Peiran Qin

Pre-Doctoral M.S. Student
University of Chicago
5035 East End, Chicago, IL 60615

Email: peiranqin@uchicago.edu
Phone: +1 (773) 812 5328
https://qinpr.github.io/

I'm activately looking for **2025 Fall Ph.D. Opportunities**!

Research Interests

Areas: Operating and Storage Sytems and Distributed Sytems and their intersections with Machine Learning

Focus: Building more performant systems in the contex of:

1) High Efficiency (e.g. prefetch and cache eviction co-design for higher cache efficiency)
2) Low Latency (e.g. mitigating tail latencies in cloud storage systems)
3) Workload Awared (e.g. workload characterization and prediction for distributed machine learning clusters)

Education

University of Chicago

Pre-Doctoral M.S. in Computer Science; GPA: 3.81/4.00 — Sep 2023 - Present

- Advisor: Haryadi S. Gunawi
- Merit-based scholarship recipient (4 / CS class of 2024)

Chinese University of Hong Kong, Shenzhen

B.E. in Computer Engineering; First Class Honours — Sep 2019 - July 2023

- Advisor: Yeh-Ching Chung

Publications

In Submission '24	Anonymous Author(s). **Heimdall: Accurate and Efficient I/O Admission Policy with Extensive Machine Learning Pipeline**
Near Submission '24	Anonymous Author(s). **Storage Prefetching Needs Feedback: An Efficient Prefetching/Eviction Co-Design with Small Cache Footprints** [Temporary Title]

Research Experience

University of Chicago; Advisor: Haryadi S. Gunawi

Research Assistant, UCARE Group — April, 2023 - Present

Cache Prefetching & Eviction Co-Design

- Discovered the cache prefetching and eviction co-designed policy. Compared to state-of-the-art algorithms, the proposed method achieves the best cache efficiency, at the same time incurs 62% less memory footprint
- Implemented the system on multiple platforms including python simulatior, linux kernel, Ceph.
- Implemented and characterized state-of-the-art prefetchers (e.g. Pythia) and eviction algorithms (e.g. SIEVE, S3FIFO).
- One of the major members of structuring, organizing, and conducting the evaluation experiments.

(Paper near submission '24)

I/O Admission Policy

- Explored the extensive ML pipeline to build an efficient I/O admission policy. Compared to state-of-the-art algorithms, our system has 15%-35% lower I/O latency than state-of-the-art, and 2x faster than baseline.
- Hacked the deployment of our system on linux kernel level and C client level.
- Implemented the joint-inference which maintain the scalability of our systems under 8x heavier workload.
- Implemented the SSD wear-leveling emulator which is cheap, scalable and extensible, based on FEMU.

(Paper in submission '24)

Microsoft Research, Asia; Mentor: Lei Qu

Research Intern, Networking Research Group — May, 2022 - Dec, 2022

Workload Characterization and Prediction for ML Clusters Design

- One of the major contributors to the AISim project, supporting ML distributed systems design by workloads analysis.
- Completed the design, implementation, and validation to extract the general workload characteristic statistics from first-part clusters and proposed a pipeline to predict the future workload's distribution. The outcome has a superior goodness-of-fit (p-value=0.3) and generalizability.

(Research report delivered to Microsoft Engineering Group '22)

Workload Profiling in ML Clusters

- One of the major contributors to DMLProfiling project, focusing on ML distributed systems profiling.
- Analyzed the model composition of jobs in clusters and resource metrics of different models.
- Devised a high-precision solution for inferring the job types and model types (ResNet, Bert, ViT, etc) of jobs run on clusters by analyzing the utilization of key resources, with the accuracy reaching over 99%.

(Research report delivered to Microsoft Engineering Group ’22)

Alibaba Group; Advisor: Yeh-Ching Chung

Student Research Assistant Dec, 2021 - May, 2022

Graph Computing Sytems Optimization

- Worked on the large-scale sub-graph matching systems. Focused on reducing memory footprint.
- Optimized the design by replacing the stale tabular-based catalog with graph-based representation equipped with dynamic programming, which reduces the memory usage and improves the efficiency.
- Proposed a solution for encoding and decoding sub-graphs in graph computing with reduced memory footprint.

(Paper accepted at ATC ’23)

Other Experience

Repeto Program, Summer of Reproducibility Fellowship. *funded from NSF*; Contributor Jun, 2024 - Present

Zhongshen Agricultural Innovation Technology Co., Ltd; Software Development Engineer May, 2021 - Dec, 2021

Technical Skills

Programming Language:	C, C++, Python, Rust, Java, C#, Verilog HDL, SQL
Linux Kernel:	Experience of block layer, vfs, and syscalls hacking in Linux-6.0.0
Distributed Systems:	RocksDB, Ceph
ML Framework:	PyTorch, TensorFlow
Containerization:	Docker
Parallel Computing Libraries:	CUDA, MPI, Pthread, OpenMP, NCCL
Other	LaTeX

References

Haryadi S. Gunawi
Associated Professor
University of Chicago
haryadi@cs.uchicago.edu

Yongqiang Xiong
Principal Researcher
Microsoft Research, Asia
yqx@microsoft.com

Yeh-Ching Chung
Professor
Chinese University of Hong Kong, Shenzhen
ychung@cuhk.edu.cn