

JIARUI LI

✉ jiaruil2@andrew.cmu.edu · ☎ (+1) 878-999-6130 · 🌐 willendless

🎓 EDUCATION

Carnegie Mellon University, Pittsburgh, PA

August.2021 – May.2023(Expected)

M.S. in Information Networking, GPA 4.0/4.0

Core courses Storage Systems, Parallel Computer Architecture, Intro to Computer Systems

University of California, Berkeley, CA, USA

Jan.2020 – August.2020

International Exchange Program in EECS, GPA 3.97/4.0

Core courses OS and Systems Programming, Computer Security, Digital Design and Integrated Circuits

East China Normal University, Shanghai, China

Sep.2017 – July.2021

B.E. in Software Engineering, GPA 3.81/4.0

👥 EXPERIENCE

PingCAP: TinyKV Study Camp

Team Leader, Nov.2021 – present

- Implemented a horizontally scalable, highly available multi raft-based key-value distributed storage engine
- Built a load-balancing scheduler to support dynamic configuration change and region split.
- Constructing a multi-version concurrency control layer to support distributed transaction.

Ping An Healthcare and Technology Inc.

Software Engineer Intern, Sep.2020 – Dec.2020

- Focused on source code, architecture and operation of Etcd, a strongly consistent, distributed KV store and introduce the internals to colleagues.
- Designed an active-active configuration for two Etcd clusters located in two data centers to improve system reliability.

⚙️ PROJECTS

CloudFS: a hybrid cloud backed file system with snapshot support.

FUSE, AWS S3, C++

- Constructed a hybrid user-level file system by storing small files in local and large files in Amazon S3 cloud storage.
- Used the content-based rabin fingerprinting to deduplicate large files to decrease the capacity cost. Also implemented snapshot to allow failure recovery and a ssd-based persistent cache to reduce S3 operations cost.

MyFTL: a page mapping log structured flash translation layer.

SSD, C++

- Implemented a log structured flash translation layer to improve the endurance of the SSD.
- Used a greedy garbage collection policy to reduce write amplification and improve performance.
- Optimized the data structure and limited the memory cost to below 200kB and the write amplification to below 1.2.

RustOS: a Rust AArch64 operating System in Raspberry Pi 3.

AArch64, Rust

- Implemented a AArch64 embedded operating system for Raspberry Pi with virtual memory support using Rust.
- Constructed a read only FAT32 file system and a LRU buffer cache to improve its performance.
- Implemented several drivers, including GPIO, UART, MailBox, etc. And able to render fonts on HDMI devices.

CUDA Render: a simple CUDA circle Render.

CUDA, GPU

- Partitioned images into several boxes and mapped a dedicated CUDA thread block to each for better load balancing.
- Parallelized the task across circles within a block and utilized block shared memory to achieve 20x speedup on average.

DonkeyRISC-V: a 3-stage RISC-V pipeline processor.

FPGA, RISC-V, Verilog

- Designed a 3-stage (ID, EX, WB) RISC-V CPU with RV32I and CSR instructions support using Verilog. Then optimized its performance to 76.92MHz, with CPI equals to 1.18.
- Implemented and integrated a hardware-accelerated 2D convolutional filter and achieved over 100x speedup.

💻 SKILLS & INTERESTS

- Programming Languages: C/C++, Go, Java, Kotlin, Rust, Python, Verilog
- Platforms: Linux, FPGA, Android, x86, AArch64
- Interests: Operating Systems, Embedded Systems, Storage Systems, Distributed Systems, Computer Architecture