Akash Banerjee

Last Updated on 30th August 2023

Contact

akashb.me

Kings Langley
United Kingdom

Mobile no:
+44 7471050392

Email:-
akash@akashb.me

Languages

English, Hindi, Bengali

Programming Languages

C
C++, JAVA
C#, JavaScript
Python
Flex/Bison
LLVM, MLIR
Git, GDB, LaTeX

About Me

Compiler Engineer at AMD, UK. Part of the HPC compiler's team for the Frontier supercomputing project where I work on the new llvm-flang driver for Fortran as well as device code offloading support for the OpenMP dialect in MLIR.

Previously I've worked at Imagination Technologies UK as a back-end graphics compiler engineer for their proprietary compiler.

My interests lie in Compilers, Program Analysis, Formal Verification and SAT Solvers. I've attained my master's in Computer Science & Engineering under the guidance of Prof. Saurabh Joshi in the field of Software Verification and applications of Formal Methods in Computer Security. During this time I have also worked under Prof. Ramakrishna Upadrasta in improving Compiler Optimizations through better profiling.

Qualifications

2022–Pres	**Sr Compiler Engineer** at AMD	Milton Keynes, UK
2021–2022	**Graphics Compiler Engineer** at Imagination Technologies	Kings Langley, UK
2018–2021	**M.Tech.** in Computer Science and Engineering - 9.50/10 CGPA	IIT Hyderabad
2013–2017	**B.Tech.** in Computer Science and Engineering - 8.37/10 CGPA	RERF, Kolkata

Interests

Compiler Optimizations
Using novel techniques and engineering principles for optimizing software systems.

Software Verification
Exploring techniques for formal verification of programs like Symbolic Execution, Abstract Interpretation, etc.

SAT Solvers
Studying and exploring techniques and encoding to make SAT/MaxSAT solvers more efficient

Skills

Programming Ability
Skilled in C, C++ and able to adapt quickly to new languages

Frameworks
LLVM Compiler Infrastructure, MLIR, CPRover Verification Framework

Tools
Git, LaTeX, GDB, LLDB, Eclipse

Projects

Sep. - 2022 **OpenMP & LLVM-Flang** Phabricator

Currently working on adding target offloading support to the OpenMP dialect in MLIR. Also, working on adding OpenMP support to the new MLIR based Fortran compiler llvm-flang. I have recently been awarded an **AMD Spotlight award** in recognition of this work. My LLVM Phabricator profile is available at reviews.llvm.org/p/TIFitis.

Jun. - 2021 **Proteus: Polymorphic Compilation**
Proteus is a compiler tool which uses polymorphic compilation and execution techniques to mitigate a class of side channel attacks with minimal performance overhead, compared to the other state-of-the-art solutions available. This work was done as part of my master's thesis project.

Apr. - 2020 **BPI Enhancements** GitHub Repo
Proposed and implemented improvements to the Branch Probability Information pass in LLVM to allow better static profiling leading to speed-up of up to 1.07x, as part of the course project for Advanced Compiler Optimizations - CS6240.
Accepted as a poster in **EuroLLVM-20** held at Paris, France.

Oct. - 2019 **Loop Acceleration** GitHub Repo
Added a loop acceleration module to the Pinaka verifier for quick detection of counterexamples in loops simulating polynomial functions. Pinaka is developed by IITH Software Verification Group which won the third-fastest verifier position in SV-COMP'20 Floats sub-category, amongst other positions and was the only entry from Indian academia.
Appreciated by the Dept. of CSE for this work here.

Sep. - 2019 **LLVM2GOTO** GitHub Repo
Created a tool to translate LLVM IR to CBMC-GOTO. LLVM supports multiple front-ends like C, C++, FORTRAN, Swift, etc., which get converted to LLVM-IR. CBMC is a tool to verify programs which has its own GOTO IR, this tool translates LLVM-IR to GOTO IR, allowing us to potentially verify all the languages that are supported by LLVM's front-end.

Mar. - 2019 **SAT Solvers** GitHub Repo
Implemented DPLL SAT Solver with MOMS heuristics, CDCL SAT Solver with Lazy data structure and Watch Literals, MaxSAT with Totalizer encoding and an Incomplete SAT Solver based on Break-only-poly algorithm and WalkSAT. As part of the course project for Constraint Programming - CS6483.

Nov. - 2018 **Hybrid Mutual Exclusion in Distributed Systems** GitHub Repo
An efficient implementation of a hybrid mutual exclusion algorithm for distributed systems by combining Raymond's and Maekawa's algorithms by multiplexing between them when communicating within clusters and across clusters, based on load, latency and throughput. As part of the course project for Distributed Computing - CS5320.

Co-Curricular

Jan. - 2020 **Teaching Assistant** IIT Hyderabad
Helped in grading and evaluating assignments for the CS6483-Constraint Programming course

Aug. - 2019 **Webpage Moderation** sat-smt.in
Maintainer for the Indian SAT+SMT School website :https://sat-smt.in

Jul. - 2019 **FMUpdate-India 2019** fmindia.cmi.ac.in
Organizing team member at the Formal Methods Update Meeting 2019

Jun. - 2019 **System Security** COEP Pune
Attended ACM India Summer School on Detection and Analysis of Malware

References

Dr. Saurabh Joshi - sbjoshi@cse.iith.ac.in

Dr. Ramakrishna Upadrasta - ramakrishna@cse.iith.ac.in