Long Tran

long.h.tran1904@gmail.com | https://www.linkedin.com/in/longtran1904/ | https://github.com/longtran1904
https://people.cs.rutgers.edu/~lht21/

Mission Statement

My research focuses on enabling timely and resource-efficient execution of traditional and AI workloads on constrained computing platforms. As edge systems become increasingly incorporating diverse accelerators, over-subscribing applications, and agentic AI workflows, existing OS runtimes fall short of managing contention and security. To address this, I design runtime OS mechanisms and software–hardware co-designs that improve performance and power efficiency under tight resource constraints.

Education

Rutgers University — New Brunswick, NJ
Ph.D in Computer Science — September 2024 - Present
Advisor **Prof. Sudarsun Kannan**

Rutgers University — New Brunswick, NJ
B.S in Computer Science — Dec 2023

Publications

[1] Long Tran, River Bartz, Ramakrishnan Durairajan, Ulrich Kremer, and Sudarsun Kannan. 2025. ***Are Edge MicroDCs Equipped to Tackle Memory Contention?*** In Proceedings of the 17th ACM Workshop on Hot Topics in Storage and File Systems (HotStorage '25). Association for Computing Machinery, New York, NY, USA, 38–44. https://doi.org/10.1145/3736548.3737827

[2] Long Tran, River Bartz, Uli Kremer, Ramakrishnan Durairajan and Sudarsun Kannan. 2025. ***Stateful Triage for Reliable and Secure Wildfire Monitoring at the Edge***. In Proceedings of 5th International Workshop on the Internet of Things for Adversarial Environments co-located with IEEE MILCOM, Los Angeles, CA.

Research Experience

Resource-Efficient Computer Vision/AI Inference Pipeline

- Developing a heterogeneous video decoder that combines hardware decoders and software decoder throughput for low-latency video decoding, as main input for AI inference pipeline.
- Fine-tuning Computer Vision Pipeline performance for resource constraints: memory, GPU memory, energy, computational costs.
- Researching methods to reduce latency to decode and inference on a single 4k video using block-based (H.264/H.265) codecs.
- Speeding up 32.4% for the end-to-end video to inference process by offloading Color Conversion and YOLO preprocessing to GPU using NVIDIA NPP, and Pytorch.

Secured AI Applications on Multi-tenant Edge System

- Accelerated YOLO inference speed by $\geq$ 2x by tuning batch size and multiprocessing.
- Analyzed performance degradation of YOLO and RocksDB in Edge Systems.
- Detected security vulnerabilities (OOM-killed, side-channel attacks) for edge systems empowered with GPU.
- Work presented at HotStorage 17th ACM Workshop on Hot Topics in Storage and File System (HotStorage'25)**[Paper]**
- Work presented at 5th International Workshop on the Internet of Things for Adversarial Environments (co-located with IEEE MILCOM)[**Paper**]

YOLO Training Prefetching

- Developed prefetch mechanism for YOLO (You Only Look Once) Training.
- Reduced memory usage by 67% while remaining high speed data loading, preventing 23% performance degradation due to memory contention under multitenant scenarios.
- Resolved a potential memory leak - **Contribution Patch**

Work Experience

REHAB Lab | Rutgers University, NJ June 2022 - September 2022

- Renovated REHAB's research lab database from MS Excel to MS Access software.
- Designed interactive UI/UX for querying 1500+ user personal info using MySQL queries.
- Increased 65% database performance by migrating 2000+ data rows to Microsoft Access Database.
- Allowed 5+ concurrent users by launching a centralized database replication.

SiGlazVN | Ho Chi Minh City, Vietnam March 2021 - May 2021

- Redesigned mobile home pages using C#, reduced homepage load time by 72% implementing lazy loading.
- Delivered mobile inbox notification feature on iOS, Android app using cross-platform framework in C#, increased promotion campaigns' reachability towards in-app users by 66%.
- Implemented 400+ unit test cases on .NET RESTful API, decreased in-deploy errors by 58.3%.

Academic Projects

Online Gomoku Server with AI Bot | *C/C++*

- Designed multi-threaded server with custom-written communication protocol for Tic-tac-toe game in C
- Allowed 5+ concurrent games and handle player's interruptions for surrender, sudden program exit.
- Launched an AI bot with 70% win-rate over human using a min-max tree algorithm to play on 20x20 version of tic-tac-toe, boosted algorithm execution time by 89% using alpha-beta pruning and bit-hashing approximation heuristics.

User-space Virtual Memory & Virtual File System Library | *C*

- Developed user-friendly API supports efficient memory management using virtual memory techniques.
- Empowered user launch their own file directory on one Linux file by implementing FUSE virtual file system.

Movie Recommendation System & Remote Key-value store Interface | *Java Spark*

- Designed a remote key-value storage service on application layer using TCP protocol.
- Implemented a recommendation service that can analyze over 100m rows of movie data using Java Spark

Technical Skills

Languages: C/C++, Python, CUDA, Bash
Frameworks: Pytorch, Tensorflow, Docker, Kubernetes, Grafana
Developer Tools: Git, Linux, Bash, Vim, AWS, CI/CD