Sridhar Gopinath

+1 (512) 545-7642 | sridhar.g@utexas.edu | sridhargopinath.in | SridharGopinath | sridhar-gopinath

Education

University of Texas at Austin, USA ***GPA: 4.0 / 4.0***

Ph.D. in Computer Science *Aug. 2019 - Present*

- **Courses**: Program synthesis
- Teaching Assistant for Advanced Topics in Compilers (CS 380C)

Indian Institute of Science (IISc), Bengaluru, India ***GPA: 7.0 / 8.0***

Master by Research in Computer Science *Aug. 2015 - May 2017*

- **Courses**: Advanced Software Engineering | Program Analysis and Verification | Operating Systems | Algorithm Design
- **Thesis**: Efficient Whole program path tracing

Sri Jayachamarajendra College of Engineering (SJCE), Mysuru, India ***GPA: 9.0 / 10.0***

B.E. in Computer Science and Engineering *Sep. 2011 - May 2015*

- **Courses**: Design and Analysis of Algorithms | Data Structures | Databases | Computer Networks
- **Thesis**: Loop Fusion in LLVM Compiler

Languages and Technologies

Java, Python, MATLAB, LLVM, SOOT, C++, AWS, EC2

Experience

Research Fellow **Microsoft Research** *Bengaluru, India*

Project: Machine Learning on Edge devices *Microsoft/EdgeML* *Oct. 2017 - Jul. 2019*

- Developed a framework called SeeDot that compiles ML models to C code to run efficiently on KB-sized IoT devices.
- Implemented novel techniques to replace all floating-points with integers and to optimize expensive functions like e^x and SpMV.
- SeeDot outperforms hand-optimized code by up to 12× and MATLAB-generated code by up to 82×.
- Research paper published at **PLDI 2019** conference.

Research Intern **Microsoft Research** *Bengaluru, India*

Project: Finding bugs in Windows 10 Device Drivers *Boogie/Corral* *Jun 2017 - Sep. 2017*

- Implemented a property checker to detect Interrupt Request Level (IRQL) violations, which is a major cause for crashes in Windows.
- The property checker generates preconditions that are used by the Z3 theorem prover to identify defects.
- Found **26 unknown defects** in Windows 10 which have been verified by developers.
- Tool currently deployed in Windows Driver Kit (WDK) and used internally at Microsoft for testing device drivers.

Software Development Intern **Compiler Tree Technologies** *Mysuru, India*

Project: Loop Fusion in LLVM Compiler *Jan. 2015 - May 2015*

- Designed a function pass in LLVM that fuses adjacent loops to improve cache locality.
- Implemented feasibility and data dependency analyses to identify candidate loops for fusion.
- Verified the correctness of fusion and observed up to 20% performance improvement on micro-benchmarks.

Projects

Efficient Whole Program Path Tracing *(Java, SOOT, Tamiflex, Bash, DaCapo)*

- Designed a program analysis to minimize the overhead to derive the control-flow trace (whole program path) of an execution.
- Proved that the problem is a variant of the hitting-set problem, an NP-hard problem, using control-flow graph properties.
- Implemented efficient data structures and approximation algorithms to scale the analysis for large Java programs.
- Tool outperforms state-of-the-art by up to 5.4× on the DaCapo benchmark suite.

Null pointer dereference analysis *(Java, WALA, jGraph)*

- Implemented an analysis that identifies potential null pointer dereferences of objects or fields in Java programs.
- Analysis uses inter-procedural data-flow facts to identify variables that are *maybe* null.
- Designed transfer functions that handle object creation and performed pointer analysis to scale for complex Java programs.

Results analytics website - SJCEResults.com *(AWS, PHP, MySQL, HTML, CSS, C++, DOM parser)*

- Developed a website that provides deep analytics on the examination results data of around 6000 students over 3 academic years.
- Implemented name-wise search, multi-level rankings and performance analysis for each student.
- Designed outcome analysis for each course and performance analysis for each department.
- Website currently has more than **2.1 Million** page views.

PintOS Projects *(C++, Qemu, Bash)*

- Designed and implemented the following key functionalities on top of base OS.
- Threads: Priority scheduling, priority donation, mlfqs scheduling.
- User programs: Execution of multi-threaded user programs, system calls for accessing file system.
- Virtual memory: Demand paging, stack growth, swapping, memory-mapped files.
- File systems: Indexed and extensible files, sub-directories, caching file blocks.