

# Evolving Beyond Transactions?

Michael L. Scott

University of Rochester

Thoughts for closing panel  
TRANSACTION '07

- But we aren't even there yet!
- Maybe how we *get* there will help us move beyond

# TM Design Philosophies

- All SW
  - » needed for legacy HW anyway
  - » *might* turn out to be enough
  - » HW GC and type checking were premature
- Best-effort HW
  - » fast most of the time
  - » correct all the time
  - » “merely correct” part runs on legacy machines
- All HW
  - » arguably fastest, and more predictable:
  - » avoid anomalies due to SW fallback
  - » for multiple policies, parameterize the HW

# A Different Kind of Hybrid

**RTM** [TR Dec.'05, TRANSACT '06,  
PPoPP '07 poster, SPAA '07, ISCA '07]

- Leave policy in SW; accelerate with HW
  - » decouple conflict detection from speculative writes
  - » perform the latter on “real” data, former on metadata
- AOU jumps to handler on conflicting metadata write
- PDI permits cache inconsistency under SW control
  - » enable lazy resolution of RW and even WW conflicts (as in RSTM, FSTM, and TL2)

# Policy/Mechanism Split

- Maximizes freedom in programming model design
- Avoids pitfall of putting wrong policy in HW
- Facilitates dynamic tuning/adaptation
- May increase opportunities to use the HW for other purposes
- But metadata management is costly — ~2X hit compared to HTM

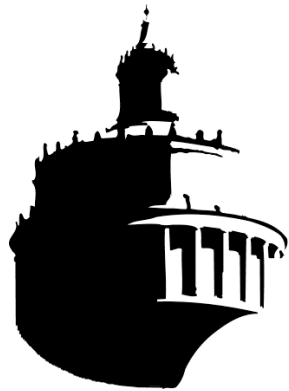
# FlexTM

- (recent work; not yet published: Arrvindh Shriraman and Sandhya Dwarkadas)
- Do conflict detection on real data, but decouple from conflict *tracking and resolution* (and from buffering of speculative state)
- HW read / write signatures for every processor
- Additional bit vectors to record conflicts with other processors
- SW can choose to trap to handler on conflict or to poll the vectors later (e.g. at commit, which remains *local*)
- All structures visible in main memory, so OS can virtualize
  - » Vectors that summarize signatures of descheduled threads

# Other Uses

(See PPopP poster & TR version of our ISCA paper)

- AOU for
  - » fast mutexes
  - » active messages
  - » rollback in ad hoc nonblocking algorithms
  - » ABA avoidance
  - » debugger watchpoints
  - » misc. security and fault-containment mechanism
- PDI for ordered thread-level speculation  
(for performance or reliability)
- CSTs? not sure yet



UNIVERSITY OF  
**ROCHESTER**  
COMPUTER SCIENCE

[www.cs.rochester.edu/research/synchronization/](http://www.cs.rochester.edu/research/synchronization/)

# PPoPP'08

The 13th ACM SIGPLAN Symposium on  
Principles and Practice of Parallel Programming

---

20–23 February 2008

Salt Lake City, Utah (co-located with [HPCA-14](#))

Submission deadline:  
13 (abstracts) / 20 Aug. 2007

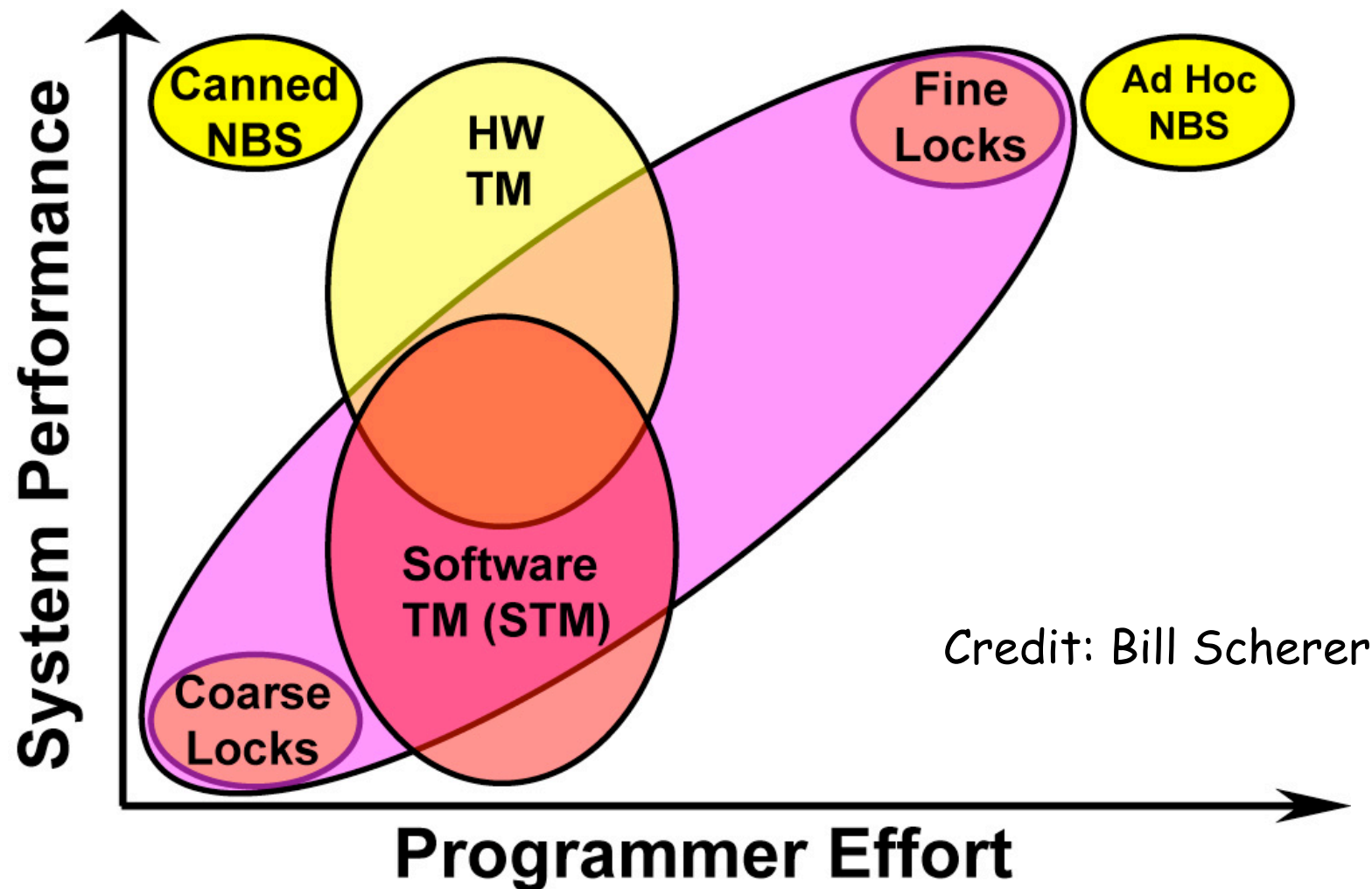
[www.ppop.org](http://www.ppop.org)







# Concurrency Design Space



# Transactional Sharing Models

- Contract between the user & the system
  - » Cf. programmer-centric memory consistency models
  - » ideally enforced by compiler
- Transactions *appear* to be strongly isolated if programmer follows the rules
  - » static partition — **too restrictive**
  - » partition within global consensus phases
    - e.g. via barriers
  - » privatizing transactions
    - multiple possible implementations
  - » strong isolation
    - probably too expensive for software — **overkill**