

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

Journals

2016 **Oracle-Guided Scheduling for Controlling Granularity in**

Implicitly Parallel Languages
JFP (Umut A Acar, Charguéraud, and Rainey 2016a).

2011 **Lazy Tree Splitting**
JFP (Bergstrom et al. 2012)

2010 **Implicitly threaded parallelism in Manticore**
JFP (Fluet et al. 2010)

Conferences

2017 **Contention in Structured Concurrency: Provably Efficient Dynamic NonZero Indicators for Nested Parallel Computation**
PPoPP (Umut A Acar, Ben-David, and Rainey 2017)

2016 **Dag-Calculus: A Calculus for Parallel Computation**
ICFP (Umut A Acar et al. 2016)

2015 **A Work-efficient Algorithm for Parallel Unordered Depth-first Search**
Supercomputing (Acar, Charguéraud, and Rainey 2015)

2014 **Theory and Practice of Chunked Sequences**
ESA (Acar, Charguéraud, and Rainey 2014)

2013 **Scheduling Parallel Programs by Work Stealing with Private Deques**
PPoPP (Acar, Charguéraud, and Rainey 2013)

Data-only Flattening for Nested Data Parallelism
PPoPP (Bergstrom et al. 2013)

2011 **Oracle Scheduling: Controlling Granularity in Implicitly Parallel Languages**
OOPSLA (Acar, Charguéraud, and Rainey 2011)

2010 **Lazy Tree Splitting**
ICFP (Bergstrom et al. 2010)

2008 **A Scheduling Framework for General-purpose Parallel Languages**
ICFP (Fluet, Rainey, and Reppy 2008)

Implicitly-threaded Parallelism in Manticore
ICFP (Fluet et al. 2008)

2005 **A Paradigm for Parallel Matrix Algorithms: Scalable Cholesky**
EuroPar (Wise et al. 2005)

Workshops

2012 **Efficient Primitives for Creating and Scheduling Parallel Computations**
DAMP (Acar, Charguéraud, and Rainey 2012)

2008 **Calling Variadic Functions from a Strongly-typed Language**

ML Workshop (Blume, Rainey, and Reppy 2008)
Scheduling_framework_08
2007 **Manticore: A Heterogeneous Parallel Language**
DAMP (Fluet et al. 2007)

Books

2016 **An Introduction to Parallel Computing in C++**
Online course materials (Umut A Acar, Charguéraud, and Rainey 2016b)
2010 **Effective Scheduling Techniques for High-Level Parallel Programming Languages**
PhD Disertation (Rainey 2010)
2009 **Semantics Engineering with PLT Redex**
Contributed chapter (Felleisen, Findler, and Flatt 2009, chap. 25)

Not published

2017 **Parallel Work Inflation, Memory Effects, and their Empirical Analysis**
arXiv (Umut A. Acar, Charguéraud, and Rainey 2017)

Software artifacts

- **Manticore** This artifact consists of a 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 project 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 (Acar, Charguéraud, and Rainey 2014). <http://deepsea.inria.fr/chunkedseq>
- **PASL** (Parallel Algorithm Scheduling Library. I coauthored this library with Umut Acar and Arthur Charguéraud. <http://deepsea.inria.fr/pasl/>
- **SML/NJ** (Standard ML of New Jersey) I worked on 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 project features a C++ implementation of the fast DFS-like graph-traversal algorithm from the SC'15 paper (Acar,

Charguéraud, and Rainey 2015). <http://deepsea.inria.fr/graph>

Professional service

Program committee member

2018	FHCP (co-chair) ACM SIGPLAN Workshop on Functional High-Performance Computing
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 (Umut A Acar, Charguéraud, and Rainey 2016a, @Oracle_scheduling_11).

In 2014, we used the course materials to teach a two-week, NSF-funded, undergraduate course at the University of Puerto Rico PASL Workshop.¹ 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.²

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** video at <https://www.youtube.com/watch?v=kOausvmMtmM>; slides at <http://mike-rainey.site/sc15-pdfs-talk.pdf>; Supercomputing, 2016
- **Scheduling Parallel Programs by Work Stealing with Private Deques** slides at <http://mike-rainey.site/ppopp2013.pdf>; Ppopp, 2013
- **Higher-level Implicit Parallelism with PASL** slides at <http://mike-rainey.site/lame2013.pdf>; LAME Workshop, 2013

¹<https://sites.google.com/site/paslpr14/home>

²<http://www.cs.cmu.edu/~15210/pasl.html>

- **Fork-join Model and Work stealing** slides at http://mike-rainey.site/reading_group_slides.pdf; MPI-SWS weekly seminar, 2011

All publications with links

Acar, Umut A., Arthur Charguéraud, and Mike Rainey. 2017. “Parallel Work Inflation, Memory Effects, and Their Empirical Analysis.” *CoRR* abs/1709.03767. <http://arxiv.org/abs/1709.03767>.

Acar, Umut A, Naama Ben-David, and Mike Rainey. 2017. “Contention in Structured Concurrency: Provably Efficient Dynamic Nonzero Indicators for Nested Parallel Computation.” ACM. <http://gallium.inria.fr/~rainey/dynsnzi.pdf>.

Acar, Umut A, Arthur Charguéraud, and Mike Rainey. 2011. “Oracle Scheduling: Controlling Granularity in Implicitly Parallel Languages.” In *Proceedings of the 2011 ACM International Conference on Object Oriented Programming Systems Languages and Applications*, 46:499–518. 10. ACM. http://gallium.inria.fr/~rainey/oracle_scheduling.pdf.

———. 2012. “Efficient Primitives for Creating and Scheduling Parallel Computations.” In *Declarative Aspects of Multicore Programming*. http://gallium.inria.fr/~rainey/damp2012_primitives.pdf.

———. 2013. “Scheduling Parallel Programs by Work Stealing with Private Deques.” In *18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 48:219–28. 8. ACM. <http://gallium.inria.fr/~rainey/full.pdf>.

———. 2014. “Theory and Practice of Chunked Sequences.” In *The 22nd Annual European Symposium on Algorithms*, 25–36. Springer. http://gallium.inria.fr/~rainey/chunked_seq.pdf.

———. 2015. “A Work-Efficient Algorithm for Parallel Unordered Depth-First Search.” In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 67:1–67:12. ACM. [pdfs_sc15.pdf](http://gallium.inria.fr/~rainey/pdfs_sc15.pdf).

———. 2016a. “Oracle-Guided Scheduling for Controlling Granularity in Implicitly Parallel Languages.” *Journal of Functional Programming*. Cambridge University Press. <http://gallium.inria.fr/~rainey/jfp-oracle-guided.pdf>.

———. 2016b. *An Introduction to Parallel Computing in C++*.

Acar, Umut A, Arthur Charguéraud, Mike Rainey, and Filip Sieczkowski. 2016. “Dag-Calculus: A Calculus for Parallel Computation.” In *The 26th ACM SIGPLAN International Conference on Functional Programming*. ACM. <http://gallium.inria.fr/~rainey/dag-calculus.pdf>.

- Bergstrom, Lars, Matthew Fluet, Mike Rainey, John Reppy, Stephen Rosen, and Adam Shaw. 2013. “Data-Only Flattening for Nested Data Parallelism.” In *18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 48:81–92. 8. ACM. <http://gallium.inria.fr/~rainey/ppopp13-flat.pdf>.
- Bergstrom, Lars, Matthew Fluet, Mike Rainey, John Reppy, and Adam Shaw. 2012. “Lazy Tree Splitting.” *Journal of Functional Programming* 22 (4-5). Cambridge University Press: 382–438. <http://gallium.inria.fr/~rainey/jfp-lts-submitted.pdf>.
- Bergstrom, Lars, Mike Rainey, John Reppy, Adam Shaw, and Matthew Fluet. 2010. “Lazy Tree Splitting.” In *The 20th ACM SIGPLAN International Conference on Functional Programming*, 45:93–104. 9. ACM. <http://gallium.inria.fr/~rainey/icfp10-lts.pdf>.
- Blume, Matthias, Michael Rainey, and John Reppy. 2008. “Calling Variadic Functions from a Strongly-Typed Language.” In *Proceedings of the 2008 ACM SIGPLAN Workshop on ML*, 47–58. ACM. <http://gallium.inria.fr/~rainey/ml-varargs.pdf>.
- Felleisen, Matthias, Robert Bruce Findler, and Matthew Flatt. 2009. *Semantics Engineering with Plt Redex*. The MIT Press.
- Fluet, Matthew, Mike Rainey, and John Reppy. 2008. “A Scheduling Framework for General-Purpose Parallel Languages.” In *The 13th ACM SIGPLAN International Conference on Functional Programming*, 43:241–52. 9. ACM. <http://gallium.inria.fr/~rainey/icfp08-sched.pdf>.
- Fluet, Matthew, Mike Rainey, John Reppy, and Adam Shaw. 2008. “Implicitly-Threaded Parallelism in Manticore.” In *The 13th ACM SIGPLAN International Conference on Functional Programming*, 43:119–30. 9. ACM. <http://gallium.inria.fr/~rainey/icfp08-implicit.pdf>.
- . 2010. “Implicitly Threaded Parallelism in Manticore.” *Journal of Functional Programming* 20 (5-6). Cambridge University Press: 537–76.
- Fluet, Matthew, Mike Rainey, John Reppy, Adam Shaw, and Yingqi Xiao. 2007. “Manticore: A Heterogeneous Parallel Language.” In *Proceedings of the 2007 Workshop on Declarative Aspects of Multicore Programming*, 37–44. ACM.
- Rainey, Mike. 2010. “Effective Scheduling Techniques for High-Level Parallel Programming Languages.” PhD thesis, University of Chicago. <http://gallium.inria.fr/~rainey/rainey-phd.pdf>.
- Wise, David S., Craig Citro, Joshua Hursey, Fang Liu, and Michael Rainey. 2005. “A Paradigm for Parallel Matrix Algorithms: Scalable Cholesky.” In *In Euro-Par 2005 – Parallel Processing*. http://dx.doi.org/10.1007/11549468_76.