

REFERENCES

- [1] Umut A. Acar. 2005. *Self-Adjusting Computation*. Ph.D. Dissertation. Department of Computer Science, Carnegie Mellon University.
- [2] Umut A. Acar, Daniel Anderson, Guy E. Blelloch, Laxman Dhulipala, and Sam Westrick. 2020. Parallel Batch-dynamic Trees via Change Propagation. In *European Symposium on Algorithms (ESA)*.
- [3] Umut A. Acar, Guy E. Blelloch, Matthias Blume, Robert Harper, and Kanat Tangwongsan. 2009. An experimental analysis of self-adjusting computation. *ACM Trans. Program. Lang. Syst. (TOPLAS)* 32, 1 (2009), 1–53.
- [4] Umut A. Acar, Guy E. Blelloch, and Robert Harper. 2002. Adaptive functional programming. In *ACM Symposium on Principles of Programming Languages (POPL)*.
- [5] Umut A. Acar, Guy E. Blelloch, and Robert Harper. 2003. Selective Memoization. In *ACM Symposium on Principles of Programming Languages (POPL)*.
- [6] Umut A. Acar, Guy E. Blelloch, Robert Harper, Jorge L. Vites, and Shan Leung Maverick Woo. 2004. Dynamizing static algorithms, with applications to dynamic trees and history independence. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*.
- [7] Umut A. Acar, Guy E. Blelloch, Ruy Ley-Wild, Kanat Tangwongsan, and Duru Türkoğlu. 2010. Traceable Data Types for Self-Adjusting Computation. In *Programming Language Design and Implementation (PLDI)*.
- [8] Umut A. Acar, Guy E. Blelloch, Kanat Tangwongsan, and Duru Türkoğlu. 2008. Robust Kinetic Convex Hulls in 3D. In *European Symposium on Algorithms (ESA)*.
- [9] Umut A. Acar, Guy E. Blelloch, Kanat Tangwongsan, and Jorge L. Vites. 2006. Kinetic Algorithms via Self-Adjusting Computation. In *European Symposium on Algorithms (ESA)*.
- [10] Umut A. Acar, Andrew Cotter, Benoît Hudson, and Duru Türkoğlu. 2010. Dynamic Well-Spaced Point Sets. In *Symposium on Computational Geometry (SoCG)*.
- [11] Umut A. Acar, Andrew Cotter, Benoît Hudson, and Duru Türkoğlu. 2011. Parallelism in Dynamic Well-Spaced Point Sets. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.
- [12] Umut A. Acar, Benoît Hudson, and Duru Türkoğlu. 2011. Kinetic Mesh-Refinement in 2D. In *Symposium on Computational Geometry (SoCG)*.
- [13] Umut A. Acar, Alexander Ihler, Ramgopal Mettu, and Özgür Sümer. 2007. Adaptive Bayesian Inference. In *Neural Information Processing Systems (NIPS)*.
- [14] Pramod Bhatotia, Pedro Fonseca, Umut A. Acar, Björn B Brandenburg, and Rodrigo Rodrigues. 2015. iThreads: A threading library for parallel incremental computation. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*.
- [15] Pramod Bhatotia, Alexander Wieder, Rodrigo Rodrigues, Umut A. Acar, and Rafael Pasquin. 2011. Incoop: MapReduce for Incremental Computations. In *ACM Symposium on Cloud Computing (SoCC)*.
- [16] Guy E. Blelloch. 1996. Programming Parallel Algorithms. *Commun. ACM* 39, 3 (March 1996).
- [17] Guy E. Blelloch, Daniel Anderson, and Laxman Dhulipala. 2020. ParlayLib-A Toolkit for Parallel Algorithms on Shared-Memory Multicore Machines. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.
- [18] Guy E. Blelloch, Jeremy T. Fineman, Yan Gu, and Yihan Sun. 2020. Optimal parallel algorithms in the binary-forking model. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.
- [19] Richard P. Brent. 1974. The parallel evaluation of general arithmetic expressions. *J. ACM* 21, 2 (1974), 201–206.
- [20] Sebastian Burckhardt, Daan Leijen, Caitlin Sadowski, Jaeheon Yi, and Thomas Ball. 2011. Two for the Price of One: A Model for Parallel and Incremental Computation. In *ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*.
- [21] Yan Chen, Umut A. Acar, and Kanat Tangwongsan. 2014. Functional Programming for Dynamic and Large Data with Self-Adjusting Computation. In *International Conference on Functional Programming (ICFP)*.
- [22] Yan Chen, Joshua Dunfield, and Umut A. Acar. 2012. Type-Directed Automatic Incrementalization. In *Programming Language Design and Implementation (PLDI)*.
- [23] Yan Chen, Joshua Dunfield, Matthew A. Hammer, and Umut A. Acar. 2011. Implicit Self-Adjusting Computation for Purely Functional Programs. In *International Conference on Functional Programming (ICFP)*.
- [24] Alan Demers, Thomas Reps, and Tim Teitelbaum. 1981. Incremental evaluation of attribute grammars with application to syntax directed editors. In *ACM Symposium on Principles of Programming Languages (POPL)*.
- [25] P. F. Dietz and D. D. Sleator. 1987. Two algorithms for maintaining order in a list. In *ACM Symposium on Theory of Computing (STOC)*.
- [26] Jason Evans. 2006. A scalable concurrent malloc (3) implementation for FreeBSD. In *Proceedings of the BSDCan conference, Ottawa, Canada*.
- [27] Mingdong Feng and Charles E. Leiserson. 1999. Efficient detection of determinacy races in Cilk programs. *Theory of Computing Systems* 32, 3 (1999), 301–326.
- [28] Matteo Frigo, Pablo Halpern, Charles E. Leiserson, and Stephen Lewin-Berlin. 2009. Reducers and other Cilk++ hyperobjects. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.
- [29] Matthew Hammer, Umut A. Acar, Mohan Rajagopalan, and Anwar Ghuloum. 2007. A proposal for parallel self-adjusting computation. In *ACM SIGPLAN Workshop on Declarative Aspects of Multicore Programming (DAMP)*. ACM.
- [30] Matthew Hammer, Georg Neis, Yan Chen, and Umut A. Acar. 2011. Self-Adjusting Stack Machines. In *ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*.
- [31] Matthew A. Hammer, Umut A. Acar, and Yan Chen. 2009. CEAL: a C-Based Language for Self-Adjusting Computation. In *Programming Language Design and Implementation (PLDI)*.
- [32] Matthew A. Hammer, Khoo Yit Phang, Michael Hicks, and Jeffrey S. Foster. 2014. Adapton: Composable, demand-driven incremental computation. In *Programming Language Design and Implementation (PLDI)*.
- [33] W. Pugh and T. Teitelbaum. 1989. Incremental computation via function caching. In *ACM Symposium on Principles of Programming Languages (POPL)*.