Jingji Chen

Personal Data

Phone:	+1 (323) 447-1406
email:	jingji.chen.000@gmail.com(preferred)/chen3385@purdue.edu
Homepage:	https://amadeuschan.github.io
Google Scholar:	https://scholar.google.com/citations?user=KviNHgEAAAAJ&hl=en
Github:	https://github.com/AmadeusChan

Education

Starting from 2022 Aug.	Ph.D. in Computer Science (transferred from USC, passed the preliminary exam) Advisor: Prof. Xuehai Qian Department of Computer Science, **Purdue University**, West Lafayette, USA Earliest Graduation Date: Summer 2024
2019 Aug.-2022 Aug.	Ph.D. in Computer Engineering (passed the screening exam, GPA: 3.96/4.0) Advisor: Prof. Xuehai Qian Ming Hsieh Department of Electrical and Computer Engineering, **University of Southern California**, Los Angeles, USA
2015 Sept.-2019 July	Bachelor of Engineering (Last-two-year GPA: 3.93/4.0) Department of Computer Science and Technology, **Tsinghua University**, Beijing, China

Experience

May 2022 - Aug. 2022	Research Intern at Microsoft Research, Redmond, WA (RiSE lab, supervisor: Saeed Maleki)
July 2018 - Sept. 2018	Summer Research Intern at USC, Los Angeles, CA (supervisor: Xuehai Qian)

Research Interests and Background

I am working on research projects related to **machine learning systems (MLSys)**, **graph analytic systems (GraphSys)** and **high-performance computing (HPC)**.

I have built a few state-of-the-art graph analytic systems (using **HPC libraries** like MPI and OpenMP, with tens of thousands of lines of code) and published the papers on top-tier conferences like ACM ASPLOS. I recently built a **distributed GPU-based GNN training framework** tailored for deep models from scratch with CUDA, cuDNN, and NCCL. I am actively working on **large language model (LLM) serving** (especially MoE serving), and familiar with open-source serving frameworks like **vLLM**. With these projects, **I have developed solid programming skills and a strong background in both High-Performance Computing and Machine Learning Systems.** I am also familiar with Computer Architecture and Operating System knowledges (e.g., out-of-order processors), and can use them to design and develop highly-optimized software systems.

Besides, personally, I am a competitive programmer and familiar with various alogrithms and data structures. I won the silver medalist (I was ranked top 100 in China) in National Olympiad in Informatics 2014, China (NOI'2014).

Skills

Language	Chinese (native), English (TOEFL:104(R30,L27,S22,W25))
Programming	Familiar with **C/C++**, **Python** Basic knowledge about VHDL, SQL, C#, JavaScript, Java, MATLAB etc.
Tools/Frameworks	Familiar with **MPI**, **OpenMP**, **CUDA**, **cuDNN**, **NCCL**, **cuBLAS/cuSPARSE**, **PyTorch (C++ extension)** Basic knowledge about Git, Tensorflow, LaTeX, Numpy, Lex&Yacc, clang-llvm, Make, CMake, etc. I am a fast learner in new programming languages, algorithms and frameworks.

Preprints

J. Chen, Z. Chen, X. Qian, GNNPipe: Scaling Deep GNN Training with Pipelined Model Parallelism (arXiv 2023, link).

Publications or Accepted Papers

J. Chen, X. Qian, Khuzdul: Efficient and Scalable Distributed Graph Pattern Mining Engine. The 28nd Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'2023).

J. Chen, X. Qian, DecoMine: A Compilation-based Graph Pattern Mining System with Pattern Decomposition. The 28nd Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'2023).

G. Rao, J. Chen, J. Yik, X. Qian, SparseCore: Stream ISA and Processor Specialization for Sparse Computation. The 27nd Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'2022).

Y. Zhuo*, J. Chen* (* equal contribution), G. Rao, Q. Luo, Y. Wang, H. Yang, D. Qian, X. Qian, Distributed Graph Processing System and Processing-In-Memory Architecture with Precise Loop-Carried Dependency Guarantee. ACM Transactions on Computer Systems (TOCS), Volume 37, Issue 1-4, Artical No. 5, 2021.

Y. Zhuo*, J. Chen* (* equal contribution), Q. Luo, Y. Wang, H. Yang, D. Qian, X. Qian, SympleGraph: Distributed Graph Processing with Precise Loop-carried Dependency Guarantee. The 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'2020).

Completed HPC System Research Projects

May 2023 – Sept. 2021 | GNNPipe: Scaling Deep GNN Training with Pipelined Model Parallelism (paper in submission, first author)

Advisor: Prof. Xuehai Qian

Problem: Existing distributed GNN training frameworks failed to scale to deep models like GCNII.

Our Solution: We theoretically analyze the worst-case communication complexity of existing solutions, and point out that the use of data parallelism is the root cause. We propose to leverage layer-level model parallelism instead, which is proved by us to have a lower communication complexity. We also adopted a set of GNN-specific techniques to tackle the unique challenges for model parallelism.

System Implementation: We built a new distributed GNN training system **from scratch** on top of **CUDA**, **cuDNN**, **cuBLAS**, **cuSPARSE**, **NCCL**, and **MPI** with **roughly 16K lines of C++ code**, and outperforms SOTA systems (e.g., DGL) significantly for deep models.

Aug. 2021 – July 2020 | SparseCore: Stream ISA and Processor Specialization for Sparse Computation (paper accepted by ASPLOS'22, second author)

Advisor: Prof. Xuehai Qian

Problem: Computation on sparse data is becoming increasingly important for many applications. Recent sparse computation accelerators are designed for specific algorithm/application, making them inflexible with software optimizations.

Our Solution: We propose SparseCore, the first general-purpose processor extension for sparse computation that can flexibly accelerate complex code patterns and fast-evolving algorithms. 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. The simulation results show that SparseCore achieves significant speedups for sparse tensor computation and graph pattern computation.

Aug. 2021 – June 2020 | Khuzdul: An Efficient and Scalable Distributed Graph Pattern Mining Engine (paper accepted by ASPLOS'23, first author)

Advisor: Prof. Xuehai Qian

Problem: Existing distributed Graph Pattern Mining (GPM) systems leverage effective but inefficient techniques to optimize the communication. Therefore, only a small portion of CPU cycles are used for actual computation.

Our Solution: We proposed to take both the communication problem and the system efficiency issue into account for distributed GPM systems design. Specifically, we proposed a new lightweight task abstraction for GPM and a new set of lightweight communication optimizing techniques.

System Implementation: We implemented Khuzdul, an efficient and scalable distributed GPM engine on top of MPI and OpenMP with **roughly 5000 lines of C++ code**. Khuzdul outperforms previous SOTA by at least one order of magnitude.

Aug. 2020 – Aug. 2019 | DecoMine: A Compilation-based Graph Pattern Mining System with Pattern Decomposition (paper accepted by ASPLOS'23, first author)

Advisor: Prof. Xuehai Qian

Problem: Existing Graph Pattern Mining (GPM) systems are slow due to its adoption of general but inefficient GPM algorithms. It is unkown how to incorporate pattern decomposition, a SOTA pattern counting algorithm, into existing GPM systems.

Our Solution: We proposed to solve the problem with compilation techniques. Guided by the accurate cost models proposed by us, our compiler can automatically make hard and performance-critical decisions for pattern decomposition, conduct various optimizations, and generate high-performance executable binaries for any user-specified GPM application.

System Implementation: Our high-performance compilation-based systems are implemented with **roughly 10K lines of C++ code**, and outperforms previous SOTA by orders of magnitude.

Dec. 2019 – Aug. 2019 | GraphS: Eliminating Redundant Computation and Communication in PIM-Based Graph Processing with Dependence Scheduling (paper appeared in TOCS as an extension version of our SympleGraph PLDI'20 paper, co-first author)

Advisor: Prof. Xuehai Qian

We further explore the research opportunities of enforcing precise loop-carried dependency in PIM-based graph processing architectures. To this end, we designed and implemented two PIM-based graph processing architectures (GraphS and its variant GraphSR) supporting dependency propagation. The proposed architectures dramatically over-perform any existing PIM-based systems.

Apr. 2019 | SympleGraph: Distributed Graph Processing with Precise Loop-Carried Dependency Guarantee (paper appeared in PLDI'20, co-first author)

Jul. 2018 | *Advisor: Prof. Xuehai Qian*

Problem: We observed that existing distributed graph processing systems (supporting graph tasks like PageRank) failed to enforce precise loop-carried dependency, leading to computation and communication redundancy.

Our Solution: We proposed a new distributed system with a precise loop-carried dependency guarantee, and also proposed a set of techniques to offset of overhead for maintaining the loop dependencies.

System Implementation: We built our new system with MPI and OpenMP with C++, which outperforms state-of-the-art baselines (Gemini and D-Galois) by up to 2.30x and 7.76x.

Honors and Awards

Aug. 2019	Annenberg Fellowship, University of Southern California
Oct. 2018	Award for **Academic Excellence**, Tsinghua U
	Hengda Scholarship
	Scholarship for Technology Innovation and Excellence, Tsinghua U
Apr. 2018	**Outstanding Winner**(Highest award, 16/10000), ICM Contest, COMAP
	Xiao Shu-tie Scholarship for Applied Mathematics, In award of ICM
Oct. 2017	Scholarship for Academic Progress, Tsinghua U
May. 2017	3rd Prize, Hua-Luogeng Mathematical Modeling Contest, Tsinghua U
Aug. 2014	**Silver Medalist, National Olympiad in Informatics, China**

Course Projects

Dec. 2017 | A THCO-MIPS CPU on FPGA

Nov. 2017 | *Team Leader, Computer Organization Course Project*

This is one of the coolest projects I finished during my undergraduate career at Tsinghua University! We designed a THCO-MIPS 16-bit (a variant of MIPS) CPU in **VHDL** with 5-stage pipeline and eventually run it on a real FPGA device and executed various programs. We also supported external devices like PS2 keyboard, VGA monitor, etc.; (The Github Repo)

My responsibility: In charged of the overall architecture design and several modules (controller & hazard detectors)

Mentored Undergraduate Students

Sept. 2021 - Aug. 2022	Zhuoming Chen (Tsinghua, currently a Ph.D. student@CMU)
June 2021 - Sept. 2021	Jingjia Luo (Tsinghua, currently a Ph.D. student@Tsinghua)
June 2020 - Sept. 2020	Dingyuan Cao (Tsinghua, currently a Ph.D. student@UIUC)
June 2020 - Sept. 2020	Sean Syed (University of Southern California)

Teaching Activities

Spring 2021 Teaching Assistant of USC EE-451 (Parallel and Distributed Computation)