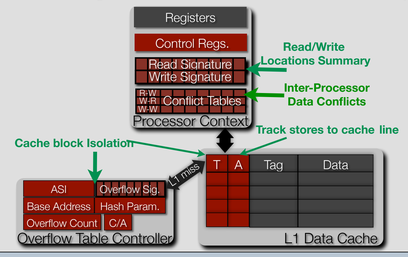
- **Data Isolation (DI)**
  - ➡ allows control over propagation of writes to remote threads
  - ➡ buffer written locations and commit or undo as an atomic unit

Apps: Sand-boxing, Transactional programming, Speculation



DIMM Hardware Support

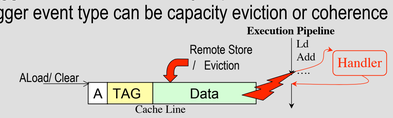
- Decoupled hardware primitives for DIMM help
  - ➡ refine architecture incrementally
  - ➡ software evolve the API and use in varying applications
  - ➡ decouple policy from mechanism
- Memory Monitoring primitives
  - ➡ **Alert-On-Update**: precise but bounded size
  - ➡ **Signatures**: imprecise but unbounded
  - ➡ **CST:** track inter-processor conflicts for all watched locations
- Data Isolation primitives
  - ➡ **PDI:** private caches←→speculative-write buffer
  - ➡ **Redo-Log:** holds cache overflows in virtual memory



Memory Monitoring

Alert-On-Update (AOU)

- New instruction, **ALoad,** loads and marks cache line
- A-tagged line on invalidation jumps to handler
  - ➡ trigger event type can be capacity eviction or coherence



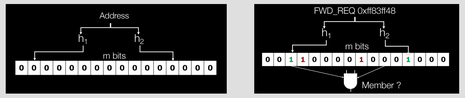
Access summary Signatures

- Insert addresses accessed by thread in hardware bloom filters. (Reads update R_{sig} & Writes update W_{sig})

\+ unboundedness, decouples tracking from caches

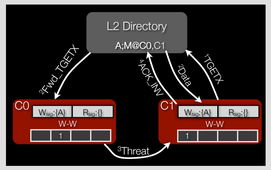
\- false positives

- Special instructions access cache blocks and insert physical address into bloom filter
- Coherence requests snoop signatures, test for membership and piggy-back conflict type on response message



Conflict Summary Tables (CST)

- Record inter-processor R-W, W-W & R-R conflicts
- Decouples access conflict tracking from access tracking



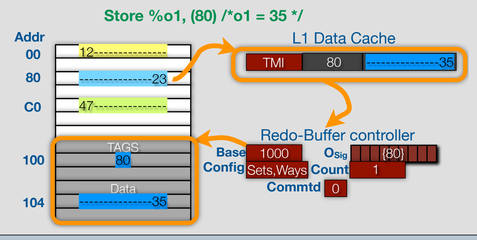
Data Isolation

Lazy Coherence

- Caches detach lines selectively from coherence protocol
  - ➡ track coherence messages and choose time to enforce rules
- Cache protocol extended by two 'T' bit tagged states
  - ➡ **TMI** buffers TStores; **TI** allows incoherence with remote TMI
- **TMI** allows concurrent sharers & isolates data in cache
- TMI & TI require just a flash-clear to convert lines to MESI

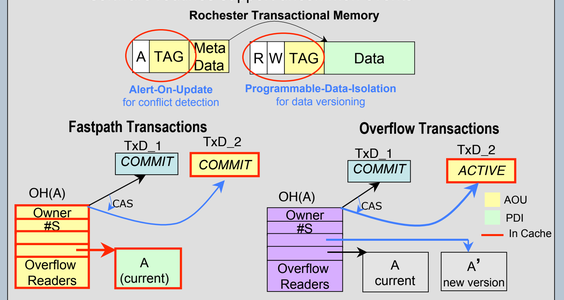
Redo-Buffer

- A per-thread hash-table in virtual memory
- Hardware controller
  - ➡ fills table with "TMI" write-back data blocks
  - ➡ performs look-aside transparently on L1 misses



RTM [ISCA'07]

- Integrated Hardware-Software approach to flexible transactional memory
- Hardware-Software transactions
  - ➡ DIMM mechanisms accelerate common STM operations
  - ➡ software makes policy decisions
  - ➡ software routines support uncommon events



FlexTM [ISCA'08]

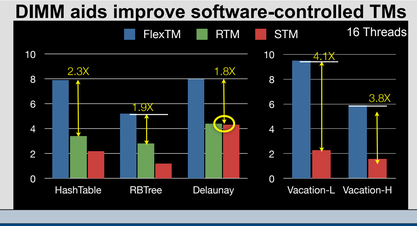
- FlexTM deploys
  - ➡ **Signatures** for detecting and notifying conflicts
  - ➡ **CSTs** for noticing and managing conflicts
  - ➡ **Lazy caches** for in-cache data isolation and **Redo-Buffer** for handling cache overflows
  - ➡ **AOU** for propagating abort events to remote transactions
- FlexTM software
  - ➡ checkpoints registers at Begin_Tx
  - ➡ manages conflicts; controls Tx aborts using AOU trigger
  - ➡ controls commit phase

```
Begin_Tx  abort_pc
TLD A
TST B
.........
Foreach I set in W-R or W-W
    CAS (Status[i], ACT, ABORT)
CAS-Commit  Status[id]
```

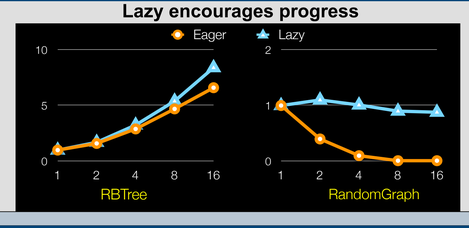
- Checkpoint processor registers and record abort handler PC
- Issue TLoad/TStore for speculative memory operations
- Iterate over CSTs and update status word of conflicting transactions
- Logically commit on status word; start physical commit of hardware state

Hardware-acceleration of Software-controlled transactions

DIMM aids improve software-controlled TMs



Lazy encourages progress



FlexWatcher Memory Debugger

- Extend ISA to support signatures and AOU as first-class entities
  - ➡ insert,member,activate,clear etc
- Compiler/ Programmer specifies addresses to be tracked
- Hardware triggers trampoline on snoop hits

Benchmark	Bug	FlexWatcher	Discover
BC	BO	1.5X	75X
GZIP	BO	1.15X	17X
GZIP[2]	IV	1.05X	N/A
Man	BO	1.80X	65X
Squid	ML	2.50X	N/A

Discover is a SPARC binary instrumentation tool from OpenSPARC
Discover overheads were estimated on a Sun T1000 server

Buffer Overflow (BO)
Pad all heap allocated buffers with 64bytes, watch padded locations

Memory Leak (ML)
Monitor all heap allocated objects and update the address's timestamp on access.

Invariant Violation (IV)
ALoad cache line for variable X of interest. On AOU handler trigger assert program specific invariants.

Other Uses

- **Synchronization**
  - ➡ fast mutexes and asynchronous messages
- **Debugging**
  - ➡ watchpoints and race detectors
- **Security**
  - ➡ buffer overflow attacks, information-flow trackers & drivers/plugin isolation
- **Speculation**
  - ➡ Thread-level speculation and lock elision

Conclusion

- Data-Isolation and Memory-Monitoring primitives will help multi-core chips achieve widespread use across traditional and emerging application domains
- Decoupling the hardware components will help refine the architecture incrementally and help software evolve the API
- Use simple hardware to accelerate the common case, minimize hardware state and employ software for the uncommon case

Web : http://www.cs.rochester.edu/research/cosyn/

Email: {ashriram, sandhya, scott}@cs.rochester.edu