Ziqi (Astra) Zhao

979-319-2916 | astrajoan@tamu.edu | linkedin.com/in/astrajoan | github.com/astrajoan | astrajoan.github.io

Education

Texas A&M University — College Station, TX
Ph.D. in Computer Science and Engineering. GPA: 4.0/4.0 — Dec 2024

Beihang University — Beijing, China
B.S. in Electrical and Computer Engineering. GPA: 3.8/4.0 — June 2018

Experience

The Linux Foundation — Fremont, CA
Mentee. Program: Linux Kernel Bug Fixing Summer 2023 — May 2023 – Aug 2023

- Contributed a total of 6 patches to the mainline Linux kernel as a learning experience for open-source kernel development
- Fixed bugs reported by Syzkaller on various kernel subsystems, including networking, GPU driver, filesystem, and `kselftest`
- Diagnosed a reference leak in Linux bridge devices by using GDB to debug the `vmlinux` binary file, and refactored corresponding code to simplify net device lifecycle management and avoid creating error-prone references
- Resolved a deadlock in Linux CAN device driver by enforcing strict topological order on 3 nested spinlocks, based on the `dmesg` output obtained with `CONFIG_LOCKDEP` enabled and the syscall history recorded by the `strace` command
- Created a blog to share my suggestions on working with Syzbot bugs: https://astrajoan.github.io/2023/08/21/syzbot.html

Texas A&M University — College Station, TX
Graduate Research Assistant. Supervisor: Dr. Vivek Sarin — May 2022 – Present

- Employed numerical algorithms to optimize Gaussian Process (GP), a supervised ML model with inherent uncertainty measures, greatly improving its computation complexity and achieving 2× faster convergence speed compared to existing methods
- Developed and optimized our models in C++20 for both CPU-based computation with Eigen and OpenBLAS, and GPU-based computation with CUDA, cuBLAS, and cuSPARSE libraries
- Sponsored by the Linux Foundation to publish this article: https://thenewstack.io/using-gpytorch-a-researchers-experience/

Publications

Interpretation of Time Series Deep Models: A Survey — Fremont, CA
Z. Zhao, Y. Shi (co-first author), S. Wu (co-first author), F. Yang, W. Song, N. Liu — June 2022 – Present

- Reviewed widely used post-hoc interpretation methods and inherently interpretable models on time-series deep learning

Projects

Oathkeeper: Fault-Tolerant Distributed System | C++20, Boost, gRPC, gtest, AWS, CMake — July 2023 – Present

- Implemented the Raft consensus protocol from scratch, incorporating leader election, log replication, and persistent state features
- Developed a MapReduce system that enables users to supply custom Map and Reduce tasks to run on separated machines, using TCP sockets as RPC endpoints and Amazon S3 for shared storage of intermediate states
- Employed an event-driven service architecture with 100% asynchronous I/O operations, based on Boost.Asio, C++20 coroutines, and gRPC with its `CompletionQueue` API for non-blocking request processing
- Orchestrated compilation with CMake and devised 50+ unit-tests with gtest to ensure 100% consistency in concurrent execution

pastecat.io: Code Snippet Sharing Tool | React, Node.js, Docker, GCP, OAuth 2.0 — June 2023 – Aug 2023

- Built a React-based website for saving code snippets, with an accompanying Node.js CLI tool for UNIX shell-based workflows
- Constructed the backend with two coordinative components: a Firebase Firestore NoSQL database for high-availability querying based on paste IDs, and a Firebase Cloud Storage service for storing actual paste files
- Deployed the website with Docker containers and served traffic under a GCP external application load balancer
- Implemented Firebase Security Rules and user authentication based on OAuth 2.0 to protect access to the backend services

GPU-Based Strassen Algorithm | CUDA, cuBLAS, C++ — Feb 2022 – May 2022

- Implemented the Strassen algorithm for matrix multiplication that reduces its time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{\log_2 7} \approx n^{2.8})$
- Provided a GPU version of the algorithm using CUDA and cuBLAS and leveraged NVIDIA Nsight profiling tools to measure the impact of optimizations such as using CUDA streams and consolidating temporary memory usage
- Obtained a 1988× speedup compared to CPU-based Strassen when multiplying $2^{12} \times 2^{12}$ matrices on an NVIDIA RTX 3090
- Studied the implication of recursive approaches for GPU matrix multiplication by comparing the Strassen routine with a custom CUDA kernel optimized with shared memory, vectorization, and matrix block-tiling

Skills

Languages: C++, C, Python, Bash, JavaScript, SQL, Go
Frameworks: CUDA, OpenMP, PyTorch, Tensorflow, React, Node.js, RPC, REST API, Git, Docker, CMake, Make, AWS, GCP
Familiar with: Linux kernel, parallel computing, operating system, distributed system, asynchronous I/O, networking