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 issue by enforcing strict topological order on 3 nested spinlocks, based on the kernel `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

- Leveraged PyTorch to optimize the computation of Gaussian Process models, a type of probabilistic machine learning algorithm, and enhanced their prediction accuracy and efficiency
- Summarized high-level concepts in this blog: https://astrajoan.github.io/2023/08/26/research.html, and got sponsored by the Linux Foundation to publish this article: https://thenewstack.io/using-gpytorch-a-researchers-experience/

Projects

Oathkeeper: Fault-Tolerant Distributed System | C++20, Boost.Asio, 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 state object storage
- Employed an event-driven service architecture with fully 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

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

- Implemented the Strassen matrix multiplication algorithm that recursively divides square matrices into smaller sub-matrices, and computes additions instead of multiplications to reduce the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{\log_2 7} \approx n^{2.8})$
- Provided a GPU-based version of the algorithm that leverages CUDA and cuBLAS to parallelize workload onto GPU threads, as well as a CPU-based serial version in C++ as the baseline for performance comparison
- Optimized computation speed by 140× for $2^{10} \times 2^{10}$ matrices on an NVIDIA RTX 6000 GPU compared to the serial version

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

- Built a web application for sharing and exporting code snippets with React as the frontend, and included a CLI tool written in Node.js for a more developer-friendly experience working with pastes in a UNIX shell environment
- 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

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 state-of-the-art post-hoc interpretation methods and inherently interpretable models on time-series deep learning

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