Lequn Chen

lqchen@cs.washington.edu • abcdabcd987.com • abcdabcd987

Education

University of Washington **Seattle**
Ph.D., Computer Science *Sep 2018–(Expected) Aug 2023*
Research Interests: Distributed Systems, Operating Systems, Machien Learning Systems
Advisor: Prof. Arvind Krishnamurthy

ACM Honors Class, Shanghai Jiao Tong University **Shanghai**
Bachelor, Computer Science *Sep 2014–Jun 2018*
Advisors: Prof. Weinan Zhang, Prof. Gui-Rong Xue, and Prof. Yong Yu

Publications

1. **Nexus: A GPU Cluster Engine for Accelerating Neural Networks Based Video Analysis** (SOSP'19)
   - Haichen Shen, **Lequn Chen**, Yuchen Jin, Liangyu Zhao, Bingyu Kong, Matthai Philipose, Arvind Krishnamurthy, Ravi Sundaram
2. **ADARES: Adaptive Resource Management for Virtual Machines** (arXiv 1812.01837)
   - Ignacio Cano, **Lequn Chen**, Pedro Fonseca, Tianqi Chen, Chern Cheah, Karan Gupta, Ramesh Chandra, Arvind Krishnamurthy
3. **Enabling Strong Database Integrity using Trusted Execution Environments** (arXiv 1801.01618)
   - Kai Mast, **Lequn Chen**, Emin Gün Sirer
4. **Scaling Databases through Trusted Hardware Proxies** (SysTEX'17)
   - Kai Mast, **Lequn Chen**, Emin Gün Sirer

Industry Experience

Microsoft Research **Redmond, WA**
*Research Intern, **WatchFor** Project* *Jun 2021–Sep 2021*
- Investigated into the Pareto frontier of accuracy-latency trade-off of transfer learning models.
- In-depth study of how to perform transfer learning and neural architecture search effectively and efficiently.
- Explored advanced compiler optimization opportunities and challenges for the Pareto frontier models, e.g., GPU memory sharing across models, layer-based optimization caching.

Google **Kirkland, WA**
*Software Engineering Intern, **Tango** Team* *Jun 2019–Sep 2019*
- Added a new feature to the Memcache service: replication.
  - Reasoned about consistency guarantees of the new feature.
  - Implemented with 10k lines of C++ code.
  - Covered by unit tests and integration tests.
- In-depth discussion of the *Virtual Object Set* feature of the next-generation Tango.

Programming Skill Set

I am experienced with C++, Python, Rust, systems programming, asynchronous programming, concurrent computation (including multi-threading, and in distributed systems), high-performance network programming (RDMA), and solving difficult bugs.

Prior Research Experiences

Systems Lab — **Cornell University**
*Visiting Research Intern, advised by **Prof. Emin Gün Sirer** and **Kai Mast*** — *Jul 2017–Dec 2017*

- Worked on a database that provides blockchain-like guarantees of data integrity using *Trusted Execution Environments*.
  - Implemented large parts of the prototype on Intel SGX. Boosted the performance inside the SGX enclave.
  - Increased the throughput of multi-client read workload 30x and reduced the latency by 40%.
  - Implemented transaction support with optimistic concurrency control.
  - Optimized query optimizer and executor, reducing cost of join operation to almost constant in typical workloads.
  - Found and solved dozens of deadlocks and data races in the initial version of the code.
  - Designed benchmarks and conducted experiments on a distributed testbed.

APEX Data & Knowledge Management Lab — **Shanghai Jiao Tong University**
*Undergraduate Researcher, advised by **Prof. Weinan Zhang*** — *Mar 2017–Jun 2017*

- Worked on Computational Advertisement. Built a machine learning pipeline for an advertisement exchange startup.
  - Designed and trained a *Click-Through Rate* (CTR) estimation model.
  - Integrate the model with the startup's *Real-Time Bidding* (RTB) software stack.

Tianrang Network Technology Co.,Ltd — **Shanghai**
*Research Intern, advised by **Prof. Gui-Rong Xue*** — *Jun 2016–Mar 2017*

- Worked on a program Yi playing board game Go similar to Google DeepMind's *AlphaGo*. Yi runs Monte-Carlo tree search algorithm, deep neural network, and reinforcement learning algorithms.
  - Designed and Implemented a distributed system running both CPU and GPU workers on multiple machines.
  - Reduced the network latency and increased single-machine performance.
  - Refactored the code base. Trained and tuned neural networks. It could beat entry-level professional human players.

Teaching Experiences

Distributed Systems: Teaching Assistant — *Autumn 2019*

Compilers — *Spring 2017*
Student Instructor — https://acm.sjtu.edu.cn/compiler2017

- Led the teaching assistant team. Re-designed assignments.
- Built a *Continuous Integration* (CI) system abcdabcd987/acm-compiler-judge, automatically testing students' new commits and updating the leaderboard.

Principle and Practice of Computer Algorithms — *Summer 2016*
Student Instructor — https://acm.sjtu.edu.cn/wiki/PPCA_2016

- Built an online judge system for algorithm exams.
- Led a group of students to implement simplified MapReduce and Google File System. Deployed and benchmarked them on all machines of the computer room.

C++ Programming: Teaching Assistant — *Autumn 2015*

Highlighted Projects

Compiler 2016 abcdabcd987/compiler2016 — *May 2016*

- Compiles C-like language to MIPS assembly, featuring code generation and optimization, including graph coloring register allocation, and transformations in *Static Single Assignment* form.

Course Works

CSE547 Machine Learning for Big Data: Grade 3.9/4.0 — *Spring 2020*
CSE552 Parallel and Distributed Systems: Grade 4.0/4.0 — *Spring 2019*
CSE505 Computer Security and Privacy: Grade 3.9/4.0 — *Winter 2019*
CSE505 Principles of Programming Languages: Grade 4.0/4.0 — *Autumn 2018*
CSE550 Introduction to Computer Systems Research: Grade 4.0/4.0 — *Autumn 2018*