Chencheng Mao

+86 13805724402 | mcc0612@mail.ustc.edu.cn | GitHub | Anhui, China

Education

University of Science and Technology of China Anhui, China

B.Sc. in Computer Science; *Sep 2020– Jul 2024*

- **GPA**: 3.50
- **Relevant coursework:** Syllabus of Digital Logic Lab (94/100), Function of Complex Variable B (92/100), Data Structure (91/100), Principles and Techniques of Compiler (90/100), Foundations of Algorithms (89/100)

Research Experience

Optimize ThreadSanitizer in Link Time July 2023 – Oct 2023

Research Assistant, Instructor: Prof.Chenxiong Qian *The University of Hong Kong (HKU)*

- Developed effective run-time algorithms to detect data races with less false positives,less overheads and more precise synchronization analysis.
- Optimized ThreadSanitizer in LLVM by executing compiler instrumentation module in link time and eliminated unnecessary instrumentation in the whole-program optimization.
- Designed Thread Sanitizer in the context of ThinLTO and FullLTO seperately and evaluated the number of detected bugs and runtime overhead.

Performance Analysis and Characteization of Homomorphic Encryption Jan 2023 – Feb 2023

Research Internship,Instructor: Prof.Huiyang Zhou *Remote/NC State University*

- Designed packed multiplication algorithm based on SIMD operation to enable the multiplication of multiple numbers in a single operation.
- Implemented the code of open-source homomorphic encryption library HElib and performed experiments to evaluate the performance overheads.

Cminus-f Compiler Design and Optimization Sept 2022 – Dec 2022

Course Project, Instructor: Prof.Cheng Li *USTC*

- Implemented the rudimentary Cminus-f compiler to automatically generate LLVM IR for the Cminus-f language.
- Adopted Global Value Numbering for optimization of redundancy elimination.
- Achieved a performance improvement of up to 20% compared to the baseline program.

Projects

Cminus-f Compiler [Link]

- Implemented various optimization methods, including SSA, Global Value Numbering, and dead code elimination.

A Rudimentary Operating System [Link]

- Utilized the QEMU bare-metal environment to create a functional operating system.
- Functions include: (1) VGA & serial output; (2) clock interrupt; (3) shell; (4) memory management and allocation; (5) process scheduling functionality

Skills

Programming: C, C++, Python, MySQL, C#, TeX

Technologies: Git, Docker, Vscode, Vivado

Frameworks: LLVM

Languages: Bilingual in English(TOFEL:105), Chinese