**Architecture Support for Data Isolation & Memory Monitoring**

Arvind 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
  - Programmer’s view
- 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 and 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: impure but unbounded
  - CST: track inter-processor conflicts for all watched locations
- Data Isolation primitives
  - PDI: private caches speculate-write buffer
  - Redo-Log: holds cache overflows in virtual memory

**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, architecture incrementally and help software evolve the API

**Other Uses**
- Synchronization: fast mutexes and asynchronous messages
- Debugging: watchpoints and race detectors
- Security: buffer overflow attacks, information-flow trackers & drivers/plug-in isolation
- Speculation: thread-level speculation and lock elision

**Web**: [http://www.cs.rochester.edu/research/cosyn/](http://www.cs.rochester.edu/research/cosyn/)

**Email**: ashiriram, sandhya, scott)@cs.rochester.edu