LEI ZHANG

601 108th Ave NE, Bellevue, WA 98004
(+1) 404-662-8268 | geraldleizhang@gmail.com | https://geraldleizhang.com

Research Interest

My research interests are broadly in distributed systems performance and reliability.

- Reliability, observability, and performance analysis for large-scale cloud systems and microservice systems
- Performance quantification and analysis, memory management for distributed cache/memory systems
- Performance analysis and reliability for LLM training on RDMA-based AI infrastructures in large-scale data centers

WORK EXPERIENCE

Bytedance Inc. 09/2023 – Present
Research Scientist
End-to-end observability, performance analysis and root-cause analysis for optimization and reliability of LLM training on RDMA-based AI infrastructures, across Megatron, NCCL, GPU, and RoCE/InfiniBand networks.

Facebook Inc. 05/2018 – 08/2018
Ph.D. Software Engineer Intern
Worked in the video cache infrastructure team. Developed a fine-grained storage utilization monitor. Developed an automatic data management mechanism to ensure efficient resource utilization and fairness.

EDUCATION

Princeton University 01/2022 – 08/2023
Postdoc Research Associate
- Advisor: Prof. Ravi Netravali

Leveraging programming language methods (e.g. static analysis) to achieve automatic instrumentation for distributed tracing, and root-cause analysis on distributed systems. Development and performance analysis on volumetric video streaming systems.

Emory University 08/2018 – 12/2021
Ph.D. in Computer Science
- Advisor: Prof. Ymir Vigfusson
- Transferred as a post-qualified Ph.D. candidate

Georgia Institute of Technology 08/2015 – 08/2018
M.S. in Computer Science
- Advisor: Prof. Karsten Schwan (deceased)
- Was in the Ph.D. program

Tsinghua University 08/2011 – 07/2015
B.E. in Computer Science

PUBLICATION

Faulty Machine Detection for Large-scale Distributed Machine Learning Under Review

Towards Bandwidth-adaptive Live Volumetric Video Streaming Under Review

Provenance-guided Automatic Online Debugging Under Review

Operational Telemetry System Measuring Fine-grained Status for Large-scale RDMA Networks Under Review

LatenSeer: Casual Modeling of End-to-End Latency Distribution by Harnessing Distributed Tracing ACM SoCC'23
Yazhuo Zhang, Rebecca Isaacs, Yao Yue, Juncheng Yang, **Lei Zhang**, Ymir Vigfusson

The Benefit of Hindsight: Tracing Edge-Cases in Distributed Systems USENIX NSDI'23
Lei Zhang, Zhiqiang Xie, Vaastav Anand, Ymir Vigfusson, Jonathan Mace

When is the Cache Warm? Manufacturing a Rule of Thumb USENIX HotCloud'20
Lei Zhang, Juncheng Yang, Anna Blasiak, Mike McCall, Ymir Vigfusson

Optimal Data Placement for Heterogeneous Cache, Memory, and Storage Systems ACM SIGMETRICS'20
Lei Zhang, Reza Karimi, Irfan Ahmad, Ymir Vigfusson
Kenneth C. Sevcik Outstanding Student Paper Award

Deceptive Secret Sharing IEEE DSN'18
Lei Zhang, Douglas M. Blough

Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements IEEE ToC'16
Maomeng Su, **Lei Zhang**, Yongwei Wu, Kang Chen, Keqin Li

RESEARCH PROJECTS

Performance and Reliability in AI Infrastructure

Observability and Reliability in Collective Communication Libraries over RDMA-based Datacenter for LLM Training

Understanding cross-layer performance with better observability for large language model training is critical to not only providing better performance feedback for training framework optimization, but also enhancing reliability when a problem occurs. Collective communication libraries (CCL) connect training framework and RDMA network architectures, while unfortunately todays' CCL runs like a black box without efficient observability methods. In this project, we provided a practical instrumentation to CCL, aiming to provide better performance breakdown of CCL's inter mechanisms to really enable end-to-end observability from training algorithms to IB verbs in RDMA layer, together with lower level network measurements. With such observability, we identify potential performance bottlenecks within asynchronous queues, communication and computation overlaps, etc., to figure out potential performance improvements and also quickly respond to silent failures in large-scale training jobs.

Faulty Machine Detection for Large-scale Distributed Machine Learning

Large-scale distributed model training requires simultaneous training on up to thousands of machines, while faulty machine detection is critical when an unexpected fault occurs in a machine. From our experience, a training task can encounter two faults per day on average, possibly leading to a halt for hours. To address the drawbacks of the time-consuming and labor-intensive manual scrutiny, we propose an automatic faulty machine detector for distributed training tasks. The key idea is to automatically and efficiently detect faulty distinctive monitoring metric patterns, which could last for a period before the entire training task comes to a halt. The system has been deployed in a production environment for over nine months, monitoring daily distributed training tasks where each involves up to thousands of machines. In our real-world fault detection scenarios, it can accurately and efficiently react to faults within 3.6 seconds on average, with a precision of 0.904 and F1-score of 0.893.

Observability and Reliability in Distributed Systems

Tracing Rare Bugs in Distributed Systems with Low Overhead

Distributed tracing frameworks suffer from high overhead in production environments. We built Hindsight (in C and Go), a lightweight and always-on distributed tracing system to achieve fine-grained tracing to track **rare system bugs** with **low overhead**. Hindsight can track **100%** of requests at scale with nanosecond-level tracing and **improved 2x throughput** to state-of-the-art tracing framework (Jaeger) which only tracks 1% of requests. We also built an Apache Thrift gRPC microservice benchmark (in C++) to evaluate scalability for tracing frameworks.

Improving Fine-grained Observability for Dependency Bugs

Tracking cross-context data dependency is crucial for debugging real-world incidents in distributed systems. We proposed to leverage **static analysis** to preprocess distributed systems and achieved **automatic instrumentation**. We developed a system (in Java and Python) to automatically expose the related dependency, and expose those critical and responsible trace data for developers. We proved this method can practically **detect and uncover latent code bugs**, and significantly improve the debugging time in production systems.

Distributed Caching and Data Placement

Optimal Data Placement for Heterogeneous Memory Systems

Traditional data placement principles are no longer aligned with the complexities of today's memory hierarchy. We developed CHOPT (in C++), a data-driven optimal placement algorithm, to conduct optimal placement solutions in

multi-tier cache and memory systems. CHOPT applied spatial sampling, which achieved **0.2% error with 1% sampling rate.** CHOPT **improved up to 44.8% request latency** across more than 30 production traces, compared with state-of-the-art optimal algorithms. We also provided heuristics from **statistical analysis** for online cache algorithm design.

Understanding Distributed Cache Warmup
Understanding cache warmup process is challenging due to its dynamic, but crucial for production performance. We derived a **regression-based** rule-of-thumb formula to accurately estimate cache warmup time. Our estimation formula achieved **78.3% - 100% accuracy on over 170** storage and CDN cache workloads.

Fine-grained Storage Utilization Management for Facebook's Video Cache
Making video cache migration and eviction decisions is crucial. I developed a utilization monitor (in C and C++) to track downstream storage usage. I proposed to achieve approximation on aggregated trace data to tolerate data lags and achieve efficiency. I also deployed an automatic mechanism to manage video content at fine granularity.

Formalized Data Placement Optimization in Multi-cloud Storage
Optimizing placement strategies on cloud storage is crucial when considering ad-hoc user configurations. We developed Triones (in C and Python), a systematic model to utilize erasure coding and provide optimal data placement configurations. Triones can achieve **50% access latency reduction** and **improve 2x fault tolerance** with reasonable cost, compared to the state-of-the-art models.

Others

Live Volumetric Video Streaming
Live volumetric (3D) video streaming is challenging to achieve low latency with extremely high data volume and computational complexity. We proposed a novel design to transmit adaptively selected 2D stream data and render at the client side to overcome network bandwidth limitations. We developed a bandwidth-adaptive video streaming system (in C++ and Python) on top of **WebRTC**. We proved our system can support **high throughput, low end-to-end latency, and compatibility** with lightweight computing resources.

Data-driven Malware Detection
I developed a systematic data-driven approach to attribute malware behaviors to specific actors, tools, and intents. I built a system to **parse and clean large-scale data** from a real malware execution dataset. I designed an LSM-Tree-like data structure and identified correlations between system and network logs. I also designed a clustering algorithm to identify malware families.

Deceptive Secret Sharing
Confidentiality of data storage is critical for many cryptographic scenarios. We proposed a novel approach to combine confidentiality and deception with secret sharing for sensitive data storage. We developed a distributed storage system (in C++) to achieve deceptive secret sharing. Our system can provide high confidentiality and data availability with reasonable overhead.

AWARDS

Kenneth C. Sevcik Outstanding Student Paper Award — SIGMETRICS'20
Bronze Medal — 24th, 25th China Mathematical Olympiad

PROFESSIONAL SERVICES

Program Committee — ACM SoCC'22, SoCC'23
Invited Program Committee — ACM SIGMETRICS'23
External Reviewer — USENIX ATC'21, HotCloud'20, HotStorage'20
External Reviewer — ACM Eurosys'19, SoCC'18, SRDS'18

MENTORSHIP

Rajrup Ghosh (Ph.D., USC), Jingyuan Chen (Ph.D., Princeton), Yazhuo Zhang (Ph.D., Emory), Tao Zhou (Undergrad, Emory)