Qizhong Wang

408-329-2578 qwang148@ucsc.edu

Education

University of California, Santa Cruz **Sep. 2024 – Present**
Ph.D. Student in Computer Science *Santa Cruz, CA*
Advisor: Prof. Yuanchao Xu

Huazhong University of Science and Technology **Sep. 2020 – Jun. 2024**
B.E. in Artificial Intelligence *Wuhan, China*

Publications

Qizhong Wang, Xiangyue Huang, Yanan Guo, Yuanchao Xu, "Security and Performance Implications from GPU Cache Eviction Operators", In Submission, 2024.

Project Experience

Security and Performance Implications from GPU Cache Eviction Operators **July. 2024 – Nov. 2024**
Advisor: Professor Yuanchao Xu and Yanan Guo

- Designed experiments to analyze NVIDIA GPU cache eviction operators, such as evict_first and evict_last, analyzing their properties.
- Developed novel covert channel attacks (First+First and Last+Normal) utilizing eviction properties, achieving higher bandwidth than prior works.
- Leveraged cache eviction operators to optimize cache utilization, significantly improving performance on PageRank and RNN benchmarks.

Research Intern at UT Arlington **Aug. 2023 – Nov. 2023**
Advisor: Professor Hong Jiang

- Installed OpenWhisk on a single node and cluster on the UTA ACES lab server (CentOS8) and be able to run common benchmarks.
- Took the serverless work of MXFaaS, AQUATOPE, FaasCache, etc. as the baseline, which will be reproduced on our servers.

Reproducing FusionRAID (FAST'21) **Jan. 2023 – July 2023**
Advisor: Assistant Professor Zhang Jie

- Modified the kernel of linux-5.11 and add the replicated writing function to the md module so that small-scale data is directly copied and written without raid stripes, and two copies of data are stored on different SSDs.
- Added the conversion function to convert replicated writing blocks into raid strips when appropriate

Stateful Serverless Data Analytics Workloads **Sep. 2022 – Jan. 2023**
Advisor: Dr. Yue Cheng

- Aim to apply the new SSD to serverless computing, and the ZNS SSD is suitable for storing intermediate data in multi-tenant scenarios, where the user can specify data to be written to a zone in the ZNS, separated different jobs from each other, avoiding wear and additional capacity overhead caused by garbage collection.
- Specifically, generated randomly different integers as the size of the data and the workloads according to the given write and read operations in order, applied libzbd and blkzone to simulate ZNS, read the workloads using multiple threads, generated some floating points for I/O operations, saved the offset to a table during write operations to get the position of the data and tested I/O rates, Latency.

Technical Skills

Languages: C, C++, Python, CUDA
Systems: Linux