

Lorri (Gengyu) Rao



2134001242



gengyura@usc.edu



Pronouns: They/Them or Lorri (Name as pronoun)



<https://z-y00.github.io/academic/>

Education

2020 Aug. – 2024 Aug.



Ph.D. Computer Engineering

University of Southern California (USC).

Thesis: Algorithm and system co-optimization of graph and machine learning systems

GPA: 3.89

2016 Aug. – 2020 Aug.



B.S. Computer Science.

Huazhong University of Science and Technology (HUST)

Thesis: Design and Implementation of GPU Memory Management Framework for Deep Learning System (This is a followup of Capuchin: Tensor-based GPU Memory Management for Deep Learning, ASPLOS '20)

GPA: 3.85

Skills

Languages



English, Mandarin Chinese.

Coding



CPP, C, Python, ...

Misc.



Cross-Stack optimization for graph and machine-learning, from hardware architecture (Hardware-Software Co-design) to algorithm level. (Details in the Projects section)

Many of my projects used HPC clusters and are related to High Performance Computing, which involve profiling, CUDA, MPI, SIMD optimization, etc.



I mentored lab interns and junior PhD students. It was my pleasure to know some of them in person, build deep connections, and learn from them. I also did research projects collaborating with others and by myself.

Research Publications

1

Rao, G., Chen, J., Yik, J., & Qian, X. (2022). Sparsecore: Stream isa and processor specialization for sparse computation. In *Proceedings of the 27th acm international conference on architectural support for programming languages and operating systems (ASPLOS '22)* (pp. 186–199).

doi:10.1145/3503222.3507705

2

Zhuo, Y., Chen, J., **Rao, G.**, Luo, Q., Wang, Y., Yang, H., ... Qian, X. (2021). Distributed graph processing system and processing-in-memory architecture with precise loop-carried dependency guarantee. *ACM Transactions on Computer Systems*, 37(1–4). doi:10.1145/3453681

Teaching and curriculum development

2024 Summer



Introduction to Machine Learning & Graph Neural Network Accelerators (USC)

This is for updating USC hardware course contents. I designed curriculum about Systolic Array and different GEMM dataflow, Overview of a Systolic-Array-based CPU ML Coprocessor (Gemmini: <https://chipyard.readthedocs.io/en/1.4.0/Generators/Gemmini.html>), and Overview of recent arch designs for sparse computation, i.e. Extensor, Sparse Tensor Core. For graph accelerator, we didn't go into details. I added the basics of splitting computation into Graph and Neural Network, and special designs for graph data access.




2022 Fall



Introduction to Computer Networks (USC EE 450)

Teaching, Grading & Curriculum development.




Teaching and curriculum development (continued)

- 2021 Fall  **Parallel and Distributed Computation (USC EE 451)** Curriculum development
- 2020 Fall  **Advanced Topics in Microarchitecture (USC EE 653)** Curriculum development
- 2018 Fall  **Parallel and Sequential Data Structures and Algorithms (HUST)**
Teaching, Grading & Curriculum development. Based on CMU 15-210






Projects

- PhD Current – 2022  **Machine Learning optimization**
LLM inference optimization
This is part of my PhD thesis. I optimized the large language model inference system for better trade off between accuracy and performance and better user experience (SLO).
My work includes: Inference latency prediction; Optimization of scheduling algorithm, neural network operators, and memory management system on GPU.
Cache-aware DNN inference on CPU
In recent years, high-end CPU usually has private L2 cache with size of several MB. We partitioned DNN models and implemented the runtime scheduler support to best utilize the cache capacity.
My work includes: Modify the GEMM kernel library; Adding schedule support in the runtime system.
- 2022  **Graph Pattern Mining and Sparse Computation / Stream ISA**
This work is a Hardware-Software co-design for CPU. We extend the instruction set architecture (ISA) to make stream or sparse vector first-class citizens, and develop efficient architectural components to support the stream ISA. We used this architecture to accelerate Graph Pattern Mining and Sparse Computation (GEMM).
My work includes: Architecture design; Performance and Function simulation; Adding new instructions and intrinsic to compiler stack; Using the new instructions to accelerate applications.
Publish as *SparseCore: stream ISA and processor specialization for sparse computation* in ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '22).
- 2021  **Distributed Graph Computation**
Our project is based on the Processing In Memory architecture and traditional distributed system.
I designed and implemented a new communication and computation flow, which can significantly reduce or eliminate the redundant communication and computation. My work is a Hardware-Software co-design on the PIM architecture
Published as *Distributed graph processing system and processing-in-memory architecture with precise loop-carried dependency guarantee* on ACM Transactions on Computer Systems (ToCS).
- Undergrad – 2020  **GPU Memory Management Extension for Tensorflow**
This is my undergrad thesis, which is a follow up of *Capuchin: Tensor-based GPU Memory Management for Deep Learning*. (ASPLOS '20). The original work implemented static analysis of tensor graph for memory swapping.
I added dynamic runtime analysis and better support for tensor algebra optimization to the GPU memory management system.




Projects (continued)

- 2019  **Random walk optimization on GPU**
This work was done when I was a research intern at University of Southern California. I implemented optimizations in CUDA kernel, threading policy and graph storing format. Compared with SoTA system, Gunrock at that time, we reduced the runtime by 50% ~ 90%.
-  **Community Earth System Model**
This application is used to reproduce global climate change processes. We optimized this application as part of the Student Supercomputer Challenge (ASC19). We used CUDA in some hotspots, and replace the FORTRAN codes with Intel MKL library. I vectorized the codes with the help of Intel compiler and made several cache optimizations. I also added optimization based on numerical analysis.
- 2018  **RDMA enabled Hbase**
We added RDMA communication for Apache Hbase, which reduced the request latency and increased the throughput. This was part of the student RDMA Programming Competition. We got 2nd place prize for the competition.




Teaching and curriculum development

- 2024 Summer  **Introduction to Machine Learning & Graph Neural Network Accelerators (USC)**
This is for updating USC hardware course contents. I designed curriculum about Systolic Array and different GEMM dataflow, Overview of a Systolic-Array-based CPU ML Coprocessor (Gemmini: <https://chipyard.readthedocs.io/en/1.4.0/Generators/Gemmini.html>), and Overview of recent arch designs for sparse computation, i.e. Extensor, Sparse Tensor Core. For graph accelerator, we didn't go into details. I added the basics of splitting computation into Graph and Neural Network, and special designs for graph data access.
- 2022 Fall  **Introduction to Computer Networks (USC EE 450)**
Teaching, Grading & Curriculum development.
- 2021 Fall  **Parallel and Distributed Computation (USC EE 451)** Curriculum development
- 2020 Fall  **Advanced Topics in Microarchitecture (USC EE 653)** Curriculum development
- 2018 Fall  **Parallel and Sequential Data Structures and Algorithms (HUST)**
Teaching, Grading & Curriculum development. Based on CMU 15-210



Awards and Achievements

- 2020 – 2021  **PhD student Fellowship at USC Viterbi School of Engineering**
- 2019  **Student Supercomputer Challenge (ASC19)**
In this contest, we designed, built, tuned, and optimized our HPC systems, ran benchmarks and real scientific applications with our code optimization in limit time. Our team finally ranked 8th among more than 300 teams.
<http://www.asc-events.org/StudentChallenge/History/2019/index.html>
- 2018  **Student RDMA Programming Competition 2018**
We added RDMA communication for Apache Hbase. This was done by me and my friend at our sophomore year. We got Second Prize.
<http://hpcadvisorycouncil.com/events/2018/rdma/>

Open source community

- Undergrad  **Debian Security Tools Packaging Team**
https://contributors.debian.org/contributor/anon_yoo-guest@alioth/
-  **Debian Chinese Community** (Package maintaining)
<https://github.com/debiancn>
-  **Wuhan Linux user group, WHLUG**

Social activity

- 2022 – 2023  **Volunteer at LGBTQ+ Student center**
<https://lgbtqplus.usc.edu/>
- 2016 – 2017  **Volunteer at Hubei Provincial Museum**
link to Google Art and Culture

Relevant USC graduate courses

Course	Grade	Course Description
CSCI-402	A-	Operating Systems
EE-542	A	Internet and Cloud Computing
EE-557	A-	Computer Systems Architecture
CSCI-570	A	Analysis of Algorithms
EE-503	A	Probability for Electrical and Computer Engineers
EE-450	A-	Introduction to Computer Networks
CSCI-585	A	Database Systems
CSCI-567	A	Machine Learning
CSCI-551	A	Computer Networking