

CURRICULUM VITAE: JUNE 27, 2016

Suresh Jagannathan

Department of Computer Science
Purdue University
West Lafayette, IN 47907

URL: <http://www.cs.purdue.edu/homes/suresh>
E-Mail: suresh@cs.purdue.edu

Research Interests

Programming language foundations: verification, type systems, program logics;
Programming language implementation: program analysis, verified compilation, runtime systems;
Concurrent and distributed programming: programming abstractions, memory models, consistency semantics.

Education

Massachusetts Institute of Technology September 1985–June 1989
Doctor of Philosophy in Electrical Engineering and Computer Science, 1989.

Massachusetts Institute of Technology September 1983–September 1985
Master of Science in Electrical Engineering and Computer Science, September 1985.

State University of New York, Stony Brook September 1979–December 1982
Bachelor of Science in Computer Science, (with Highest Honors), December 1982.

Present Position

Defense Advanced Research Projects Agency (DARPA) **Arlington, VA**
Program Manager September 2013 – Present

Program manager in the Information Innovation office responsible for managing and directing over \$175M in funding for three programming language and software systems-centric programs - PPAML (Probabilistic Programming Advancing Machine Learning), MUSE (Mining and Understanding Software Enclaves), and BRASS (Building Resource Adaptive Software Systems) - and credited with conception and initiation of the latter two. The programs collectively involve over 60 academic institutions, research labs, startups, and government contractors. A number of SBIR initiatives are also funded under their auspices.

Department of Computer Science **Purdue University**
Professor May 2007 – Present
Professor (Courtesy), Department of Electrical and Computer Engineering

Previous Positions

Computing Laboratory <i>Visiting Faculty</i> Sabbatical visitor and research scholar	Cambridge University July 2009 – July 2010
Department of Computer Science <i>Associate Professor</i> <i>Associate Professor (Courtesy),</i>	Purdue University September 2002 – May 2007 Department of Electrical and Computer Engineering
Storage Networks <i>Senior Director</i>	Princeton, NJ 2001 – 2002
Emphora, Inc. <i>VP, Advanced Technology</i>	Princeton, NJ 2000 – 2001
NEC Research Institute <i>Senior Research Scientist</i>	Princeton, NJ 1996 – 2000
NEC Research Institute <i>Research Scientist</i>	Princeton, NJ 1990 – 1996
Indian Institute of Science <i>Visiting Senior Faculty</i>	Bangalore, India 1997
New York University <i>Adjunct Faculty</i>	New York City 1994 – 1997
Indian Institute of Science <i>Adjunct Faculty</i>	Bangalore, India 1994 – 1997
Yale University <i>Adjunct Faculty</i>	New Haven, CT. 1992 – 1994
Yale University <i>Research Faculty</i>	New Haven, CT. 1989 – 1990
Massachusetts Institute of Technology <i>Research Assistant</i>	Cambridge, MA. 1983 – 1988
At&T Bell Laboratories <i>Consultant on the Technical Staff</i>	Holmdel, NJ. 1986

Awards and Honors

1. Seed for Success Award, Purdue University, 2013, 2014.
2. Graduate Student Mentoring Award, Purdue University, 2012.
3. University Faculty Scholar, Purdue University, 2007 – 2012.
4. Burton Morgan Entrepreneurship Award, 2008.
5. Best Patent Award, NEC Research Institute, 1998, 2002.

Publications

Refereed Journal Articles

1. K.C. Sivaramakrishnan, Gowtham Kaki, and Suresh Jagannathan, *Representation without Taxation: A Uniform, Low-Overhead, and High-Level Interface to Eventually Consistent Key-Value Stores*, IEEE Data Engineering Bulletin, 39(1), pages 52-64, 2016.
2. Mike Dodds, Suresh Jagannathan, Lars Birkedal, Matthew Parkinson, and Kasper Svendsen, *Verifying Custom Synchronisation Constructs Using Higher-Order Separation Logic*, ACM Transactions on Programming Languages and Systems, 38(2):4, 2016.
3. K.C. Sivaramakrishnan, Lukasz Ziarek, and Suresh Jagannathan, *MultiMLton: A Multicore-Aware Runtime for Standard ML*, Journal of Functional Programming, 24(6), pages 613-674, December 2014.
4. Suresh Jagannathan, Gustavo Petri, Jan Vitek, David Pichardie, and Vincent Laporte, *Atomicity Refinement for Verified Compilation*, ACM Transactions on Programming Languages and Systems, 36 (2) : 6, 2014.
5. Matko Botincan, Mike Dodds, and Suresh Jagannathan, *Resource Sensitive Synchronization Inference by Abduction*, ACM Transactions on Programming Languages and Systems, 35(2):8, July 2013.
6. Jaroslav Sevcik, Viktor Vafedias, Francesco Zappa Nardelli, Suresh Jagannathan, and Peter Sewell, *COMP CERTSO: A Verified Compiler for Relaxed-Memory Concurrency*, Journal of the ACM, 60(3):22, June 2013.
7. Armand Navabi, Xiangyu Zhang, and Suresh Jagannathan, *Dependence Analysis for Safe Futures*, Science of Computer Programming, 77(6), pages 707-726, June 2012.
8. Lukasz Ziarek and Suresh Jagannathan, *Lightweight Checkpointing for Concurrent ML*, Journal of Functional Programming, 20(2), pages 137-173, March 2010.
9. Bogdan Carbutar, Muralikrishna Ramanathan, Mehmet Koyuturk, Suresh Jagannathan, and Ananth Grama, *Efficient Tag Detection in RFID Systems*, Journal of Parallel and Distributed Computing, 69(2), 180-196, January 2009.
10. Lukasz Ziarek, Stephen Weeks, and Suresh Jagannathan, *Flattening Tuples in an SSA Intermediate Representation*, Higher-Order Symbolic Computation. 21(3), pages 333-358, September 2008.
11. Ronaldo Ferreira, Mehmet Koyuturk, Suresh Jagannathan and Ananth Grama, *Semantic Indexing in Structured Peer-to-Peer Networks*, Journal of Parallel and Distributed Computing. 68(1), pages 64-77, January 2008.
12. Lukasz Ziarek, Philip Schatz, and Suresh Jagannathan, *Modular Checkpointing for Atomicity*, Electronic Notes in Theoretical Computer Science. 174(9), pages 85–115, June 2007.

13. Ronaldo Ferreria, Muralikrishna Ramanathan, Ananth Grama, and Suresh Jagannathan, *Randomized Protocols for Duplicate Elimination in Peer-to-Peer Storage Systems*, IEEE Transactions on Parallel and Distributed Systems. 18(5), pages 686–696, May 2007.
14. Muralikrishna Ramanathan, Ronaldo Ferreria, Suresh Jagannathan, Ananth Grama, and Wojciech Szpankowski, *Randomized Leader Election*, Distributed Computing. 19(5), pages 403–418, April 2007.
15. Deepak Rao Bobbarjung, Suresh Jagannathan, Cezary Dubnicki, *Improving Duplicate Elimination in Storage Systems*, ACM Transactions on Storage. 2(4), pages 424–448, November 2006.
16. Adam Welc, Suresh Jagannathan, and Antony L. Hosking, *Revocation Techniques for Java Concurrency*, Concurrency and Computation: Practice and Experience. 18(12), pages 1613–1656, October 2006.
17. Asad Awan, Ronaldo Ferreira, Suresh Jaganathan, and Ananth Grama, *Unstructured Peer-to-Peer Networks for Sharing Processor Cycles*, Parallel Computing. 32(2), pages 115–135, February 2006.
18. Ronaldo Ferreira, Suresh Jagannathan, and Ananth Grama, *Locality in Structured Peer-to-Peer Networks*, Journal of Parallel and Distributing Computing. 66(2), pages 257–273, February 2006.
19. N.V. Krishna and Suresh Jagannathan, *Dynamic State Restoration Using Versioning Exceptions*, Journal of Higher-Order Symbolic Computing. 19(1), pages 101–124, January 2006.
20. Suresh Jagannathan, Jan Vitek, Adam Welc and Antony Hosking, *A Transactional Object Calculus*, Science of Computer Programming. 57(2), pages 164–186, August 2005.
21. Yuan-Yuan Zhou, Angelos Bilas, Suresh Jagannathan, Dimitrios Xinidis, Cezary Dubnicki, and Kai Li, *VI-Attached Database Storage*, IEEE Transactions on Parallel and Distributed Systems. 16(1), pages 35–50, January 2005.
22. Suresh Jagannathan *Continuation-based Transformations for Coordination Languages* Theoretical Computer Science. 240(1), pages 117–146, July 2000.
23. Satoru Fujita, Suresh Jagannathan, Richard Kelsey, James Philbin, and Toru Yamanouchi, *Mobid-ge: A Programming Language for Mobile Agents*, NEC Journal of Research and Development, 40(1), pages 537–552, July 1999.
24. Andrew Wright and Suresh Jagannathan *Polymorphic Splitting: An Effective Polyvariant Flow Analysis* ACM Transactions on Programming Languages and Systems. 20(1). pages 166–207, March 1998.
25. Suresh Jagannathan and Andrew Wright *Compiling Functional Languages with Flow Analysis* ACM Computing Surveys, 28(2), pages 337–339, June 1996.
26. Henry Cejtin, Suresh Jagannathan, and Richard Kelsey, *Higher-Order Distributed Objects*, ACM Transactions on Programming Languages and Systems, 17(5), pages 704–739, September 1995.
27. Suresh Jagannathan. *TS-Scheme: Distributed Data Structures in Lisp*, Lisp and Symbolic Computation, 7(4), pages 291–314, December 1994.

28. Stephen Weeks, Suresh Jagannathan, and James Philbin, *A Concurrent Abstract Interpreter*, *Lisp and Symbolic Computation*, 7(2), pages 173–193, June 1994.
29. Suresh Jagannathan, *Meta-Level Building Blocks for Modular Systems*, *ACM Transactions on Programming Languages and Systems*, 16(3), pages 456–492, May 1994.

Articles Refereed in Conference Proceedings

30. He Zhu, Gustavo Petri, Suresh Jagannathan, *Automatically Learning Shape Specifications*, *ACM Conference on Programming Language Design and Implementation (PLDI)*, June 2016. (**49/304 = 16% acceptance ratio for full papers.**)
31. He Zhu, Aditya Nori, and Suresh Jagannathan, *Learning Refinement Types*, *International Conference on Functional Programming (ICFP)*, September 2015. (**35/115 = 29% acceptance ratio for full papers.**)
32. Gustavo Petri, Jan Vitek, and Suresh Jagannathan, *Cooking the Books: Formalizing JMM Implementation Recipes*, *European Conference on Object-Oriented Programming (ECOOP)*, July 2015. (**31/136 = 23% acceptance ratio for full papers.**)
33. He Zhu, Gustavo Petri, and Suresh Jagannathan, *Poling: SMT Aided Linearizability Proofs*, *Computer Aided Verification (CAV)*, July 2015. (**68/252 = 27% acceptance ratio for full papers.**)
34. Malavika Samak, Muralikrishna Ramanathan, and Suresh Jagannathan, *Synthesizing Racy Tests*, *ACM Conference on Programming Language Design and Implementation (PLDI)*, June 2015. (**58/303 = 19% acceptance ratio for full papers.**)
35. K.C. Sivaramakrishnan, Gowtham Kaki, and Suresh Jagannathan, *Declarative Programming over Eventually Consistent Data Stores*, *ACM Conference on Programming Language Design and Implementation (PLDI)*, June 2015. (**58/303 = 19% acceptance ratio for full papers.**)
36. He Zhu, Aditya Nori, and Suresh Jagannathan, *Dependent Array Type Inference from Tests*, *16th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI)*, pages 412-430, January 2015.
37. Gowtham Kaki and Suresh Jagannathan, *A Relational Framework for Higher-Order Shape Analysis*, *ACM International Conference on Functional Programming (ICFP)*, pages 311 - 324, September 2014. (**28/97 = 29% acceptance ratio for full papers.**)
38. Suresh Jagannathan, Gustavo Petri, Jan Vitek, David Pichardie, Vincent Laporte, *Atomicity Refinement for Verified Compilation*, *ACM Conference on Programming Language Design and Implementation (PLDI)*, page 5, June 2014.
39. K.C. Sivaramakrishnan, Lukasz Ziarek, Suresh Jagannathan, R^{CML} : *A Prescription for Safely Relaxing Synchrony*, *International Conference on Practical Aspects of Declarative Programming (PADL)*, pages 1-16, January 2014.
40. Gregor Richards, Christian Hammer, Francesco Zappa Nardelli, Suresh Jagannathan, Jan Vitek, *Flexible Access Control Policies with Delimited Histories and Revocation*, *OOPSLA'13: Proceedings of the 2013 ACM International Conference on Object-Oriented Programming Systems*,

- Languages, and Applications, pages 305 - 322, October, 2013. (**50/189 = 26% acceptance ratio for full papers.**)
41. He Zhu and Suresh Jagannathan, *Compositional and Lightweight Dependent Type Inference for ML*, 14th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), pages 295-314, January 2013.
 42. Delphine Demange, Vincent Laporte, Lei Zhao, Suresh Jagannathan, David Pichardie, and Jan Vitek, *Plan B: A Buffered Memory Model for Java*, ACM Symposium on Principles of Programming Languages (POPL), pages 329-342, 2013. (**43/239 = 18% acceptance ratio for full papers.**)
 43. K.C. Sivaramakrishnan, Lukasz Ziarek, and Suresh Jagannathan, *Eliminating Read Barriers Through Procrastination and Cleanliness*, International Symposium on Memory Management, pages 49-60, June 2012.
 44. Matko Botinkan, Mike Dodds, and Suresh Jagannathan, *Resource-Sensitive Synchronization Inference Using Abduction*, ACM Conference on Principles of Programming Languages, pages 309-322, January 2012, (**44/205 = 21% acceptance ratio for full papers.**)
 45. Dasarath Weeratunge, Xiangyu Zhang, and Suresh Jagannathan, *Accentuating the Positive: Atomicity Inference and Enforcement Using Correct Executions*, ACM Conference on Object-Oriented Programming, Systems, and Languages, (OOPSLA), pages 19-34, October 2011.
 46. Lukasz Ziarek, Siddharth Tiwary, and Suresh Jagannathan, *Isolating Determinism in Multithreaded Programs*, 2nd International Conference on Runtime Verification, pages 63-78, September 2011, (**22/72 = 30% acceptance ratio for full papers.**)
 47. Lukasz Ziarek, K.C. Sivaramakrishnan, and Suresh Jagannathan, *Composable Asynchronous Events*, ACM Conference on Programming Language Design and Implementation, pages 628-639 June 2011 (**55/236 = 23% acceptance ratio for full papers.**)
 48. Jaroslav Sevcik, Viko Vafedias, Francesco Zappa Nardelli, Suresh Jagannathan, and Peter Sewell, *Relaxed Memory Concurrency and Verified Compilation*, ACM Conference on Principles of Programming Languages, pages 43-54, January 2011 (**49/209 = 23% acceptance ratio for full papers.**)
 49. Mike Dodds, Suresh Jagannathan, and Matthew Parkinson, *Modular Reasoning for Deterministic Parallelism*, ACM Conference on Principles of Programming Languages, pages 259-270, January 2011 (**49/209 = 23% acceptance ratio for full papers.**)
 50. Karthik Kambatla, Naresh Rapolu, Suresh Jagannathan, and Ananth Grama, *Asynchronous Algorithms in MapReduce*, IEEE Conference on Cluster Computing, pages 245-254, September 2010 (**33/107 = 31% acceptance ratio for full papers.**)
 51. Dasarath Weeratunge, Xiangyu Zhang, William N. Sumner, and Suresh Jagannathan, *Analyzing Concurrency Bugs Using Dual Slicing*, International Symposium on Software Testing and Analysis, pages 253-264, July 2010 (**24/105 = 23% acceptance ratio for full papers.**)
 52. Dasarath Weeratunge, Xiangyu Zhang, and Suresh Jagannathan, *Analyzing Multicore Dumps to Facilitate Concurrency Bug Reproduction* Architectural Support for Programming Languages and Systems, pages 155-166, March 2010 (**32/181 = 18% acceptance ratio for full papers.**)

53. Lukasz Ziarek, K. C. Sivaramakrishnan, and Suresh Jagannathan, *Partial Memoization of Concurrency and Communication* ACM International Conference on Functional Programming, September 2009, pages 161-172 (**26/85 = 30% acceptance ratio for full papers.**)
54. Kevin Hoffman, Patrick Eugster, and Suresh Jagannathan, *Semantics-Aware Trace Analysis*, ACM Conference on Programming Language Design and Implementation, pages 453-464, June 2009. (**41/194 = 21% acceptance ratio for full papers.**)
55. Armand Navabi and Suresh Jagannathan, *Exceptionally Safe Futures*, Coordination, pages 47-65, June 2009.
56. Xiangyu Zhang, Armand Navabi, and Suresh Jagannathan, *Alchemist: A Transparent Dependence Profiling Infrastructure*, ACM Conference on Code Generation and Optimization, pages 47-58, March 2009.
57. Asad Awan, Suresh Jagannathan, and Ananth Grama, *Scalable Collection in Sensor Networks*, High-Performance Computing, December 2008. (**46/317 = 14% acceptance ratio for full papers.**)
58. Muralikrishna Ramanathan, Koushik Sen, Ananth Grama, and Suresh Jagannathan, *Protocol Inference Using Static Path Profiles*, Static Analysis Symposium, July 2008, pages 78-92.
59. Lukasz Ziarek, Adam Welc, Ali-Reza Adl-Tabatabai, Vijay Menon, Tatiana Shpeisman, and Suresh Jagannathan, *A Uniform Transactional Execution Environment for Java*, European Conference on Object-Oriented Programming, June 2008, pages 129-154. (**27/138 = 20% acceptance ratio for full papers.**)
60. Armand Navabi, Xiangyu Zhang, and Suresh Jagannathan, *Quasi-Static Scheduling for Safe Futures*, ACM Symposium on Principles and Practice of Parallel Programming, February 2008, pages 23-32. (**25/102 = 25% acceptance ratio for full papers.**)
61. Muralikrishna Ramanathan, Mehmet Koyuturk, Ananth Grama, and Suresh Jagannathan, *PHALANX: A Graph-Theoretic Framework for Test Case Prioritization*, ACM Symposium on Applied Computing, pages 667-673. (**384/1307 = 29% acceptance ratio for full papers.**)
62. Asad Awan, Ahmed Sameh, Suresh Jagannathan, and Ananth Grama, *Building Verifiable Sensing Applications Through Temporal Logic Specifications*, International Conference on Computational Science, May 2007, pages 1205-1212.
63. Muralikrishna Ramanathan, Ananth Grama, and Suresh Jagannathan, *Static Specification Inference Using Predicate Mining*, ACM Conference on Programming Language Design and Implementations, 2007. pages 123-134. (**45/178 = 25% acceptance ratio for full papers.**)
64. Asad Awan, Suresh Jagannathan, and Ananth Grama, *Macroprogramming Heterogeneous Sensor Networks using COSMOS*, European Symposium on Systems, pages 159-172, 2007. (**29/140 = 20% acceptance ratio for full papers.**)
65. Muralikrishna Ramanathan, Suresh Jagannathan, and Ananth Grama, *Inferring User-Defined Function Precedence Protocols*, IEEE International Conference on Software Engineering, 2007. pages 240-250. (**50/334 = 15% acceptance ratio for full papers.**)

66. Suresh Jagannathan, Gopal Pandurangan, and Sriram Srinivasan, *Query Protocols for Highly Resilient Peer-to-Peer Networks*, ISCA Conference on Parallel and Distributed Computing Systems, September 2006. pages 247–252.
67. Muralikrishna Ramanathan, Ananth Grama, and Suresh Jagannathan, *Automatically Detecting Variations Across Program Versions*, ACM/IEEE International Conference on Automated Software Engineering, September, 2006. pages 242–252. **(22/121 = 18% acceptance ratio for full papers.)**
68. Lukaz Ziarek, Philip Schatz, and Suresh Jagannathan, *Stabilizers: A Modular Checkpointing Abstraction for Concurrent Functional Programs*, ACM International Conference on Functional Programming, September, 2006. pages 136–147. **(24/76 = 25% acceptance ratio for full papers.)**
69. Adam Welc, Antony Hosking, and Suresh Jagannathan, *Transparently Reconciling Transactions with Locking for Java Synchronization*, European Conference on Object-Oriented Programming, Springer LNCS 4067, July 2006, pages 148–173. **(21/160 = 13% acceptance ratio for full papers.)**
70. Muralikrishna Ramanathan, Suresh Jagannathan, and Ananth Grama, *Trace-based Memory Aliasing Across Program Versions*, Fundamental Approaches to Software Engineering, Springer LNCS 3922, March 2006, pages 381–395. **(30/180 = 17% acceptance ratio for full papers.)**
71. Jeremy Manson, Jason Baker, Antonio Cuneì, Suresh Jagannathan, Marke Prochazka, Jan Vitek, and Bin Xin, *Preemptible Atomic Regions for Real-time Java*, IEEE Real-Time Systems Symposium, December 2005, pages 62–71. **(37/176 = 21% acceptance ratio for full papers.)**
72. Adam Welc, Suresh Jagannathan, and Antony L. Hosking, *Safe Futures for Java*, ACM Conference on Object-Oriented Languages, Systems and Applications, October 2005, pages 439–453. **(32/174 = 18% acceptance ratio for full papers.)**
73. Ronaldo Ferreira, Muralikrishna Ramanathan, Ananth Grama, and Suresh Jagannathan, *Randomized Protocols for Duplicate Elimination in Peer-to-Peer Storage Systems*, IEEE 5th International Conference on Peer-to-Peer Systems, September 2005, pages 201–208. **(25/130 = 19% acceptance ratio for full papers.)**
74. Ronaldo Ferreira, Muralikrishna Ramanathan, Asad Awan, Suresh Jagannathan, and Ananth Grama, *Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks*, IEEE 5th International Conference on Peer-to-Peer Systems, September 2005, pages 165–172. **(25/130 = 19% acceptance ratio for full papers.)**
75. Ronaldo Ferreira, Ananth Grama, and Suresh Jagannathan, *Plethora: An Efficient Wide-Area Storage System*, International Conference on High-Performance Computing, Springer LNCS 3296, December 2004, pages 252–261. **(48/214 = 22% acceptance ratio for full papers.)**
76. Adam Welc, Antony L. Hosking, and Suresh Jagannathan, *Preemption-Based Avoidance of Priority Inversion for Java*, IEEE International Conference on Parallel Processing (ICPP), August 2004, pages 529–538.

77. Ronaldo Ferreira, Suresh Jagannathan, and Ananth Grama, *A Locality Enhancing Peer-to-Peer Network*, IEEE International Conference on Parallel and Distributed Systems, July 2004, pages 25–36. **(65/292 = 22% acceptance ratio for full papers.)**
78. Adam Welc, Suresh Jagannathan, and Antony Hosking, *Transactional Monitors for Concurrent Objects*, European Conference on Object-Oriented Programming, Springer LNCS 3086, June 2004, pages 519–542. **(25/132 = 19% acceptance ratio for full papers.)**
79. Jan Vitek, Suresh Jagannathan, Adam Welc and Antony Hosking, *A Semantic Framework for Designer Transactions*, European Symposium on Programming, Springer LNCS 2986, March 2004, pages 249–263. **(27/118 = 22% acceptance ratio for full papers.)**
80. Suresh Jagannathan and Jan Vitek, *Optimistic Concurrency Semantics for Transactions in Coordination Languages*, Coordination: Languages and Models, Springer LNCS 2949, February 2004, pages 183–198. **(20/72 = 28% acceptance ratio for full papers.)**
81. Ronaldo Ferreira, Ananth Grama, Suresh Jagannathan *An IP Address Based Caching Scheme for Peer-to-Peer Networks*, IEEE Globecom, December 2003, pages 3845–3850.
82. Yuan-Yuan Zhou, Angelos Bilas, Suresh Jagannathan, Cezary Dubnicki, James Philbin, Kai Li, *Experiences with VI Communication for Database Storage*, IEEE International Symposium on Computer Architecture, May 2002, pages 257–270. **(27/180 = 15% acceptance ratio for full papers.)**
83. Cristian Ungureanu and Suresh Jagannathan, *Concurrency Analysis for Java*, International Static Analysis Symposium, Springer LNCS 1824, July 2000, pages 413–432.
84. Henry Cejtin, Suresh Jagannathan, Stephen Weeks, *Flow-Directed Closure Conversion for Typed Languages*, European Symposium on Programming, Springer LNCS 1782, March 2000, pages 56–71. **(27/84 = 32% acceptance ratio for full papers.)**
85. Suresh Jagannathan, Peter Thiemann, Stephen Weeks, Andrew Wright, *Single and Loving It: Must-Alias Analysis for Higher-Order Languages*, ACM Symposium on Principles of Programming Languages, January 1998, pages 329–340. **(32/175 = 18% acceptance ratio for full papers.)**
86. Suresh Jagannathan *Communication-Passing Style for Coordination Languages* Second International Conference on Coordination Models and Languages, Springer LNCS 1282, September 1997, pages 131–149. **(19/69 = 26% acceptance ratio for full papers.)**
87. Suresh Jagannathan, Andrew Wright, and Stephen Weeks *Typed-Directed Flow Analysis for Typed Intermediate Languages* 1997 International Static Analysis Symposium, Springer LNCS 1302, September 1997, pages 207–224. **(23/70 = 32% acceptance ratio for full papers.)**
88. Elisabeth Freeman, David Gelernter, and Suresh Jagannathan, *Uniformity of Computation Environment and Computation in MAP*, IEEE International Symposium on Visual Languages, September 1996, pages 130–139. **(35/91 = 30% acceptance ratio for full papers.)**
89. Suresh Jagannathan and Andrew Wright, *Flow-directed Inlining*, ACM Conference on Programming Language Design and Implementation, May 1996, pages 193–205. **(28/112 = 25% acceptance ratio for full papers.)**

90. Suresh Jagannathan and Andrew Wright, *Effective Flow Analysis for Avoiding Runtime Checks*, Second International Static Analysis Symposium, Springer LNCS 983, September 1995, pages 207–224.
91. Elisabeth Freeman, David Gelernter and Suresh Jagannathan, *In Search of a Simple Visual Vocabulary*, IEEE International Symposium on Visual Languages, September 1995, pages 302–308. **(35/140 = 25% acceptance ratio for full papers.)**
92. Suresh Jagannathan and Stephen Weeks, *A Unified Treatment of Flow Analysis in Higher-Order Languages*, ACM Conference on Principles of Programming Languages, January 1995, pages 393–407. **(34/155 = 22% acceptance ratio for full papers.)**
93. Suresh Jagannathan and Stephen Weeks, *Analyzing Stores and References in Parallel Symbolic Languages*, ACM Conference on Lisp and Functional Programming, June 1994, pages 294–305. **(24/109 = 22% acceptance ratio for full papers.)**
94. Suresh Jagannathan, *Dynamic Modules in Higher-Order Languages*, IEEE International Conference on Computer Languages, May 1994, pages 74–87. **(28/102 = 28% acceptance ratio for full papers.)**
95. Suresh Jagannathan and James Philbin *High-Level Abstractions for Efficient Concurrent Systems*, International Conference on Programming Languages and Systems Architectures, Springer LNCS 782, March, 1994. **(15/60 = 25% acceptance ratio for full papers.)**
96. Suresh Jagannathan, *Locality Abstractions for Parallel and Distributed Computing*, International Conference on Theory and Practice of Parallel Programming, November 1994, pages 320–345, Springer-Verlag LNCS 905. (invited)
97. Suresh Jagannathan and James Philbin, *A Customizable Substrate for Concurrent Languages*, ACM Conference on Programming Language Design and Implementation, June 1992, pages 55–67. **(30/150 = 20% acceptance ratio for full papers.)**
98. Suresh Jagannathan and James Philbin, *A Foundation for an Efficient Multi-Threaded Scheme System*, ACM Conference on Lisp and Functional Programming, June 1992, pages 345–357. **(32/109 = 29% acceptance ratio for full papers.)**
99. Suresh Jagannathan and Gul Agha, *A Reflective Model of Inheritance*, European Conference on Object-Oriented Programming, Springer LNCS 615, June 1992, pages 350–371. **(23/124 = 18% acceptance ratio for full papers.)**
100. Suresh Jagannathan, *Type and Sharing Analysis for Concurrent Objects*, 29th Annual Allerton Conference on Computers, Communication and Control, September, 1991.
101. Suresh Jagannathan *Customization of First-Class Tuple-Spaces in a Higher-Order Language*, Parallel Languages and Architectures, Europe, (PARLE), Springer LNCS 506, June, 1991, pages 254–271.
102. Suresh Jagannathan, *Coercion as a Metaphor for Computation*, IEEE International Conference on Computer Languages, March 1990, pages 116–127. **(32/118 = 27% acceptance ratio for full papers.)**

103. David Gelernter, Suresh Jagannathan, and Tom London, *Parallelism, Persistence and Meta-Cleanliness in the Symmetric Lisp Interpreter*, ACM Conference on Interpreters and Interpretive Techniques, June 1987, pages 274–282.
104. David Gelernter, Suresh Jagannathan, and Tom London, *Environments as First-Class Objects*, ACM Conference on Principle of Programming Languages, January 1987, pages 98–110. **(29/108 = 26% acceptance ratio for full papers.)**

Books and Book Chapters

104. Muralikrishna Ramanathan, Ananth Grama, and Suresh Jagannathan, *Path-Aware Static Program Analysis for Specification Mining*, Mining Software Specifications: Methodologies and Applications, CRC Press, pages 309-357, 2011.
105. Guy Blelloch, K. Mani Chandy and Suresh Jagannathan (editors), *Specification of Parallel Algorithms*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1994.
106. Suresh Jagannathan, *Optimizing Analysis for First-Class Tuple-Spaces*, Languages and Compilers for Parallel Computing, MIT Press, 1990.
107. David Gelernter and Suresh Jagannathan, *Programming Linguistics*, MIT Press, Cambridge, MA 02139. 1990.
108. Suresh Jagannathan, *A Model of Data Backup and Recovery in a Computer System for Functional Programming*, Formal Description of Programming Language Concepts, March 1986. North-Holland Press. pages 167–192.

Refereed Workshop Articles

109. K.C. Sivaramakrishnan, Lukasz Ziarek, Suresh Jagannathan, *Rx-CML: Migrating MultiMLton to the Cloud*, ML Workshop, September 2013.
110. K.C. Sivaramakrishnan, Lukasz Ziarek, Suresh Jagannathan, *A Coherent and Managed Runtime for ML on the SCC*, Intel MARC Symposium, November 2012. *Best Paper Award*
111. Naresh Rapolu, Karthik Kambatla, Suresh Jagannathan, and Ananth Grama, *TransMR: Data-Centric Programming Beyond Data Parallelism*, Usenix Hot Clouds Workshop, June 2011.
112. Nicholas Kidd, Suresh Jagannathan, and Jan Vitek, *One Stack to Run Them All*, SPIN workshop, September 2010.
113. Lukasz Ziarek, K.C. Sivaramakrishnan, Raghavendra Prasad, and Suresh Jagannathan, *Lightweight Asynchrony Using Parasitic Threads*, Declarative Aspects of Multicore Programming, January 2010.
114. Lukasz Ziarek, Suresh Jagannathan, Matthew Fluet, and Umut Acar, *Speculative N-Way Barriers*, Declarative Aspects of Multicore Programming, January 2009.
115. Lukasz Ziarek and Suresh Jagannathan, *Memoizing Multi-threaded Transactions*, Declarative Aspects of Multicore Programming, January 2008.

116. Suresh Jagannathan, *Weaving Atomicity Through Dynamic Dependence Tracking*, NSF Next Generation Workshop, March, 2007.
117. Lukaz Ziarek, Philip Schatz, and Suresh Jagannathan, *Modular Checkpointing for Atomicity*, Workshop on Threading and Verification, August 2006.
118. Jeremy Manson, Jan Vitek, and Suresh Jagannathan, *Dynamic Aspects for Runtime Fault Determination and Recovery*, Next Generation Software Workshop, April 2006.
119. Lukaz Ziarek, Philip Schatz, and Suresh Jagannathan, *Modular Checkpointing for Atomicity*, Workshop on Threads and Verification, August 2006.
120. Deepak Bobbarjung, Suresh Jagannathan, and Cezary Dubnicki, *Fingerdiff: Improved Duplicate Elimination in Storage Systems*, 23rd IEEE Conference on Mass Storage Systems and Technologies, May 2006.
121. Jeremy Manson, Jan Vitek, and Suresh Jagannathan, *Dynamic Aspects for Runtime Fault Determination and Monitoring*, 2nd Dynamic Aspects Workshop, pages 27–33, March 2005.
122. Filip Pizlo, Marek Prochazka, Suresh Jagannathan, and Jan Vitek, *Transactional Lock-Free Objects for Real-time Java*, Workshop on Concurrency and Synchronization in Java Programs, June 2004.
123. Michael Hicks, Suresh Jagannathan, Richard Kelsey, Jonathan Moore, and Cristian Ungureanu, *Transparent Communication for Distributed Objects in Java*, ACM Java Grande 99, August 1999, pages 160–170.
124. Suresh Jagannathan and Richard Kelsey, *On the Interaction Between Mobile Processes and Objects*, IEEE Heterogeneous Computing Workshop, April 1998, pages 163–171.
125. Aaron Hertzman, Suresh Jagannathan, Cristian Ungureanu, and Andrew Wright *Compiling Java to a Typed Intermediate Language: A Preliminary Report*, Types in Compilation, January 1998, Springer-Verlag LNCS 1473, pages 9–27.
126. Suresh Jagannathan, *TS/Scheme: Distributed Data Structures in Lisp*, Parallel Symbolic Computing: Languages, Systems, and Applications, October 1992, Springer-Verlag LNCS 748, pages 260–268.
127. James Philbin, Suresh Jagannathan and Rajiv Mirani, *Virtual Topologies: A New Concurrency Abstraction for High-Level Parallel Programs*, 5th Workshop on Languages and Compilers for Parallel Computing, August 1995, Springer-Verlag LNCS 1033, pages 450–464.
128. Suresh Jagannathan, *Reflective Building Blocks for Modular Systems*, ACM International Workshop on Reflection and Meta-Level Architectures, November 1992, pages 61–69.
129. Suresh Jagannathan, *Expressing Fine-Grained Parallelism Using Distributed Data Structures*, Workshop on Research Directions in High-Level Parallel Languages, June 1991. Springer-Verlag LNCS 574, pages 77–92,

Thesis Committees

Purdue University

Primary or Co-Advisor

1. Ronaldo Ferreria, Ph.D December 2006. Co-advisor with Ananth Grama. Topic: *Distributed Algorithms in Peer-to-Peer Networks*. (Currently: Professor at Universidade Federal de mato Grossu do Sul, Brazil)
2. Adam Welc, Ph.D May 2006. Co-advisor with Antony Hosking. Topic: *Optimistic Concurrency Control for Programming Languages*. (Currently: Research Staff Member at Oracle Labs)
3. Deepak Rao Bobbarjung, Ph.D May 2007. Primary advisor. Topic: *Content-based Storage Systems*. (First position: Software Engineer at VMWare)
4. Asad Awan, Ph.D December 2007. Co-advisor with Ananth Grama. Topic: *Macroprogramming Scalable Sensor Networks*. (First position: Member of Technical Staff, Conviva)
5. Muralikrishna Ramanathan, Ph.D May 2008. Primary advisor. Topic: *Path-Aware Analysis of Program Invariants*. (Currently: Assistant Professor at Indian Institute of Science, Bangalore, India)
6. Lukasz Ziarek, Ph.D, May 2011. Primary advisor. Topic: *Abstractions for Robust Higher-Order Message-Based Communication*. (Currently: Assistant Professor at SUNY Buffalo)
7. Armand Navabi, Ph.D, May 2011. Primary advisor. Topic: *The Semantics and Analysis of Safe Futures*. (Currently: Software Engineer at Google)
8. Dasarath Weeratgune, Ph.D, May 2012. co-advisor. Topic: *Dynamic Analysis for Detecting and Remedying Concurrency Bugs* (Currently: Research Scientist at Intel)
9. K.C. Sivaramakrishnan, Ph.D December 2014. Primary advisor. Topic: *Functional Programming Abstractions for Weak Consistency*. (Currently: Researcher OCaml Labs, Fellow Cambridge University, UK)
10. He Zhu, Ph.D expected June 2016. Primary advisor. Topic: *Learning Program Invariants*
11. Gowtham Kaki, Ph.D expected June 2017. Primary advisor. Topic: *Relational Logic and Dependent Types*.

Ph.D. Committee Member

1. Baskar Sridharan, Ph.D August 2003. Committee member. Title: *Enforcing Safety in Pervasive Computing Environments*. Advisor: Aditya Mathur.
2. Di Ma, Ph.D May 2004. Committee member. Title: *Bounding the Stack Size of Interrupt-Driven Programs*. Advisor: Jens Palsberg.
3. Maleq Khan, Ph.D May 2007. Committee member. Topic: *Algorithms for Low-Power Networks*. Advisor: Gopal Pandurangan.
4. Maxim Martynov, Ph.D August 2007. Committee member. Topic: *Network Processors*. Advisor: Douglas Comer.
5. Ahmad Zeshan-Chishti, Ph.D August 2007. Committee member. Topic: *Cache Architectures for Multiprocessors*. Advisor: T.N. Vijaykumar (ECE).

6. Srinivasan Venkatakrishnan, Ph.D October 2008. Committee member. Topic: *A Computational Framework for Hierarchical Synthesis in the Presence of Cracks*. Advisor: Ganesh Subbaryan (ECE).
7. Barry Wittman, Ph.D November 2008. Committee member. Topic: *Approximation Algorithms for Time-Constrained Vehicle Routing Problems*. Advisor: Greg Fredrickson.
8. David Cunningham, Ph.D January 2010. External committee member. (Imperial College). Topic: *Locking Atomic Sections*. Advisor: Susan Eisenbach.
9. Filip Pizlo, Ph.D April 2012. Committee member. Topic: *Fragmentation Tolerant Real Time Garbage Collection*. Advisor: Jan Vitek.
10. K.R. Jayaraman, Ph.D June 2012. Committee member. Topic: *Abstractions and Mechanisms for Scalable Event-based Distributed Systems*, Advisor: Patrick Eugster.
11. William Sumner, Ph.D June 2013. Committee member. Topic: *Execution Indexing for Program Debugging*, Advisor: Xiangyu Zhang.
12. Kevin Hoffman, Ph.D June 2013. Committee member. Topic: *Ribbons: A Partially Shared Memory Programming Model*, Advisor: Patrick Eugster.
13. Gregor Richards, Ph.D June 2014. Committee member. Topic: *Refinement of Web Software Motivated by Real-World Patterns*, Advisor: Jan Vitek.
14. Jorge Samayoa Ranero, Ph.D December 2014. Committee member. Topic: *A Calculus for Decision Systems*. Advisor: Abhijit Deshmukh.

M.S Students

1. Vamsi Vytla, M.S. May 2005. Committee member. Topic: *Logical Models for Naive Physics*. Advisor: Jeffrey Siskind (ECE).
2. Philip McGachey. M.S. May 2004, Committee member. Topic: *Compiling Object-Oriented Languages*. Advisor: Antony Hosking.
3. Filip Pizlo, M.S. May 2006. Committee member. Topic: *Hybrid Points-to Analysis*. Advisor: Jan Vitek.
4. Hiroshi Yamauchi, M.S. December 2006. Committee member. Topic: *Register Allocation*. Advisor: Jan Vitek.

Prior to Purdue

1. Pradeep Varma, Yale University, Ph.D 1994. *Compile-time Analyses and Run-Time Support for a Higher-Order, Distributed Data-structures based Parallel Language*. Primary advisor. (IBM Research, India)
2. T.S. Mohan, Indian Institute of Science, Ph.D 1994, *Interaction Paradigms for Distributed Object-Oriented Programming*. Primary advisor. (Infosys, Bangalore, India)
3. Chi Yao, New York University, Ph.D 1994. *Representing Control in Parallel Applicative Programming*. Committee member.
4. Shakil Ahmed, Yale University, Ph.D 1994. *A Higher Level Parallel Programming Environment*. Committee member.

5. Rajiv Mirani, Yale University. Ph.D 1994 *Using Annotations to Control Parallelism in Functional Languages*. Committee member.
6. James Philbin, Yale University, Ph.D 1993, *An Operating System for Modern Programming Languages*. Primary advisor. (Johns Hopkins, Division of Health Sciences Informatics)
7. Alvaro Campos, Princeton University. Ph.D 1993 *Distributed, Garbage-Collected, Persistent, Virtual Address Spaces*. Committee member.
8. Lesley Matheson, Princeton University, Ph.D 1993 *Multigrid Algorithms on Massively Parallel Computers*. Committee member.

Undergraduate Research Advisor

1. Brent Woodhouse. 2012-2013. *Specification and Analysis of Church Using Coq*.
2. Jeremy Orlow. 2006-2007. *Memoization Techniques for Concurrent Programs*.
3. Zach Tatlock. 2005-2006. *Proving the Absence of Bugs Using Theorem Proving*.
4. Philip Schatz. 2005-2006. *Compiling Concurrent ML on MLton*.
5. Brett Daniel. 2004-2005. *Interpreted Graphical Composition Programming Language*.
6. Lee Ballard. 2004-2005. *Interpreted Graphical Composition Programming Language*.
7. Chris Willmore. 2004-2005. *Interpreted Graphical Composition Programming Language*.

Funding

Federal Agencies

1. Air Force Research Laboratory, \$746,000, *Verified Compilation for Managed Languages*, June 2013 - June 2017. PI. (50% for Jagannathan)
2. National Science Foundation, \$475,000, *Havoc: Verified Compilation for Managed Languages*, August 2013 - August 2016. PI. (50% for Jagannathan)
3. National Science Foundation, \$200,000. *SI2: Conceptualization: Dyanmic Languages for Scalable Data Analytics*, October 1, 2012 - March 31, 2014. co-PI (30% for Jagannathan)
4. National Science Foundation, \$450,000. *Programming with Non-Coherent Memory*, September 1, 2012 - August 30, 2015. PI. (100% for Jagannathan)
5. National Science Foundation, \$80,000. *EAGER: Verified Compilation and High-Level Memory Models*, May 1, 2012 - May 1, 2013. PI. (100% for Jagannathan)
6. National Science Foundation, \$1,600,000. *Robust Distributed Wind Power Engineering*, September 1, 2011 - September 30, 2014. Co-Principal Investigator. (25% for Jagannathan)
7. National Science Foundation, \$25,000,000. *Science and Technology Center on the Emerging Frontiers of Science of Information*, September 2010 - August 2015. Senior Personnel (5% for Jagannathan)

8. National Science Foundation. \$556,300. June 1, 2010 – June 30, 2013. *A Computational Infrastructure for Scalable Transactional Memory Abstractions in Managed Languages*. Principal Investigator. (75% for Jagannathan)
9. National Science Foundation. \$12,000. May 2010. *Third International Summer School on Trends in Concurrency*. co-PI.
10. National Science Foundation. \$450,000. February 1, 2009 – January 31, 2013. *Eager Maps and Lazy Folds for Graph-Structured Applications*. Principal Investigator. (60% for Jagannathan)
11. National Science Foundation. \$1,000,000. September 15, 2008 – January 1, 2013. *An Open Source Transactional Infrastructure*. co-PI. (33% for Jagannathan).
12. National Science Foundation. \$350,000. September 1, 2007 – September 1, 2010. *Kala: An Efficient and Scalable Time Travel Infrastructure for Concurrent Systems*. Principal Investigator.
13. National Science Foundation. \$249,857. September 1, 2007 – September 1, 2009. *CT-ER: Relax: Controlled Declassification Using Transactional Memory*. co-PI. (50% for Jagannathan).
14. National Science Foundation. \$99,979. March 1, 2006 – March 1, 2008, *CRI: A Computational Infrastructure for Experimentation on Relaxed Concurrency Abstractions and their Applications*. Principal Investigator. (65% for Jagannathan).
15. National Science Foundation. \$350,000. August 1, 2005 – August 31, 2008. *CSR/AES: Fault Determination and Recovery in Cycle Sharing Infrastructures*. Principal Investigator. (65% for Jagannathan).
16. National Science Foundation. \$549,635. September 15, 2003 – August 31, 2006. *Plethora: A Wide-Area Read-Write Object Repository for the Internet*. Principal Investigator. (80% for Jagannathan).

Industry Support

1. Samsung Corporation, \$100,000. 2011. *Programming Next Generation MIMD Processors*, Principal Investigator. (100% for Jagannathan)
2. Intel Corporation. \$114,000. 2011. *Profile-Guided Speculation for Non-Coherent Multicore Architectures*. Principal Investigator. (75% for Jagannathan)
3. Microsoft Corporation. \$349,500. 2010. *Language and Runtime Support for Safe and Scalable Programs*. co-PI. (25% for Jagannathan).
4. Microsoft Corporation. \$50,000. 2005. *Compiler Backend Experimentation and Extensibility Using Phoenix*. Principal Investigator.
5. NEC Labs, America. \$175,350. 2005. *Distributed Storage Systems Research*. Principal Investigator.
6. Cisco Systems, \$70,000, 2004. *Investigation of Storage over IP Protocols*. Principal Investigator.

Other Sources

1. Purdue Research Foundation. \$17,059, September 2011 – January 2013. *Programming Non-Coherent Memory*. Principal Investigator.

2. Purdue Research Foundation. \$29,672, March 2004 – August 2006. *Programm Analysis for Resource-Aware Compilation*. Principal Investigator.
3. Purdue Research Foundation. \$12,939, July 2005 – July 2006. *Knowledge Engineering Application in Chemical Reaction*. Principal Investigator.

Patents

1. Distributed Agent Software System and Method Having Enhanced Process Mobility and Communication in a Computer Network. Received December 2002. Patent number 6,496,871.
2. Transmission of Higher-Order Objects Across a Network of Heterogeneous Machines. Received September 1998. Patent number 5,745,703.
3. Software Architecture for Control of Highly-Parallel Computer Systems. Received November, 1997. Patent number 5,692,193.

Software

1. **DOrder**: An automatic learning-based invariant generator for OCaml.
Available at <http://www.https://github.com/rowangithub/DOrder>
2. **Quelea**: A Haskell library for declarative programming of eventually consistent NoSQL data stores.
Available at <http://gowthamk.github.io/Quelea/index.html>
3. **Catalyst**: A dependent type system for MLton supporting relational type specifications.
Available at <https://github.com/tycon/catalyst/tree/master/catalyst>
4. **MultiMLton**: An extension of the MLton whole-program compiler for SML targeting scalable multicore platforms.
Available at <http://multimlton.cs.purdue.edu/mML/Welcome.html>
5. **CompCertTSO**: A verified C compiler for x86 multiprocessors.
Available at <http://www.cl.cam.ac.uk/~pes20/CompCertTSO>.
6. **MLton**: A whole-program optimizing compiler for Standard ML.
Available at <http://www.mlton.org>.
7. **Mobidget**: An Agent-Based System for Mobile Java Applications.
Currently marketed and supported by NEC America and Hewlett-Packard.

Professional Activities

Editor, Computer Languages: Systems & Structures. 2010–present.

Associate Editor, International Journal of Parallel Programming. 1993–present.

Associate Editor, Advances in Software Engineering, 2008–2013.

Steering Committee: ACM Principles of Programming Languages (chair: 2014), ACM Transact (2006-20014).

Program Committees:

ACM International Conference on Functional Programming, (2016), ERC.
 Foundations of Software Engineering, Visions Track (2016).
 International Conference on Trends in Functional Programming (2016).
 Workshop on Off-the-Beaten-Track (2016).
 Workshop on Probabilistic Programming Semantics (2016).
 Workshop on Machine Learning and Programming Languages (2015).
 ACM Workshop on ML (2015).
 ACM Conference on Programming Language Design and Implementation (2014), ERC.
 ACM Symposium on Principles of Programming Languages - General Chair, (2014).
 ACM Workshop on Functional High-Performance Computing (2013).
 International Conference on Runtime Verification (2013).
 ACM Conference on Trends in Functional Programming (2013).
 ACM Conference on Programming Language Design and Implementation (2013).
 ACM Conference on Architectural Support for Programming Languages and Operating Systems (2013), ERC.
 ETAPS Symposium on Programming (2012,2013).
 ACM Workshop on Scheme (2012).
 Symposium on Runtime Verification (2012).
 Static Analysis Symposium (2012).
 Workshop on Software Foundations for Multicore Architectures (2012).
 ACM Workshop on Deterministic Parallelism (2012).
 ACM Symposium on Principles of Programming Languages (2012).
 IEEE International Symposium on Reliable Distributed Systems (2011).
 ACM Workshop on Program Analysis for Software Tools and Engineering (2011).
 ACM International Conference on Functional Programming (2011).
 ACM Conference on Programming Language Design and Implementation (2011).
 ACM SIGPLAN TRANSACT Workshop, General Chair (2011).
 ETAPS Conference on Compiler Construction (2010).
 ACM Conference on Object-Oriented Programming Languages, Systems, and Applications (2009).
 International Static Analysis Symposium (2009).
 IEEE International Conference on High Performance Computing and Communications (2008).
 IEEE International Conference on Parallel and Distributed Systems (2008).
 International Workshop on Declarative Aspects of Multicore Programming (2007, 2008).
 Asian International Symposium on Programming Languages and Systems (2007).
 ACM SIGPLAN Workshop on Languages, Compilers and Hardware Support for Transactional Computing (2007).
 ACM International Conference on Functional Programming (2006).
 ACM SIGPLAN Workshop on Languages, Compilers and Hardware Support for Transactional Computing, PC chair (2006).
 IEEE Future Directions of Distributed Computing Systems (2004).
 ACM Workshop on Continuations (2004).
 IEEE International Parallel and Distributed Processing Symposium (2002).
 ACM SIGPLAN Conference on Practical Applications of Java. (2000).

ACM Conference on Semantics, Application, and Implementation of Program Generation (2000).
 International Conference on Coordination Languages and Models (1999).
 ACM Conference on Partial Evaluation and Program Manipulation (1997).
 ACM Conference on Reflection and Meta-Level Architectures (1996).
 Parallel Architectures and Languages, Europe (1996).
 ACM International Conference on Reflection (1996).
 ACM Scheme Workshop. (1996)
 DIMACS Workshop on Specification of Parallel Algorithms, May 1994. (chair)

Reviewer for:

Journal of the ACM
 ACM Transactions on Programming Languages and Systems.
 ACM Transactions on Sensor Networks.
 IEEE Transactions on Software Engineering.
 IEEE Transactions on Computers.
 Journal of Functional Programming.
 Journal of Parallel and Distributed Systems.
 International Journal of Parallel Programming.
 ACM Conference on Principles of Programming Languages.
 ACM Conference on Programming Language Design and Implementation.
 ACM International Conference on Functional Programming.

Organizer, Schloss-Dagstuhl meeting on *Language Based Verification Tools for Functional Programs*, 2016.

Committee member, SIGPLAN Outstanding Dissertation Award, 2008, 2014, 2015.

NSF Workshop on High-Level Programming Models for Parallelism, Chair, 2013.

Committee Member, ETS GRE Computer Science Committee (2012 - 2014).

Initiated and organized three biennial summer schools on *Trends in Concurrency*, held in 2006 (Bertinoro, Italy), 2008 (Prague, Czech Republic), and 2010 (Bangalore, India).

National Science Foundation proposal panelist (2000–2012).

Invited panelist at DARPA ISAT Meeting (2009,2010).

Invited panelist at ACM Workshop on Foundations of Object-Oriented Languages (2000).

Invited panelist at ACM/CRA Symposium on Strategic Directions in Computer Science (1997).

Invited consultant to the United Nations Development Program on Information Technology (1994).

Selected Invited Talks

1. *The Promise and Rationale for MUSE: Mining and Understanding Software Enclaves*, 13th International Conference on Mining Software Repositories, (keynote), May 2016.

2. *Relational Refinement Types for Higher-Order Shape Transformers*, Foundations of Software Technologies and Theoretical Computer Science (keynote), December 2015.
3. *Declarative Programming for Eventually Consistency*, Max Plank Institute, November 2015.
4. *Declarative Programming for Eventually Consistency*, Northeastern University, November 2015.
5. *MUSE: Mining and Understanding Software Enclaves*, Schloss-Dagstuhl meeting on Programming with Big Code, November 2015.
6. *Probabilistic Programming Advancing Machine Learning*, Workshop on Machine Learning for Programming Languages (ML4PL), July 2015.
7. *Taming Uncertainty, Scale, and Change: A Programming Languages Perspective*, ECOOP Summer School, July 2015.
8. *Taming Uncertainty, Scale, and Change: A Programming Languages Perspective*, Boston University, April 2015.
9. *Taming and Exploiting Big Data: A Programming Languages Perspective*, SUNY Buffalo, Nov. 2014. (Distinguished Lecture)
10. *Atomicity Refinement for Verified Compilation*, Workshop on Exploiting Concurrency Correctly, July 2014. (Keynote)
11. *Atomicity Refinement for Verified Compilation*, York University, June 2014.
12. *Mining and Understanding Software Enclaves*, Conference on High Confidence Software Systems, May 2014.
13. *Atomicity Refinement for Verified Compilation*, Johns Hopkins University, September 2013.
14. *Atomicity Refinement for Verified Compilation*, University of Maryland, September 2013.
15. R_x^{CML} : *A Prescription for Safely Relaxing Synchrony*, Indiana University, April 2013. (Distinguished Lecture)
16. *A Relational Framework for Higher-Order Shape Analysis*, Workshop on Higher-Order Program Analysis, June 2013.
17. *Plan B: A Buffered Memory Model for Java*, Indian Institute of Technology, Chennai, India, December 2012.
18. *Plan B: A Buffered Memory Model for Java*, Indian Institute of Science, Bangalore, India, December 2012.
19. *Lightweight and Compositional Dependent Type Inference of ML*, Harvard University, Cambridge, MA, September 2012.
20. *Open Issues in Asynchronous Programming*, International Workshop on Languages for the Multicore Era, Beijing, China, June 2012.
21. *Composable Asynchronous Events*, Microsoft Research, India, December, 2011.
22. *Using Proofs-from-Tests to Verify Higher-Order Programs*, Nii Shonan Meeting on Higher-Order Verification, September, 2011.
23. *Composable Asynchronous Events*, Cambridge University, May 2011.
24. *Composable Asynchronous Events*, INRIA, Paris May 2011.
25. *Isolating Determinism in Multithreaded Programs*, Microsoft Research Workshop on Chemistry of Concurrency, February, 2011.

26. *Relaxed Memory Concurrency and Verified Compilation*, Infosys, February, 2011.
27. *Speculation, Communication, and Introspection for Multicore Programming*, Infosys, May 2010.
28. *Serializability Enforcement for Concurrent ML*, INRIA Rocquencourt, November 2009.
29. *Serializability Enforcement for Concurrent ML*, Cambridge University, November 2009.
30. *Relaxed Synchronization and Eager Scheduling for Map Reduce*, National Science Foundation, October 2009.
31. *Dissecting Transactional Semantics for Fun and Profit*, UPMARC Summer School on Multicore Computing, Uppsala University, June 2009.
32. *Partial Memoization of Concurrency and Communication*, Microsoft Research, June 2009.
33. *Making the Future Safe for the Multicore Era*, ECOOP 2008 Summer School, July 2008.
34. *Quasi-Static Scheduling for Safe Futures*, FORTH, University of Crete, June 2008.
35. *Speculating about the Future*, Trends in Concurrency Summer School, June 2008.
36. *Memoizing Communication*, Microsoft Research India, March 2008.
37. *Quasi-Static Scheduling for Safe Futures*, Indian Institute of Science, March 2008.
38. *Quasi-Static Scheduling for Safe Futures*, Intel, January 2008.
39. *Memoizing Communication*, Toyota Technology Institute, November, 2007.
40. *A Computational Infrastructure for Experimentation on Relaxed Concurrency Abstractions and their Application*, NSF Computing Resource Infrastructure Workshop, June 2007.
41. *Weaving Atomicity Through Dynamic Dependence Tracking*, NSF Next Generation Software Workshop, April 2007.
42. *Transactions Considered Harmful*, Workshop on Transactional Memory and Programming Technologies, IBM, March 2007.
43. *Stabilizers: A Modular Checkpointing Abstraction for Concurrent Programs*, Indian Institute of Information Technology, Bangalore, March 2007.
44. *Safely Relaxing Isolation for Software Transactions*, Intel Research, February 2007.
45. *Implementation Techniques for Transactions*, Trends in Concurrency Summer School, Bertinoro, Italy, July 2006.
46. *Dissecting Transactional Implementations for Fun and Profit*, Microsoft Research, June 2006.
47. *Stabilizers: Modular Checkpoint and Recovery for Concurrent Programs*, Ecole Polytechnique Fédérale de Lausanne (EPFL), May 2006.
48. *Dynamic Aspects for Runtime Fault Determination and Recovery*, NSF Next Generation Software Workshop, Rhodes, Greece, April 2006.
49. *Stabilizers: A Safe Checkpointing Abstraction for Concurrent Functional Programs*, Declarative Languages for Multicore Architectures, Intel, January 2006.
50. *Dissecting Transactional Implementations for Fun and Profit*, New England Programming Languages and Systems, Brown University, October, 2005. **Keynote presentation.**
51. *Sieve: A Tool for Automatically Detecting Variations in Program Versions*, IFIP 2.4 Annual Meeting, October, 2005.
52. *Transactional Monitors for Java*, Microsoft Corporation, June 2005.

53. *Safe Futures for Java*, Dagstuhl Workshop on Types and Tools for Applications, June 2005.
54. *Transactional Support for Permissive Monitors*, University of Maryland, October, 2004.
55. *Permissive Monitors*, DiCapo Meeting, Purdue University, July 2004.
56. *Transactional Support for Coordination Languages*, University of Pisa, February 2004.
57. *VI-Based Communication for Database Storage*, Purdue University, April 2002.
58. *User-Level Communication Protocols for Distributed Storage Systems*, NEC Laboratories, March 2002.
59. *MLton: A Whole-Programming Optimizing Compiler for Standard ML*, University of Toronto, March 2002.
60. *User-space Communication for Next-Generation Database Storage*, Professional Association of SQL Servers, June 2001. **Keynote presentation.**
61. *Communication-Passing Style*, ACM Workshop on Continuations, January 2001. **Keynote presentation.**
62. *Flow-Directed Closure Conversion*, AT&T Bell Laboratories, September 1998.
63. *Distributing Control*, NEC Research Symposium, May 1998. **Keynote presentation.**
64. *MLton: A Whole-Program Optimizing Compiler for Standard ML*, University of Pennsylvania, October 1998.
65. *Flow-Directed Compilation*, Georgia Tech., March 1998.
66. *First-Class Distributed Objects*, UC Berkeley, May 1998.
67. *Must-Alias Analysis for Higher-Order Languages*, Indian Institute of Science, November 1997.
68. *Flow-Directed Inlining*, Princeton University, April 1996.
69. *Higher-Order Distributed Objects*, Lotus Development Corporation, August 1995.
70. *Higher-Order Distributed Objects*, Yale University, August 1995.
71. *Effective Flow-Analysis for Avoiding Runtime Checks*, INRIA Rocquencourt, May 1995.
72. *Effective Flow-Analysis for Avoiding Runtime Checks*, Aarhus University, May 1995.
73. *A Unified Framework for Parallel and Distributed Computing*, NEC C&C Laboratories, January 1994.
74. *A Customizable Substrate for Concurrent Programming*, Rutgers University, February 1992.
75. *Sting: A Virtual Machine Framework for Concurrency*, UCLA, January 1992.

University and Departmental Service

1. School of Science Dean Search, 2008.
2. School of Science Faculty Council, 2005-2007.
3. School of Science Area Promotions Committee, 2006-2008. 2011-2013.
4. Department of Computer Science Head Search, 2007-2009. 2011-2012.
5. Departmental Faculty Search, 2004-2009, 2010-2013. (Hiring Chair 2007, 2008, 2010-2013)
6. Graduate Admissions Committee Chair, 2002-2004. (Chair 2003).

Teaching

At Purdue

1. **CS 565**: Graduate Programming Languages, Fall 2003, Spring 2005, Spring 2006, Spring 2007, Spring 2009, Fall 2010, Fall 2011, Spring 2013.
2. **CS 502**: Graduate Compiler Design, Spring 2003, Spring 2004, Fall 2006, Fall 2008.
3. **CS 456**, Undergraduate Programming Languages, Fall 2005.
4. **CS 390**: Undergraduate Principles of Concurrency, Spring 2008, Spring 2012.
5. **CS 352**: Undergraduate Compilers, Fall 2007.
6. **CS 240**, C Programming, Spring 2011, Fall 2012.

Prior to Purdue

New York University

Adjunct Faculty

Honors Compilers (graduate) and Programming Languages (graduate seminar).

New York City

1994 – 1997

Indian Institute of Science

Visiting Senior Faculty

Distributed Systems and Programming Languages (graduate).

Bangalore, India

1997

Yale University

Instructor

Programming Linguistics (graduate).

New Haven, CT

1991