

Santosh Ganapati Nagarakatte

santosh.nagarakatte@cs.rutgers.edu

<http://www.cs.rutgers.edu/~santosh.nagarakatte/>

Research Interests

My research interests span programming languages, compilers, and computer architecture. I am also interested in software engineering, security, and energy-efficient computing. Ongoing projects include building verified compilers, verifying implementations of cryptographic software, programming and debugging parallel programs and FPGAs.

Education

- **PhD, Computer Science**, University of Pennsylvania, 2012
- **MS, Computer Science**, Indian Institute of Science, 2007
- **BE, Computer Engineering**, National Institute of Technology Karnataka, 2005

Employment

- **Undergraduate Program Director of CS**, Rutgers University, New Brunswick, July 2020-
- **Associate Professor**, Rutgers University, New Brunswick, July 2018-
- **Assistant Professor**, Rutgers University, New Brunswick, January 2013- June 2018.
- **Visiting Fellow**, Princeton University, Princeton, Spring 2016.
- **Research Visitor/Intern**, Microsoft Research, Redmond, October 2012-December 2012.
- **Research Intern**, Microsoft Research Redmond, June 2009-September 2009.

Honors and Awards

- *ACM SIGPLAN PLDI 2021 Distinguished Paper Award*: High Performance Correctly Rounded Math Libraries for 32-bit Floating Point Representations.
- *Distinguished Alumnus Award* by NITK Surathkal, August 2020.
- *Facebook Networking Systems Research Award*: “Safe and Efficient Packet Processing with Verification of ebpf Programs” with Srinivas Narayana Ganapathy. September 2019.
- *Adviser of David Menendez’s dissertation that won the ACM SIGPLAN John C. Reynolds Outstanding Dissertation Award*.
- *CACM Research Highlights Paper 2018*: Practical Verification of Peephole Optimizations with Alive.
- *Computing Review’s Notable Articles and Books of 2016*: Termination-Checking for LLVM Peephole Optimizations.
- *ACM SIGPLAN Research Highlights Paper 2016*: Provably Correct Peephole Optimizations with Alive.
- *ACM SIGSOFT ICSE 2016 Distinguished Paper Award*: Termination-Checking for LLVM Peephole Optimizations.
- *ACM SIGPLAN PLDI 2015 Distinguished Paper Award*: Provably Correct Peephole Optimizations with Alive.
- *NSF CAREER Award*: “Semantics, Abstractions, and Tools for a Pragmatic Verified LLVM Compiler” (2015-2019).
- *Intel Corporation Gift*: Project on “Verifying Floating Point Optimizations” (2017).
- *Intel Corporation Gift*: Project on “Data Driven Precondition Inference for Peephole Optimizations” (2016).
- *Intel Corporation Gift*: Project on “Lightweight Formal Methods for LLVM Verification” (2015).
- *Intel Corporation Gift*: Project on memory safety with SoftBoundCETS and Intel MPX (2014).
- *Google Research Award*: Project on Verified Compilers (2014).
- *IEEE MICRO Top Picks paper*: Hardware-Enforced Comprehensive Memory Safety (2013).
- *IEEE MICRO Top Picks paper*: iCFP: Tolerating Cache Misses in Inorder Processors (2010).
- *Philips Research Fellowship* awarded by Philips Research and IISc.
- *University Gold Medal* by NIT Karnataka Surathkal awarded to the academic topper in CS.

Scientific Publications

Journal Articles

1. *One Polynomial Approximation to Produce Correctly Rounded Results of an Elementary Function for Multiple Representations and Rounding Modes*. Jay Lim and Santosh Nagarakatte. Proceedings of the ACM Programming Languages (PACMPL), 49th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL-2022) issue, Philadelphia, USA, January 16-22, 2022.
2. *An Approach to Generate Correctly Rounded Math Libraries for New Floating Point Variants*. Jay P Lim, Mridul Aanjaneya, John Gustafson, and Santosh Nagarakatte. Proceeding of the ACM Programming Languages (PACMPL), 48th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL-2021) issue, Jan 17-22, 2021.
3. *Practical Verification of Peephole Optimizations with Alive*. Nuno Lopes, David Menendez, Santosh Nagarakatte, and John Regehr. In Communication of the ACM, Research Highlights, Volume 61, Issue 2, Pages 84-91, February 2018
4. *Hardware-Enforced Comprehensive Memory Safety*. Santosh Nagarakatte, Milo M K Martin, and Steve Zdancewic. In IEEE Micro Top Picks of Architecture Conferences of 2012, May/June, 2013.
5. *iCFP: Tolerating All Level Cache Misses in In-Order Processors*. Andrew Hilton, Santosh Nagarakatte, and Amir Roth. In IEEE Micro Top Picks of Architecture Conferences of 2009, January, 2010.

Conference and Workshop Papers

6. *Parallel Shadow Execution to Accelerate the Debugging of Numerical Errors*. Sangeeta Chowdhary and Santosh Nagarakatte Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE-2021), Athens, Greece, August 23-28, 2021.
7. *High Performance Correctly Rounded Math Libraries for 32-bit Floating Point Representations*. Jay P. Lim and Santosh Nagarakatte. Proceedings of the 2021 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI-2021), June 20-26, 2021. 16 pages.
8. *On-the-fly Data Race Detection with the Enhanced OpenMP Series-Parallel Graph*. Nader Boushehrinejadmoradi, Adarsh Yoga, and Santosh Nagarakatte. Proceedings of the 16th International Workshop on OpenMP (IWOMP-2020), Austin, Texas, September 21-24, 2020. 16 pages.
9. *Debugging and Detecting Numerical Errors in Computation with Posits*. Sangeeta Chowdhary, Jay P. Lim, and Santosh Nagarakatte. Proceedings of the 2020 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI-2020), London, UK, June 15-20, 2020. 16 pages.
10. *Approximating Trigonometric Functions for Posits using the CORDIC Method*. Jay P. Lim, Matan Shachnai, and Santosh Nagarakatte. Proceedings of the ACM International Conference on Computing Frontiers (CF-2020), Sicily, Italy, May 11-13, 2020. 9 pages.
11. *Parallelism-centric What-If and Differential Analyses*. Adarsh Yoga and Santosh Nagarakatte. Proceedings of the 2019 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI-2019), Phoenix, Arizona, USA, June 22-26, 2019. 17 pages. Pages: 485–501.
12. *Automatic Equivalence Checking for Assembly Implementations of Cryptography Libraries*. Jay P Lim and Santosh Nagarakatte. Proceedings of the International Symposium on Code Generation and Optimization (CGO-2019), Washington DC, USA, February 16-19, 2019. 13 pages. Pages: 37-49.
13. *A Parallelism Profiler with What-If Analyses for OpenMP Programs*. Nader Boushehrinejadmoradi, Adarsh Yoga, and Santosh Nagarakatte. To appear in the Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC-2018), Dallas, Texas, November 2018. 14 pages.

14. *Compiler Optimizations with Retrofitting Transformations: Is there a Semantic Mismatch?* Jay P Lim, Vinod Ganapathy, and Santosh Nagarakatte. Proceedings of the ACM SIGSAC Workshop on Programming Languages and Analysis for Security (PLAS 2017), Dallas, Texas, October, 2017.
15. *A Fast Causal Profiler for Task Parallel Programs.* Adarsh Yoga and Santosh Nagarakatte. Proceedings of the 25th International Symposium on the Foundations of Software Engineering (FSE), 2017.
16. *ALIVE-INFER: Data-Driven Precondition Inference for Peephole Optimizations in LLVM.* David Menendez and Santosh Nagarakatte. Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2017.
17. *Parallel Data Race Detection for Task Parallel Programs with Locks.* Adarsh Yoga, Santosh Nagarakatte, and Aarti Gupta. Proceedings of the 24th International Symposium on the Foundations of Software Engineering (FSE), 2016.
18. *Alive-FP: Automated Verification of Floating Point Based Peephole Optimizations in LLVM.* David Menendez, Santosh Nagarakatte, and Aarti Gupta. Proceedings of the 23rd Static Analysis Symposium (SAS), 2016.
19. *Termination-Checking for LLVM Peephole Optimizations.* David Menendez and Santosh Nagarakatte. Proceedings of the 38th International Conference on Software Engineering (ICSE), 2016.
20. *Atomicity Checking for Task Parallel Programs.* Adarsh Yoga and Santosh Nagarakatte. Proceedings of the 2016 International Conference on Code Generation and Optimization (CGO), 2016.
21. *Testing Native Cross-Platform Mobile App Development Frameworks.* Nader Boushehrinejadmoradi, Vinod Ganapathy, Santosh Nagarakatte, and Liviu Iftode. Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015.
22. *Provably Correct Peephole Optimizations with Alive.* Nuno Lopes, David Menendez, Santosh Nagarakatte, and John Regehr. Proceedings of the 36th Annual ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2015.
23. *Everything You Want to Know about Pointer-Based Checking.* Santosh Nagarakatte, Milo M K Martin, and Steve Zdancewic. Proceedings of the First Summit on Advances in Programming Languages (SNAPL), 2015.
24. *ApproxHadoop: Bringing Approximations to MapReduce Frameworks.* Inigo Goiri, Ricardo Bianchini, Santosh Nagarakatte, and Thu Nguyen. Proceedings of the 20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2015
25. *Hardware-Accelerated Compiler-Based Pointer Checking.* Santosh Nagarakatte, Milo M K Martin and Steve Zdancewic. Proceedings of the 2014 International Symposium on Code Generation and Optimization (CGO), 2014.
26. *Ironclad C++: A Library-Augmented Type-Safe Subset of C++.* Christian Delozier, Richard Eisenberg, Santosh Nagarakatte, Peter-Michael Osera, Milo M K Martin and Steve Zdancewic. Proceedings of the 28th ACM Conference on Object-Oriented Programming, Systems and Applications (OOPSLA), 2013.
27. *Formal Verification of SSA-Based Optimizations for LLVM.* Jianzhou Zhao, Santosh Nagarakatte, Milo M K Martin and Steve Zdancewic. Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2013.
28. *Watchdog: Hardware Support for Safe and Secure Manual Memory Management.* Santosh Nagarakatte, Milo M K Martin and Steve Zdancewic. Proceedings of the 39th International Symposium on Computer Architecture (ISCA), 2012.

29. *Multicore Acceleration of Priority-Based Schedulers for Concurrency Bug Detection*. Santosh Nagarakatte, Sebastian Burckhardt, Milo M K Martin and Madanlal Musuvathi. Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2012.
30. *Formalizing the LLVM Intermediate Representation for Verified Program Transformations*. Jianzhou Zhao, Santosh Nagarakatte, Milo M K Martin and Steve Zdancewic. Proceedings of the 39th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL), 2012.
31. *A Randomized Scheduler with Probabilistic Guarantees of Finding Bugs*. Sebastian Burckhardt, Pravesh Kothari, Madanlal Musuvathi, and Santosh Nagarakatte. In 15th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2010.
32. *CETS: Compiler Enforced Temporal Memory Safety for C*. Santosh Nagarakatte, Jianzhou Zhao, Milo M K Martin and Steve Zdancewic. Proceedings of the ACM SIGPLAN International Symposium on Memory Management (ISMM), 2010.
33. *SoftBound: Highly Compatible and Complete Spatial Memory Safety for C*. Santosh Nagarakatte, Jianzhou Zhao, Milo M K Martin and Steve Zdancewic. Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2009.
34. *iCFP: Tolerating All Level Cache Misses in In-Order Processors*. Andrew Hilton, Santosh Nagarakatte, and Amir Roth. Proceedings of the 15th International Conference on High Performance Computer Architecture (HPCA), 2009.
35. *Register Allocation and Optimal Spill Code Scheduling in Software Pipelined Loops using 0-1 Integer Linear Programming Formulation*. Santosh Nagarakatte and R Govindarajan. Proceedings of the 16th International Conference on Compiler Construction (CC), 2007.

Technical Reports

36. *Semantics, Verification, and Efficient Implementations for Tristate Numbers*. Harishankar Vishwanathan, Matan Shachnai, Srinivas Narayana, and Santosh Nagarakatte Department of Computer Science, Rutgers University Technical Report DCS-TR-755, May 2021
37. *RLIBM-32: High Performance Correctly Rounded Math Libraries for 32-bit Floating Point Representations*. Jay P Lim and Santosh Nagarakatte Department of Computer Science, Rutgers University, Technical Report DCS-TR-754, April 2021.
38. *A Novel Approach to Generate Correctly Rounded Math Libraries for New Floating Point Representations*. Jay P Lim, Mridul Aanjaneya, John Gustafson, and Santosh Nagarakatte. Department of Computer Science, Rutgers University, Technical Report DCS-TR-753, July 2020.
39. *Synergistic CPU-FPGA Acceleration of Sparse Linear Algebra*. Mohammadreza Soltaniyeh, and Richard P. Martin, and Santosh Nagarakatte. Department of Computer Science, Rutgers University, Technical Report DCS-TR-750, April 2020.
40. *A Fast Causal Profiler for Task Parallel Programs*. Adarsh Yoga and Santosh Nagarakatte. Department of Computer Science, Rutgers University, Technical Report DCS-TR-728, May 2017.
41. *Precondition Inference for Peephole Optimizations in LLVM*. David Menendez and Santosh Nagarakatte. Department of Computer Science, Rutgers University, Technical Report DCS-TR-727, November 2016.
42. *Alive-FP: Automated Verification of Floating Point Based Peephole Optimizations in LLVM*. David Menendez, Santosh Nagarakatte, and Aarti Gupta. Department of Computer Science, Rutgers University, Technical Report DCS-TR-723, April 2016.

43. *Termination Checking for LLVM Peephole Optimizations*. David Menendez and Santosh Nagarakatte. Department of Computer Science, Rutgers University, Technical Report DCS-TR-718, August 2015.
44. *ApproxHadoop: Bringing Approximations to MapReduce Frameworks*. Inigo Goiri, Ricardo Bianchini, Santosh Nagarakatte, and Thu Nguyen. Department of Computer Science, Rutgers University, Technical Report DCS-TR-709, August 2014.
45. *Ironclad C++: A Library-Augmented Type-Safe Subset of C++*. Christian DeLozier, Richard Eisenberg, Santosh Nagarakatte, Peter-Michael Osera, Milo M. K. Martin and Steve Zdancewic. University of Pennsylvania Technical Report MS-CIS-13-05, March 2013.
46. *A Randomized Scheduler with Probabilistic Guarantees of Finding Bugs*. Sebastian Burckhardt, Pravesh Kothari, Madanlal Musuvathi, Santosh Nagarakatte Microsoft Research Technical Report MSR-TR-2010-3, January 2010.
47. *SoftBound: Highly Compatible and Complete Spatial Memory Safety for C*. Santosh Nagarakatte, Jianzhou Zhao, Milo M K Martin, Steve Zdancewic. University of Pennsylvania Technical Report MS-CIS-09-01, January 2009.

Patents

- *Concurrency Software Testing with Probabilistic Bounds on Finding Bugs*. Sebastian Burckhardt, Pravesh Kothari, Madanlal Musuvathi and Santosh Ganapati Nagarakatte, US Patent : 8930907, Filing Date: December 1, 2009, Application number: 12/628,223, Publication date: 01/06/2015.

Theses

- *Practical Low-Overhead Enforcement of Memory Safety for C Programs*. Santosh Nagarakatte, PhD Dissertation, University of Pennsylvania, Philadelphia, PA, Supervised by Professor Milo M K Martin.

Research Grants

1. *Techniques for Generating Correctly Rounded Math Libraries*. PI: Santosh Nagarakatte. Co-PI: Mridul Aanjaneya. National Science Foundation's Software Hardware Foundation small grant. Award number: 2110861 (see https://nsf.gov/awardsearch/showAward?AWD_ID=2110861). Amount: \$ 499,979. Duration: 2021-2024.
2. *Formally Verified Sandboxing for Packet-Processing Programs*. Co-PI: Santosh Nagarakatte. PI: Srinivas Narayana. National Science Foundation's Formal Methods in the Field Track I grant. Award number: 2019302 (see https://nsf.gov/awardsearch/showAward?AWD_ID=2019302). Amount: \$749,356. Duration: 2020-2024.
3. *NSF REU Supplement for CAREER:Semantics, Abstractions, and Tools for a Pragmatic Verified LLVM Compiler*. PI: Santosh Nagarakatte. Rutgers University. Amount: \$16,000. 2020.
4. *Formalisms, Implementations, and Verification Procedures for Alternatives to Floating Point*. PI: Santosh Nagarakatte. National Science Foundation's Software Hardware Foundation small grant. Award number: 1908798 (see https://nsf.gov/awardsearch/showAward?AWD_ID=1908798). Amount: \$500,000. Duration: 2019-2022.
5. *Automated Verification for Assembly Implementations of Cryptography Libraries*. PI: Santosh Nagarakatte. National Science Foundation's Formal Methods in the Field Track-II grant. Award number: 1917897 (see https://nsf.gov/awardsearch/showAward?AWD_ID=1917897). Amount: \$99,996. Duration: 2019-2021.
6. *Safe and Efficient Packet Processing with Verification of eBPF programs*. Co-PI: Santosh Nagarakatte, PI: Srinivas Narayana. Facebook Networking Systems Research Award. \$50,000. September 2019.
7. *Semantics, Abstractions, and Tools for a Pragmatic Verified LLVM Compiler*. PI: Santosh Nagarakatte, Rutgers University, **NSF CAREER Award**. Award number: 1453086 (see http://www.nsf.gov/awardsearch/showAward?AWD_ID=1453086). Amount: \$545000.

8. *Pragmatic abstractions, techniques, and tools for LLVM verification*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2017. Amount :\$75000.
9. *Pragmatic abstractions, techniques, and tools for LLVM verification*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2016. Amount :\$75000.
10. *NSF TC: WATCHDOG: Hardware-Assisted Prevention of All Use-After-Free Security Vulnerabilities*. PI: Santosh Nagarakatte. Rutgers University. Amount: \$320,538. The grant is sub-contracted to Rutgers by the University of Pennsylvania.
11. *Lightweight Formal Methods for LLVM Verification*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2015. Amount :\$75000.
12. *Memory Safety with Intel MPX and SoftBoundCETS*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2014 . Amount :\$70000.
13. *Hardware support for OS Integrity*. PI: Santosh Nagarakatte, Former PI's: Vinod Ganapathy and Liviu Iftode, Rutgers University, NSF STARSS program. Award number:1441724 (see http://www.nsf.gov/awardsearch/showAward?AWD_ID=1441724). Amount: \$499988.
14. *Semi-Automated Verification of LLVM Optimizations using SMT Solvers*. PI: Santosh Nagarakatte, Rutgers University, **Google Research Award**. Amount: \$63500.
15. *iSpider: Automated Multithreaded Program Testing and Debugging Platform*. PI: Santosh Nagarakatte, Rutgers University **Research Council Grant**. Amount: \$2000.

Presentations

Presentations at Universities and Industrial Labs

- *Programming, Debugging, and Reasoning Techniques for Posits*. Invited talk at the University of Wisconsin-Madison Programming Languages Seminar Series. July, 2020.
- *A Case for Parallelism Profilers and Advisers with What-If Analyses*. Invited talk at the Workshop on Dependable and Secure Software Systems at ETH Zurich, Switzerland. October 2019.
- *A Case for Parallelism Profilers and Advisers with What-If Analyses*. Invited talk at Parasol-II Workshop at Rutgers University, Piscataway, New Jersey, May 2019.
- *A Case for Parallelism Profilers and Advisers with What-If Analyses*. Guest talk at the 3rd Summit on Advances in Programming Languages (SNAPL-2019) at Brown University, Providence, Rhode Island, May 2019.
- *Life After CIS PhD*. Panelist at the Department of Computer and Information Sciences event. University of Pennsylvania, Philadelphia, PA, February 2019.
- *Lightweight Formal Methods for LLVM Verification*. Invited talk at VMware Research, Palo Alto, California, US, June 2018.
- *Debugging and Profiling Task Parallel Programs with TaskProf*. Tutorial at PLDI 2018, Philadelphia, PA, USA, June 2018.
- *Compiler Optimizations and Retrofitting Transformations: Is there a Semantic Mismatch?* Dagstuhl Seminar 18201 on Secure Compilation , Dagstuhl, Wadern, Germany, May 2018.
- *Debugging and Profiling Task Parallel Programs with TaskProf*. A tutorial presented as part of PPOPP 2018 at Vienna, Austria, February 2018.
- *Practical Formal Methods for Mainstream Compiler Developers*, URCS Seminar, University of Rochester, Rochester, NY, October 2017.
- *A Fast Causal Profiler for Task Parallel Programs*, UWPLSE Seminar, University of Washington, Seattle, October 2017.
- *Pragmatic Abstractions, Techniques, and Tools for LLVM Verification*, Invited talk at Intel Corporation, September 2017.
- *Practical Formal Methods for Mainstream Compiler Developers*, Invited talk, CS Department, Rutgers University, Piscataway, September 2017.

- *Practical Formal Methods for Mainstream LLVM Developers*, Invited talk, TU Darmstadt, Darmstadt, Germany, September 2017.
- *Practical Formal Methods for Mainstream LLVM Developers*, Invited talk, Max Planck Institute-Software Systems (MPI-SWS), Kaiserslautern, Germany, August 2017.
- *Alive-FP: Automated Verification of Floating Point Optimizations in LLVM*, Invited talk, Dagstuhl Seminar 17352 – Analysis and Synthesis of Floating-point Programs, Dagstuhl, Wadern, Germany, August 2017
- *Data-Driven Precondition Inference for Compiler Optimizations*
 - Department of Computer Science and Automation, Indian Institute of Science (IISc), Bangalore, August 2017.
 - Microsoft Research India (MSR-India), Bangalore, July 2017.
- *Practical Formal Methods for Mainstream LLVM Developers*
 - National Institute of Technology Karnataka (NITK), Surathkal, August 2017.
 - Indian Institute of Technology Madras (IITM), Chennai, India, July 2017.
 - Intel Microarchitecture Research Laboratory (MRL), Bangalore, July 2017.
- *Lightweight Formal Methods for LLVM Verification*, MIT PL Seminar Series, CSAIL, MIT, MA, May 2017.
- *Lightweight Formal Methods for LLVM Verification*, Invited Talk Series, CS department, SUNY Binghamton, NY, October 2016
- *Pragmatic Abstractions, Techniques, and Tools for LLVM Verification*, invited talk at Intel Corporation, September 2016.
- *Lightweights Formal Methods for LLVM Verification*
 - Indian Institute of Science, Bangalore, August 2016.
 - Indraprastha Institute of Technology Delhi, Delhi, August 2016.
 - Indian Institute of Technology Delhi, Delhi, August 2016.
 - Tata Institute of Fundamental Research, Mumbai, August 2016.
 - Indian Institute of Technology Bombay, Mumbai, August 2016.
 - Microsoft Research India, Bangalore, July 2016.
- *Verifying Precise LLVM Floating Point Peephole Optimizations with Alive-FP*. Invited talk at Deep Spec Workshop, Princeton University, June 2016.
- *Lightweight Formal Methods for Verifying High Performance Compilers*
 - Princeton University, Princeton, April 2016.
 - School of Computer Science, Carnegie Mellon University, Pittsburgh, April 2016.
 - Computer Science department, University of California, Los Angeles, April 2016.
 - ECE Department, Purdue University, West Lafayette, March, 2016
- *Lightweight Formal Methods for LLVM Verification*,
 - At Programming Research Laboratory, Northeastern University, Boston, January 2016.
 - At IBM PL day, New York, November 2015.
 - At Rutgers University, November 2015.
 - At Constraints for Formal Verification Workshop, Austin, Texas, November 2015.
 - At Intel Corporation, Hillsboro, June 2015.
- *Pointer Checking with Intel MPX and SoftBoundCETS*, Invited Talk at Intel Corporation, Hillsboro, June 2015.

- *Enablers and Roadblocks for Mainstream Adoption of Approximate Computing*, Invited Panel Talk at the Second Workshop on Approximate Computing Across the Stack, Istanbul, Turkey, March 2015. Host: Viji Srinivasan.
- *Complete and Verified Memory Safety with Pointer-based Checking and Verified LLVM* at Computer Science Department, Stony Brook, New York, February 2014. Host: Don Porter.
- *Verified LLVM: Formalizing the Semantics of the LLVM IR*.
 - At Alcatel-Lucent Bell Labs, New Jersey, February 2014. Host: Kedar-Namjoshi. Part of Henry Landau seminar series.
 - At IBM Research India, Bangalore, July, 2013.
 - To program analysis working group at Microsoft Research Redmond, November 2012.
 - At LLVM Developer’s meeting at San Jose, November 2012.
- *Birds of a Feather session on Memory Safety and Automated Defenses* at LLVM Developer’s meeting at San Jose, November 2012.
- *Practical Low-overhead Enforcement of Memory Safety for C Programs*, PhD thesis defense at the University of Pennsylvania, October 2012.
- *Secure Low Level Programming with Hardware/Compiler Enforced Memory Safety*
 - At the Indian Institute of Science, Bangalore, July 2012.
 - At Microsoft Research India, Bangalore, July 2012.
 - At Microsoft Research Redmond, April 2012.
 - At Microsoft Research Silicon Valley, Mountain View, March 2012.
 - At Virginia Tech, ECE Department, Blacksburg, March 2012.
 - At Rutgers University, CS Department, New Brunswick, March 2012.
 - At the University of Colorado Boulder, Boulder, March 2012.
 - At Virginia Tech, CS Department, Blacksburg, February 2012.
- *NeedlePoint: A Framework to Expose Concurrency Bugs by Controlling Thread Interleavings*
 - At the Indian Institute of Science, Bangalore, April 2011.
 - At Systems Lunch, Penn, Philadelphia, March 2011.
- *A Randomized Scheduler for Finding Concurrency Bugs* at Microsoft Research Redmond, September 2009.
- *SoftBound: Highly Compatible and Complete Spatial Memory Safety for C*
 - At Microsoft Research India, Bangalore, December 2009.
 - At LLVM Developer’s Meeting, Cupertino, September 2009.

Conference Presentations

- *Compiler Optimizations and Retrofitting Transformations: Is there a Semantic Mismatch?* A short talk at Principles of Secure Compilation Workshop (PriSC 2018), Los Angeles, CA, January 2018.
- *Lightweight Formal Methods for LLVM Verification*, Invited Junior Researcher Presentation at Monterey, CA, SNAPL, May, 2017
- *Alive-FP: Automated Verification of Floating Point Based Peephole Optimizations in LLVM* at Edinburgh, SAS, September 2016.
- *Everything You want to Know About Pointer-Based Checking* at Monterey, CA, SNAPL, May 2015.
- *WatchdogLite: Hardware-Accelerated Compiler-Based Pointer Checking* at Orlando, CGO, February 2014.
- *Formal Verification of SSA Optimizations for LLVM* at Seattle, PLDI, June 2013.

- *Multicore Acceleration of Priority-Based Schedulers for Concurrency Bug Detection* at Beijing, PLDI, June 2012.
- *CETS: Compiler Enforced Temporal Safety for C* at Toronto, ISMM 2010.
- *SoftBound: Highly Compatible and Complete Spatial Memory Safety for C* at PLDI 2009, Dublin, Ireland.
- *Register Allocation and Optimal Spill Code Scheduling* at CC 2007, Braga, Portugal.

Teaching And Mentoring

Teaching at Rutgers University

- Fall 2020: Computer Architecture (01:198:211)
 - 197 students. Instructor Rating: 4.14/5. Course Rating: 4.01/5
- Spring 2020: Computer Security (16:198: 544)
 - 29 students. Instructor Rating: 4.77/5. Course Rating: 4.62/5
- Fall 2019: Computer Architecture (01:198:211)
 - 178 students. Instructor Rating: 4.32/5. Course Rating: 4.14/5
- Spring 2019: Computer Security (16:198: 544)
 - 21 students. Instructor Rating: 4.64/5. Course Rating: 4.36/5
- Fall 2018: Computer Architecture (01:198:211)
 - 250 students. Instructor Rating: 4.14/5. Course Rating: 4.02/5
- Spring 2018: Programming Languages and Compilers II (16:198:516)
 - 8 students. Instructor Rating: 4.67/5. Course Rating: 4.83/5
- Fall 2017: Computer Architecture (01:198:211)
 - 230 students. Instructor Rating: 4.30/5. Course Rating: 4.18/5
- Spring 2017: Computer Architecture (01:198:211)
 - 231 students. Instructor Rating: 3.87/5. Course Rating: 3.64/5
- Fall 2016: Programming Languages and Compilers I (16:198:515)
 - 12 students. Instructor Rating: 4.89/5. Course Rating: 4.78/5
- Fall 2015: Computer Architecture (01:198:211)
 - 185 students. Instructor Rating: 4.12/5. Course Rating: 4.05/5
- Spring 2015: Computer Architecture (01:198:211)
 - 155 students. Instructor Rating: 3.55/5. Course Rating: 3.93/5
- Fall 2014: Programming Languages and Compilers I (16:198:515).
 - 18 students. Instructor Rating: 4.44/5. Course Rating: 4.22/5
- Spring 2014: Programming Languages and Compilers II (16:198:516)
 - 13 students. Instructor Rating: 4.67/5. Course Rating: 4.33/5
- Fall 2013: Computer Architecture (01:198:211),
 - 148 students. Instructor Rating: 2.90/5. Course Rating: 3.10/5
- Spring 2013: Performance Aware Reliable Software for Multicore Processors, (16:198:671)
 - 5 students. Instructor Rating: 4.0/5. Course Rating: 3.75/5

Teaching at IIT Hyderabad

- Summer 2013: Mini (1-credit) course on Robust Software with Verified Programming.
 - 12 students. Instructor Ratings 9.2/10.

Graduated PhD Students

- Jay Lim. PhD, October 2021.
 - Dissertation: “Novel Polynomial Approximation Methods For Generating Correctly Rounded Elementary Functions”.
 - PhD committee: Santosh Nagarakatte (adviser), Rich Martin, Uli Kremer, Zachary Tatlock (University of Washington).
- Nader Boushehrinejad Moradi. PhD, January 2021.
 - Dissertation: “Performance Profilers and Debugging Tools for OpenMP Applications”.
 - PhD committee: Santosh Nagarakatte (adviser), Badri Nath, Srinivas Narayana, Martha Kim.
- Adarsh Yoga. PhD, October 2019.
 - Dissertation: “Parallelism-Driven Performance Analysis Techniques for Task Parallel Programs”.
 - PhD committee: Santosh Nagarakatte (adviser), Ulrich Kremer, Madan Musuvathi, Sudarsun Kannan.
- David Menendez, PhD, January 2018, Rutgers University.
 - Dissertation: “Practical Formal Techniques and Tools for Developing LLVM’s Peephole Optimizations”. **Winner of ACM SIGPLAN John C. Reynolds Outstanding Dissertation Award.**
 - PhD committee: Santosh Nagarakatte (adviser), Rajeev Alur, Uli Kremer, and Thu Nguyen.

Current Ph.D Students

- Mohammed Reza Soltaniyeh: Advising since January 2017.
- Sangeeta Chowdhary: Advising since January 2017.
- Matan Shachnai: Advising since September 2020.
- Harishankar Vishwanathan (co-advised with Srinivas Narayana): Advising since January 2021.
- Sehyeok Park: Advising since Fall 2021.

Ph.D Thesis Committees

- Shaohua Duan (advisor: Manish Parashar)
- Jonas Wagner (EPFL, Switzerland) (advisor: Professor George Candea)
- Rezwana Karim (advisor: Professor Vinod Ganapathy).
- Amruta Gokhale (advisor: Professor Vinod Ganapathy).

Ph.D Qualifying Committees

- Harsha Srimath Tirumala (advisor: Prof. Eric Allender)
- David Pham (advisor: Prof. Yongfeng Zhang)
- Yanshi Luo (advisor: Prof. Mridul Aanjaneya)
- Yujie Ren (advisor: Prof. Sudarsun Kannan)
- Guangzhi Tang (advisor: Prof. Konstantinos Michmizos)
- Gang Qiao (advisor: Vladimir Pavlovic)

- Georgiana Haldeman (advisor: Prof. Thu Nguyen)
- Daniel Bittner (advisor: Prof. Rebecca Wright)
- Kazem Cheshmi (advisor: Prof. Maryam Mehri Dehnavi)
- Cong Chen (advisor: Prof. Naftaly Minsky).
- Ioannis Manousakis (advisor: Prof. Thu Nguyen and Prof. Ricardo Bianchini).
- Benjamin Lund (advisor: Prof. Shubhangi Saraf).
- Nader Boushehri (advisor: Prof. Liviu Iftode).
- Jinjing liu (advisor: Professor Dmitris Metaxas).
- William Katsak (advisor: Prof. Thu Nguyen).
- Zakary Littlefield (advisor: Prof. Kostas Bekris).
- Hai Nguyen (advisor: Prof. Vinod Ganapathy).
- Jongpil Kim (advisor: Prof. Vladimir Pavlovic).

Undergraduate Supervision and Outreach

- Anika Kumar: Spring 2021: Project SUPER mentee on Floating Point and Posits.
- Shehneel Asharaf: Spring 2021: Project SUPER mentee on Floating Point and Posits.
- Josh Cooper: Summer 2020: NSF REU on Verification and Static Analysis.
- Sammy Berger: Summer 2020: NSF REU on Verification and Static Analysis
- Matan Shachnai: Spring 2019- Summer 2020: Posits
- Mihai Andrei: Aresty RA, Fall 2017-Spring 2018: Crypto Verification
- Steven Hsu: Aresty RA, Fall 2017-Spring 2018: Crypto Verification
- Aditya Shastri: Fall 2017-Spring 2018: Crypto Verification
- Chun Lau: Summer 2017: Undergraduate Research Participant.
- Justin Chong: Summer 2017 : Undergraduate Research Participant.
- Steve Hsu: Summer 2017 - Spring 2018 : Aresty Research Assistant.
- Mihai Andrei: Summer 2017 - Spring 2018 : Aresty Research Assistant.
- Oliver Katz: Summer 2016 - Spring 2017: Aresty Research Assistant.
- Matthew Tantoy: Fall 2016-Spring 2017: Aresty Research Assistant.
- Shreyas Hirday: Summer 2014: Aresty Summer Science program: Project on data race detector for C/C++ programs.
- Mauricio Trajano :Fall 2014: Aresty Research Assistant: Memory Safety for C/C++
- Mentored five high school students as part of NJ Governor's school of engineering and technology summer program on their project on verification of algorithms with the Dafny program verifier.

Professional Activities

Conference and Workshop Organization

- General chair: Virtual Execution Environments (VEE), 2020
- Sponsorship chair: Programming Language Design and Implementation (PLDI), 2015
- Program Co-chair: Workshop on Dynamic Analysis, WODA, 2014
- Sponsorship chair: Programming Language Design and Implementation (PLDI), 2014

Program Committee Member

- PLDI 2022: Program committee member: Programming Language Design and Implementation (PLDI), 2022.
- ASPLOS 2022: Program committee member: Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2022.
- CGO 2022: Program committee member: Code Generation and Optimization (CGO), 2022.
- National Science Foundation Panel (NSF), Panelist, 2021.
- Department of Energy Panel (DoE), Panelist, 2021.
- PriSC 2021: Program committee member: Principles of Secure Compilation, 2021.
- PLDI 2020: Program committee member: Programming Language Design and Implementation (PLDI), 2020.
- CC 2020: Program committee member: Compiler Construction, 2020.
- PLDI 2020 SRC: Program committee member: Student Research Competition. Programming Language Design and Implementation (PLDI-SRC), 2020.
- National Science Foundation Panel (NSF), Panelist, 2019.
- CGO 2019: Program committee member: Code Generation and Optimization, 2019.
- PLDI 2019: External review committee member: Programming Language Design and Implementation (PLDI), 2019.
- ASPLOS 2019: External review committee member: Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2019.
- OOPSLA Distinguished papers 2018: Distinguished paper committee: Object Oriented Programming, Systems, and Applications, 2018.
- PLDI 2018 SRC: Program committee member: Student Research Competition of PLDI 2018.
- OOPSLA 2018: Program committee member, Object-Oriented Programming, Systems, and Applications, 2018
- CC 2018: Program committee member, Compiler Construction, 2018
- ASPLOS 2018: External review committee member, Architectural Support for Programming Languages and Operating Systems, 2018
- POPL 2018: Program committee member, Principles of Programming Languages (POPL), 2018
- National Science Foundation Panel (NSF), Panelist, 2018.
- APLAS 2017: Asian Programming Languages and Systems (APLAS), 2017
- National Science Foundational Panel, 2017.
- PLDI 2017: External review committee member, Programming Language Design and Implementation (PLDI), 2017
- POPL 2017: External review committee member, Principles of Programming Languages (POPL), 2017
- CGO 2017: International Symposium on Code Generation and Optimization (CGO), 2017
- ICSE 2017: Program committee of SRC, International Conference on Software Engineering (ICSE), 2017
- ISEC 2017: Innovations in Software Engineering Conference (ISEC), 2017
- National Science Foundation Panel (NSF), 2016
- PLDI 2016: Programming Language Design and Implementation (PLDI), 2016
- ISEC 2016: 9th India Software Engineering Conference (ISEC), 2016
- PLDI 2015: Programming Language Design and Implementation (PLDI), 2015
- National Science Foundation Panel (NSF), 2015
- FCS 2015: Workshop on Foundations of Computer Security (FCS), 2015
- APLAS 2015: 13th Asian Symposium on Programming Languages and Systems (APLAS), 2015

- ISEC 2015: 8th India Software Engineering Conference (ISEC), 2015
- PERTEA 2014: 2014 International Workshop on Software and System Performance Testing, Debugging, and Analytics
- WODET 2014: Workshop on Determinism and Correctness in Parallel Programming, 2014
- OOPSLA 2014: Object-Oriented Programming, Systems, and Applications, 2014
- ASPLOS 2014: Architectural Support for Programming Languages and Operating Systems, 2014
- PPOPP 2014: External review committee member, Principles and Practice of Parallel Programming, 2014
- WODA 2013: Workshop on Dynamic Analysis, 2013

Reviewing

- Journals: ACM Transaction on Computer Systems (TOCS), IEEE Transactions on Computers (TC), IEEE Transactions on Parallel and Distributed Systems (TPDS), IEEE Transactions on Dependable and Secure Computing (TDSC), ACM Transactions on Programming Languages and Systems (TOPLAS), IEEE Transactions on Software Engineering, and ACM Transactions on Architecture and Code Optimization (TACO).
- Conferences: International Symposium on Microarchitecture (MICRO), ACM Symposium Principles of Programming Languages (POPL), International Conference on Supercomputing (ICS), International Conference on Parallel Architectures and Compilation Techniques (PACT), International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS).

University and Departmental Service

- SAS Curriculum Committee Member. Fall 2021 - current.
- SAS Scholarship and Service Award Committee Member. Fall 2021- current.
- Rutgers New Brunswick Academic Integrity Facilitator: from Fall 2020 - current.
- Faculty Mentor, LSAMP program: from Fall 2020 - current.
- Faculty Mentor, Rutgers Connection Network Mentoring Program: from Fall 2020 - current.
- Rutgers CS Undergraduate Director: from July 2020 - current.
- Rutgers CS Faculty Hiring Committee Chair: Spring 2020.
- School of Graduate Studies Academic Cluster Committee: Fall 2019 - current.
- Executive Committee: Fall 2019 - current.
- Systems Hiring Committee: Spring 2019.
- Department Honors Chair: Summer 2018- current.
- Faculty Hiring Committee: Spring 2018.
- SAS Honors Faculty Mentor: Fall 2017 - current.
- Masters Admission committee: Fall 2017.
- Faculty Hiring Committee: Spring 2017.
- MS Admissions Committee: Fall2016-Spring 2017.
- Undergraduate Advising Committee: Fall 2015.
- Started system's reading group: Fall 2014 - current.
- Colloquium Committee: Fall 2013- Spring 2015.
- Graduate Committee: Fall 2013 - Spring 2015.
- Started Rutgers Programming Languages Seminar Series with Abhishek Bhattacharjee: Spring 2013 - Summer 2014.

Professional Societies

- Senior Member of ACM and ACM SIGPLAN

Software Releases

- Faculty mentor of the RLIBM-32 project that provides math libraries with correctly rounded results for 32-bit representations (float, posit32). The tool is publicly available at <https://github.com/rutgers-apl/rlibm-32>.
- Faculty mentor of the RLIBM project that provides math libraries with correctly rounded results for many elementary functions for various data types (half precision, bfloat16, posit16). The tool is publicly available at <https://github.com/rutgers-apl/rlibm>.
- Faculty mentor of the PositDebug tool to detect and debug numerical errors in applications using the posit representation. The tool is publicly available at <https://github.com/rutgers-apl/PositDebug>.
- Faculty mentor of the FPSanitizer tool to detect and debug numerical errors in applications that use the floating point representation. The tool is publicly available at <https://github.com/rutgers-apl/fpsanitizer>.
- Faculty mentor of the CORDIC math library implementations for the posit representation. The tool is publicly available at <https://github.com/rutgers-apl/CordicWithPosits>.
- Faculty mentor of the ALIVE-NJ tool kit (<https://github.com/rutgers-apl/alive-nj>) developed by David Menendez. Alive-NJ includes reasoning about the correctness of integer and floating point based peephole optimizations. It also includes termination checking for a suite of peephole optimizations.
- Faculty mentor of the ALIVE project (<https://github.com/nunoplopes/alive>). Alive provides a domain-specific language for specifying LLVM peephole optimizations (InstCombine) and automatically generating verification conditions using Z3 and the C++ implementation. We have already discovered many bugs in the LLVM compiler. The Alive tool is actively used by LLVM developers.
- Faculty mentor of the TaskProf project (<http://github.com/rutgers-apl/taskprof>) developed by Adarsh Yoga. TaskProf profiles task parallel programs leveraging performance counters and in parallel using multicores. It also enables programmers to estimate the increase in parallelism when a region of code is optimized by some amount.
- Faculty mentor of the project on dynamic analysis for task-parallel programs (<https://github.com/rutgers-apl>). The project aims to build sound, precise, and complete dynamic analyses for task-parallel programs. The goal is to reduce the complexity of debugging task-parallel programs to the task of debugging sequential programs.
- Faculty mentor of the OpenMP profiler project (<https://github.com/rutgers-apl/omp-whip>) developed by Nader Boushehrinejadmoradi. OMP-WHIP is a parallelism profiler with What-If analyses for accurately identifying parallelism bottlenecks in OpenMP programs.
- A contributor to the Verified LLVM Project (<http://www.cis.upenn.edu/~stevez/vellvm>). Vellvm provides a framework for reasoning about programs written using the LLVM IR. Vellvm provides semantics of the LLVM IR in the Coq proof assistant.
- Primary developer of SoftBoundCETS, a memory safety checker (<http://www.cs.rutgers.edu/~santosh.nagarakatte/softbound>). Intel MPX uses an approach similar to SoftBoundCETS to enforce spatial safety.

Personal Information

- Born November 1983, India. US Citizen. Married, one son.