

Mike Rainey

me@mike-rainey.site

<http://mike-rainey.site>

Education

2010	Ph.D.	Computer Science	University of Chicago, USA
2007	M.S.	Computer Science	University of Chicago, USA
2004	B.S.	Computer Science	Indiana University, USA
2004	B.S.	Cognitive Science	Indiana University, USA

Employment

2014	Researcher at INRIA Rocquencourt in Gallium / DeepSea (to present)
2010	Postdoctoral researcher at Max Planck Institute for Software Systems
2007	Summer internship with Anwar Gluloum at Intel Corp.
2006	Programming project. Extended the MLRISC code generator to support the AMD64. Supported by NSF Grant CRI: Standard ML Software Infrastructure. Principle investigators: David MacQueen and John H. Reppy.
2003	Research assistant at Indiana University with David S. Wise. Designed and evaluated performance of cache-aware and multiprocessor matrix-factoring algorithms. Supported by NSF Grant CRI: A Paradigm of Parallel Programming for Morton Ordered Matrices.

Publications

Conferences, Journals & Workshops

- **Contention in Structured Concurrency: Provably Efficient Dynamic NonZero Indicators for Nested Parallel Computation** PPOPP 2017 [1]
- **Dag-Calculus: A Calculus for Parallel Computation** ICFP 2016 [10]
- **Oracle-Guided Scheduling for Controlling Granularity in Implicitly Parallel Languages** JFP 2016 [6].
- **A Work-efficient Algorithm for Parallel Unordered Depth-first Search** Supercomputing 2015 [2]
- **Theory and Practice of Chunked Sequences** ESA 2014 [9]
- **Scheduling Parallel Programs by Work Stealing with Private Deques** PPOPP 2013 [8]
- **Data-only Flattening for Nested Data Parallelism** PPOPP 2013 [12]
- **Lazy Tree Splitting** JFP 2012 [11]
- **Oracle Scheduling: Controlling Granularity in Implicitly Parallel Languages** OOPSLA 2011 [5]
- **Efficient Primitives for Creating and Scheduling Parallel Computations** DAMP 2012 [4]
- **Lazy Tree Splitting** ICFP 2010 [13]
- **Implicitly threaded parallelism in Manticore** JFP 2010 [17]
- **A Scheduling Framework for General-purpose Parallel Languages** ICFP 2008 [16]
- **Implicitly-threaded Parallelism in Manticore** ICFP 2008 [18]
- **Calling Variadic Functions from a Strongly-typed Language** ML Workshop 2008 [14]
- **Manticore: A Heterogeneous Parallel Language** DAMP 2007 [19]
- **A Paradigm for Parallel Matrix Algorithms: Scalable Cholesky** EuroPar 2005 [21]

Books

- **An Introduction to Parallel Computing in C++** Online course materials [3]
- **Effective Scheduling Techniques for High-Level Parallel Programming Languages** PhD Dissertation [20]
- **Semantics Engineering with PLT Redex** Contributed chapter [15]

Not reviewed, arXiv

- **Parallel Work Inflation, Memory Effects, and their Empirical Analysis** 2017 [7]

Software artifacts

- **Manticore** This compiler and runtime system supports an experimental parallel, functional programming language aimed at general-purpose applications running on multi-core processors. I contributed code to the compiler and runtime system, with particular focus on load balancing for parallel workloads and on code generation.
 - <http://manticore.cs.uchicago.edu/>
- **Chunkedseq** This software package features a C++ template library which implements ordered, in-memory containers that are based on the B-tree-like data structure I coauthored with Umut Acar and Arthur Charguéraud [9].
 - <http://deepsea.inria.fr/chunkedseq>
- **PASL** (Parallel Algorithm Scheduling Library) This library implements the work-stealing algorithm that I coauthored with Umut Acar and Arthur Charguéraud [8].
 - <http://deepsea.inria.fr/pasl/>
- **SML/NJ** (Standard ML of New Jersey) I primarily contributed to the back end of the compiler. My main projects covered code generation for the x86_64 and support for foreign-function calls.
 - <http://smlnj.org/>
- **PDFS** (Parallel DFS) This C++ package implements the fast DFS-like graph-traversal algorithm from our SC'15 paper [2].
 - <http://deepsea.inria.fr/graph>

Professional service

Program committee member

2016	FHCP ACM SIGPLAN Workshop on Functional High-Performance Computing
2015	ICFP ACM SIGPLAN International Conference on Functional Programming
2014	ECOOP European Conference on Object-Oriented Programming Artifact Evaluation Committee
2013	FHCP ACM SIGPLAN Workshop on Functional High-Performance Computing

Reviewer for

- **ICFP** ACM SIGPLAN International Conference on Functional Programming
- **PLDI** ACM SIGPLAN Programming Language Design and Implementation
- **PPoPP** ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming
- **POPL** ACM SIGPLAN Symposium on Principles of Programming Languages
- **SPAA** ACM Symposium on Parallelism in Algorithms and Architectures
- **TOPLAS** ACM Transactions on Programming Languages and Systems
- **JFP** Journal of Functional Programming
- **ESOP** European Joint Conferences on Theory and Practice of Software
- **Euro-Par** International Conference on Parallel Programming

Teaching

Course materials

In 2014, Umut Acar and I authored course materials on multicore computing. We were motivated by a gap between the teaching of theory and practice of parallel computing that we believe existed in the undergraduate algorithms course at Carnegie Mellon University, where Acar teaches. We were also motivated by

the desire to hone an automatic granularity-control technique that we developed as part of our research collaboration [6].

In 2014, we used the course materials to teach a two-week, NSF-funded, undergraduate course at the University of Puerto Rico PASL Workshop. Later, in 2014, we introduced updated materials into the undergraduate algorithms course at Carnegie Mellon University. I served as guest lecturer in 2014 and 2015, and since then, this material has become a permanent component of the course. The course materials are available on the course website.

Teaching assistantships

2009	Teaching Assistant, Implementation of Computer Languages - II, University of Chicago, Spring
2007	Teaching Assistant, CSPP Networks, University of Chicago, Winter
2006	Teaching Assistant, Introduction to Computer Systems, University of Chicago, Spring
	Teaching Assistant, CSPP Unix Systems Programming, University of Chicago, Fall

Selected talks

- **A Work-efficient Algorithm for Parallel Unordered Depth-first Search**, Supercomputing, 2016
 - video at <https://youtu.be/Nr2s1-fds3w>
 - slides at <http://gallium.inria.fr/~rainey/sc15-pdfs-talk.pdf>
- **Scheduling Parallel Programs by Work Stealing with Private Deques**, PPOPP, 2013
 - slides at <http://gallium.inria.fr/~rainey/ppopp2013.pdf>
- **Higher-level Implicit Parallelism with PASL**, LAME Workshop, 2013
 - slides at <http://gallium.inria.fr/~rainey/lame2013.pdf>
- **Fork-join Model and Work stealing**, MPI-SWS weekly seminar, 2011
 - slides at http://gallium.inria.fr/~rainey/reading_group_slides.pdf

All publications with links

- [1] Acar, U.A. et al. 2017. Contention in structured concurrency: Provably efficient dynamic nonzero indicators for nested parallel computation. (2017).
- [2] Acar, U.A. et al. 2015. A work-efficient algorithm for parallel unordered depth-

first search. *Proceedings of the international conference for high performance computing, networking, storage and analysis* (2015), 67:1–67:12.

- [3] Acar, U.A. et al. 2016. *An introduction to parallel computing in C++*.
- [4] Acar, U.A. et al. 2012. Efficient primitives for creating and scheduling parallel computations. *Declarative aspects of multicore programming* (January 2012).
- [5] Acar, U.A. et al. 2011. Oracle scheduling: Controlling granularity in implicitly parallel languages. *Proceedings of the 2011 ACM international conference on object oriented programming systems languages and applications* (2011), 499–518.
- [6] Acar, U.A. et al. 2016. Oracle-guided scheduling for controlling granularity in implicitly parallel languages. *Journal of Functional Programming*. (2016).
- [7] Acar, U.A. et al. 2017. Parallel Work Inflation, Memory Effects, and their Empirical Analysis. *ArXiv e-prints*. (Sep. 2017).
- [8] Acar, U.A. et al. 2013. Scheduling parallel programs by work stealing with private deques. *18th ACM SIGPLAN symposium on principles and practice of parallel programming* (2013), 219–228.
- [9] Acar, U.A. et al. 2014. Theory and practice of chunked sequences. *The 22nd annual european symposium on algorithms*. Springer. 25–36.
- [10] Acar, U.A. et al. 2016. Dag-calculus: A calculus for parallel computation. *The 26th ACM SIGPLAN international conference on functional programming* (2016).
- [11] Bergstrom, L. et al. 2012. Lazy tree splitting. *Journal of Functional Programming*. 22, 4-5 (2012), 382–438.
- [12] Bergstrom, L. et al. 2013. Data-only flattening for nested data parallelism. *18th ACM SIGPLAN symposium on principles and practice of parallel programming* (2013), 81–92.
- [13] Bergstrom, L. et al. 2010. Lazy tree splitting. *The 20th ACM SIGPLAN international conference on functional programming* (2010), 93–104.
- [14] Blume, M. et al. 2008. Calling variadic functions from a strongly-typed language. *Proceedings of the 2008 ACM SIGPLAN workshop on ml* (2008), 47–58.
- [15] Felleisen, M. et al. 2009. *Semantics engineering with plt redex*. The MIT Press.
- [16] Fluet, M. et al. 2008. A scheduling framework for general-purpose parallel languages. *The 13th ACM SIGPLAN international conference on functional programming* (2008), 241–252.
- [17] Fluet, M. et al. 2010. Implicitly threaded parallelism in manticore. *Journal of Functional Programming*. 20, 5-6 (2010), 537–576.
- [18] Fluet, M. et al. 2008. Implicitly-threaded parallelism in manticore. *The*

13th ACM SIGPLAN international conference on functional programming (2008), 119–130.

[19] Fluet, M. et al. 2007. Manticore: A heterogeneous parallel language. *Proceedings of the 2007 workshop on declarative aspects of multicore programming* (2007), 37–44.

[20] Rainey, M. 2010. *Effective scheduling techniques for high-level parallel programming languages*. University of Chicago.

[21] Wise, D.S. et al. 2005. A paradigm for parallel matrix algorithms: Scalable cholesky. *In euro-par 2005 – parallel processing* (2005).