



























- [6] M. Abadi et al. Semantics of transactional memory and automatic mutual exclusion. In *POPL*, 2008.
- [7] M. Abadi et al. Transactional memory with strong atomicity using off-the-shelf memory protection hardware. In *PPoPP*, 2009.
- [8] A.-R. Adl-Tabatabai et al. Compiler and runtime support for efficient software transactional memory. In *PLDI*, 2006.
- [9] Y. Afek et al. Lowering STM overhead with static analysis. In *LCPC*, 2010.
- [10] M. D. Allen et al. Serialization sets: A dynamic dependence-based parallel execution model. In *PPOPP*, 2009.
- [11] A. Aviram et al. Efficient system-enforced deterministic parallelism. 2010.
- [12] N. E. Beckman et al. Verifying correct usage of atomic blocks and tystate. In *OOPSLA*, 2008.
- [13] N. E. Beckman et al. Reducing STM overhead with access permissions. In *IWACO*, 2009.
- [14] T. Bergan et al. CoreDet: A compiler and runtime system for deterministic multithreaded execution. In *Int'l. Conf. on Arch. Support for Programming Langs. and Operating Sys. (ASPLOS)*, 2010.
- [15] E. D. Berger et al. Grace: Safe Multithreaded Programming for C/C++. In *OOPSLA*, 2009.
- [16] R. D. Blumofe et al. Cilk: An efficient multithreaded runtime system. *PPOPP*, 1995.
- [17] R. L. Bocchino et al. A type and effect system for Deterministic Parallel Java. In *OOPSLA*, 2009.
- [18] R. L. Bocchino et al. Parallel programming must be deterministic by default. In *HotPar*, 2009.
- [19] R. L. Bocchino Jr. *An Effect System and Language for Deterministic-by-Default Parallel Programming*. PhD thesis, University of Illinois, Urbana-Champaign, IL, 2010.
- [20] C. Boyapati et al. Ownership types for safe programming: Preventing data races and deadlocks. In *OOPSLA*, 2002.
- [21] J. Boyland. Checking interference with fractional permissions. *SAS*, 2003.
- [22] N. G. Bronson et al. Feedback-directed barrier optimization in a strongly isolated STM. In *POPL*, 2009.
- [23] Z. Budimlic et al. Multicore implementations of the concurrent collections programming model. In *CPC*, 2009.
- [24] S. Burckhardt et al. Concurrent programming with revisions and isolation types. In *OOPSLA*, 2010.
- [25] M. J. Carey et al. A status report on the OO7 OODBMS benchmarking effort. In *OOPSLA*, 1994.
- [26] L. P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *SCG*, 1993.
- [27] J. Devietti et al. DMP: Deterministic Shared Memory Multiprocessing. In *ASPLOS*, 2009.
- [28] D. Dice et al. Transactional locking II. In *DISC*, 2006.
- [29] C. Flanagan et al. Types for atomicity: Static checking and inference for Java. *TOPLAS*, 2008.
- [30] A. Ghuloum et al. Ct: A flexible parallel programming model for tera-scale architectures. Intel White Paper, 2007.
- [31] T. Harris et al. Composable memory transactions. In *PPoPP*, 2005.
- [32] T. Harris et al. Optimizing memory transactions. In *PLDI*, 2006.
- [33] T. Harris and K. Fraser. Language support for lightweight transactions. In *OOPSLA*, 2003.
- [34] B. Jacobs et al. A programming model for concurrent object-oriented programs. *TOPLAS*, 2008.
- [35] A. Kulkarni et al. Task types for pervasive atomicity. In *OOPSLA*, 2010.
- [36] M. Kulkarni et al. Optimistic parallelism requires abstractions. In *PLDI*, 2007.
- [37] J. Larus and R. Rajwar. *Transactional Memory (Synthesis Lectures on Computer Architecture)*. Morgan & Claypool Publishers, 2007.
- [38] E. A. Lee. The problem with threads. *Computer*, 2006.
- [39] R. Lublinerman et al. Parallel programming with object assemblies. In *OOPSLA*, 2009.
- [40] M. Martin, C. Blundell, and E. Lewis. Subtleties of transactional memory atomicity semantics. *IEEE Comp. Arch. Letters*, 5(2):17, 2006.
- [41] N. D. Matsakis and T. R. Gross. A time-aware type system for data-race protection and guaranteed initialization. In *OOPSLA*, 2010.
- [42] K. F. Moore and D. Grossman. High-level small-step operational semantics for transactions. In *POPL*, 2008.
- [43] M. Olszewski et al. Kendo: Efficient deterministic multithreading in software. In *ASPLOS*, 2009.
- [44] C. Papadimitriou. *The theory of database concurrency control*. Computer Science Press, Inc., 1986.
- [45] M. C. Rinard and M. S. Lam. The design, implementation, and evaluation of Jade. *TOPLAS*, 1998.
- [46] F. T. Schneider, V. Menon, T. Shpeisman, and A.-R. Adl-Tabatabai. Dynamic optimization for efficient strong atomicity. In *OOPSLA*, 2008.
- [47] T. Shpeisman et al. Enforcing isolation and ordering in STM. In *PLDI*, 2007.
- [48] M. Vakilian et al. Inferring Method Effect Summaries for Deterministic Parallel Java. Technical Report UIUCDCS-R-2009-3038, U. Illinois, 2009.
- [49] C. von Praun et al. Implicit parallelism with ordered transactions. In *PPOPP*, 2007.
- [50] A. Welc et al. Safe futures for Java. In *OOPSLA*, 2005.
- [51] A. Welc et al. Revocation techniques for Java concurrency. *Concurrency and Computation: Practice and Experience*, 2006.
- [52] R. M. Yoo et al. Kicking the tires of software transactional memory: Why the going gets tough. In *SPAA*, 2008.