



Transactional Programming In A Multi-core Environment



Ali-Reza Adl-Tabatabai

Intel Corp.

Christos Kozyrakis

Stanford U.

Bratin Saha

Intel Corp.

PACT-2006 Tutorial, September 17th, 2006

Presenters

- **Ali-Reza Adl-Tabatabai**, Intel

- Principal Engineer, Programming Systems Lab at Intel
- Compilers & runtimes for future Intel architectures
- PhD. CMU

- **Christos Kozyrakis**, Stanford University

- Assistant Professor, Electrical Eng. & Computer Science
- Architectures & programming models for transactional memory
- Ph.D. UC Berkeley

- **Bratin Saha**

- Senior Staff Researcher, Programming Systems Lab
- Design of highly scalable runtimes for multi-core processors
- Ph.D. Yale



Tutorial Motivation & Goals

○ Motivation

- Transactions are a good synchronization abstraction
- How can transactions be implemented and used ?

○ Goals

1. Introduction to transactional memory
 - A research technology for easier parallel programming
 - Overview, uses, and implementation



Agenda



- Transactional Memory (TM)
 - TM Introduction
 - TM Implementation Overview
 - Hardware TM Techniques
 - Software TM Techniques
- Q&A

TM Bibliography

- Active, online bibliography at <http://www.cs.wisc.edu/trans-memory>



STM & TM Languages References

1. N. Shavit and S. Touitou. Software Transactional Memory. In *Proc. of the 14th Symposium on Principles of Distributed Computing*, Aug. 1995.
2. M. Herlihy et al. Software Transactional Memory for Dynamic-sized Data Structures. In *Proc of the 22nd Symposium on Principles of Distributed Computing*, July 2003.
3. T. Harris and K. Fraser. Language Support for Lightweight Transactions. In *Proc. of the 18th Conference on Object-oriented Programming, Systems, Languages, and Applications*, 2003.
4. V. Marathe, W. Scherer, and M. Scott. Adaptive Software Transactional Memory. In *Proc. of the 19th Intl. Symposium on Distributed Computing*, Sept. 2005.
5. P. Charles et al. X10: An Object-oriented Approach to Nonuniform Cluster Computing. *Proc. of the 20th Conference on Object-oriented Programming, Systems, Languages, and Applications*. Oct. 2005.
6. E. Allen et al. *The Fortress Language Specification*. Sun Microsystems, 2005.
7. *Chapel Specification*. Cray, February 2005.
8. Hudson et al. McRT-Malloc: A Scalable Transaction Aware Memory Allocator. ISMM 2006
9. B. D. Carlstrom et al. The Atomos Transactional Programming Language. In *Proc. of the Conference on Programming Language Design and Implementation*, June 2006.
10. N. Shavit and D. Dice, What Really Makes Transactions Faster. Workshop on Languages, Compilers, and Hardware Support for Transactional Computing, June 2006.
11. B. Saha et al. Implementing a high performance software transactional memory. In *Proc. of the Conference on Principles and Practices of Parallel Processing*, March, 2006
12. A. Adl-Tabatabai et al, Compiler and runtime support for efficient software transactional memory. In *Proc of the Conference on Programming Language Design and Implementation*, June 2006.
13. T. Harris et al. Optimizing Memory Transactions. In *Proc of the Conference on Programming Language Design and Implementation*, June 2006.



HTM & Hybrid-TM References

1. T. Knight. An Architecture for Mostly Functional Languages. In *Proc. of the ACM Conference on LISP and Functional Programming*, 1986.
2. M. Herlihy and J. E. B. Moss. Transactional Memory: Architectural Support for Lock-Free Data Structures. In *Proc. of the 20th Annual Intl. Symp. on Computer Architecture*, May 1993
3. R. Rajwar and J. R. Goodman. Transactional Lock-Free Execution of Lock-Based Programs. In *Proc. of the 10th Intl. Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 2002.
4. L. Hammond, et al. Transactional Memory Coherence and Consistency. In *Proc. Of the 31st Annual Intl. Symp. on Computer Architecture*, June 2004.
5. L. Hammond, et. al. Programming with transactional coherence and consistency. In *Proc. of the 11th Intl. Conference on Architecture Support for Programming Languages and Operating Systems*, Oct. 2004.
6. S. Ananian et al. Unbounded Transactional Memory. In *Proc. of the 11th Intl. Symposium on High Performance Computer Architecture*, Feb. 2005.
7. R. Rajwar, M. Herlihy, and K. Lai. Virtualizing Transactional Memory. In *Proc. of the 32nd Annual Intl. Symp. On Computer Architecture*, Jun. 2005.
8. C. Blundell, et al. Deconstructing Transactional Semantics: The Subtleties of Atomicity. In *ISCA Workshop on Duplicating, Deconstructing, and Debunking*, June 2005.



HTM & Hybrid-TM References

9. A. McDonald et al. Characterization of TCC on Chip-Multiprocessors. In *Proc. of the 14th Intl. Conference on Parallel Architectures and Compilation Techniques*, Sept. 2005.
10. K. Moore et al. LogTM: Log-Based Transactional Memory. In *Proc. of the 12th Intl. Conference on High Performance Computer Architecture*, Feb. 2006.
11. S. Kumar et al. Hybrid Transactional Memory. In *Proc. of the Conference on Principles and Practices of Parallel Processing*, March, 2006
12. J. Chung et al. The Common Case Transactional Behavior of Multithreaded Programs. In *Proc. of the 12th Intl. Conference on High Performance Computer Architecture*, Feb. 2006.
13. A. Sriraman et al. Hardware Acceleration of hardware Transactional Memory, Workshop on Languages, Compilers, and Hardware Support for Transactional Computing, June 2006.
14. L. Ceze et al. Bulk Disambiguation of Speculative Threads in Multiprocessors, In *Proc. of the 32nd Intl. Symposium on Computer Architecture*, June 2006.
15. A. McDonald et al. Architectural Semantics for Practical Transactional Memory, In *Proc. of the 32nd Intl. Symposium on Computer Architecture*, June 2006.
16. P. Damron et al. Hybrid Transactional Memory, In *Proc. Of the 12th Intl. Conference on Architecture Support for Programming Languages and Operating Systems*, Oct. 2006.
17. J. Chung et al. Tradeoffs in Transactional Memory Virtualization , In *Proc. of the 12th Intl. Conference on Architecture Support for Programming Languages and Operating Systems*, Oct. 2006.

