

# 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, programming and debugging parallel programs.

## 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

- **Assistant Professor**, Rutgers University, New Brunswick, January 2013-
- **Visiting Fellow**, Princeton University, Princeton, Spring 2016.
- **Research Visitor/Intern**, Microsoft Research, Redmond, October 2012-December 2012.
- **Research Assistant**, University of Pennsylvania, September 2007-October 2012.
- **Research Intern**, Microsoft Research Redmond, June 2009-September 2009.
- **Research Assistant**, Indian Institute of Science, Aug 2005-Aug 2007.

## Honors and Awards

- *CACM Research Highlights Paper 2017*: Provably Correct Peephole Optimizations with Alive.
- *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 “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. *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.
2. *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 Papers

1. *ALIVE-INFER: Data-Driven Precondition Inference for Peephole Optimizations in LLVM*. David Menendez and Santosh Nagarakatte. To Appear in the Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2017.
2. *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.
3. *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.
4. *Termination-Checking for LLVM Peephole Optimizations*. David Menendez and Santosh Nagarakatte. Proceedings of the 38th International Conference on Software Engineering (ICSE), 2016.
5. *Atomicity Checking for Task Parallel Programs*. Adarsh Yoga and Santosh Nagarakatte. Proceedings of the 2016 International Conference on Code Generation and Optimization (CGO), 2016.
6. *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.
7. *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.
8. *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.
9. *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.
10. *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.
11. *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.
12. *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.
13. *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.
14. *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.

15. *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.
16. *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.
17. *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.
18. *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.
19. *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.
20. *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.

## 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. *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](http://www.nsf.gov/awardsearch/showAward?AWD_ID=1453086)). Amount: \$545000.
2. *Pragmatic abstractions, techniques, and tools for LLVM verification*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2016. Amount :\$75000.
3. *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.
4. *Lightweight Formal Methods for LLVM Verification*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2015. Amount :\$75000.
5. *Memory Safety with Intel MPX and SoftBoundCETS*. PI: Santosh Nagarakatte, Rutgers University, **Intel Corporation Gift**, 2014 . Amount :\$70000.
6. *Hardware support for OS Integrity*. Co-PI: Santosh Nagarakatte, PI: Vinod Ganapathy, Rutgers University, **NSF STARSS program**. Award number:1441724 (see [http://www.nsf.gov/awardsearch/showAward?AWD\\_ID=1441724](http://www.nsf.gov/awardsearch/showAward?AWD_ID=1441724)). Amount: \$499988.

7. *Semi-Automated Verification of LLVM Optimizations using SMT Solvers*. PI: Santosh Nagarakatte, Rutgers University, **Google Research Award**. Amount: \$63500.
8. *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

- *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

- *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**

- Spring 2017: Computer Architecture (01:198:211)
  - 231 students.
- 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.
  - Ratings: 9.2/10.

## **Ph.D Candidate Supervision**

- Adarsh Yoga: 4th year Ph.D student. Advising since September 2013.
- David Menendez: 5th year Ph.D student. Advising since May 2014.
- Jay Lim: 3rd year Ph.D student. Advising since September 2014.
- Mohammed Reza Soltaniyeh: 3rd year Ph.D. student. Advising since January 2017.
- Sangeeta Chowdhary: 1st year Ph.D . student. Advising since January 2017.
- Nader Boushehrinejad Moradi. 6th year Ph.D. student. Advising since January 2017.

## **Ph.D Thesis Committee**

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

## **Ph.D Qualifying Committee**

- 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**

- Steve Hsu: Summer 2017 - : Aresty Research Assistant.
- Mihai Andrei: Summer 2017 - : 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**

- 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

- POPL 2018: Program committee member, Principles of Programming Languages (POPL), 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.
- 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

- Faculty Hiring Committee - Spring 2017.
- Masters Admission Committee - Fall 2016.
- Undergraduate Advising - Fall 2015.
- Tuesday Talks Committee - Fall 2013- 2015.
- Colloquium Committee - Fall 2013- Spring 2015
- Graduate Committee - Fall 2013 - Spring 2015.
- Started Rutgers Programming Languages Seminar Series with Abhishek Bhattacharjee - Spring 2013 -



## Professional Societies

- Member of ACM, IEEE, and ACM SIGPLAN

## Software Releases

- 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.
- 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 NeedlePoint concurrency bug detection framework. NeedlePoint is a unified framework for testing real-world multithreaded programs with a wide range of testing policies by hiding the complexities of instrumenting real world programs.
- 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. Citizen of India. Permanent Resident of USA. Married.