Zisheng Cai

zisheng.cai2000@gmail.com | linkedin.com/in/zisheng-cai | github.com/llbbllbb2000 | +1 (712) 650 0367

Education

Brandeis University 08/2022 – 05/2024

M.S. in Computer Science – 3.86/4.00 GPA

- Relevant Courses: Java, Data Structures, Machine Learning, Web Design, Operating System, Computer Networking

Dalian University of Technology 09/2018 – 06/2022

B.E. in Environmental and Ecological Engineering – 86.9/100 GPA

- The Second Prize Scholarship, The First Prize in National Olympiad in Informatics in Provinces
- Relevant Courses: C++ programming; Python for Data Analysis; Mathematical Modeling with MATLAB

Technical Skills

Programming languages: C/C++, Python, Java, JavaScript (HTML/CSS), SQL, Bash, R, MATLAB

Frameworks & Tools: Git, Django, Flask, React.js, Express.js, Node.js, MongoDB, PostgreSQL, MySQL, NoSQL

Industry Experience

Brandeis University **Research Assistant** 04/2023 – 08/2023

- Built a website API using Python **Django** and **PostgreSQL** as the backend database
- Created **API wrappers** in multiple programming languages, including Python, R, and Perl

XPeng Motors **Intern, Autonomous Driving Center** 12/2021 – 05/2022

- Utilized Python to process and **visualize** a substantial dataset comprising **20 million data** points to analyze algorithm performance across more than **10 diverse scenarios**, providing valuable insights and optimizations.
- Designed the database structure for a groundbreaking parking algorithm using a **NoSQL JSON** database, playing a crucial role in supporting subsequent algorithm enhancements and optimizations.
- Applied Dijkstra and A* to optimize autonomous parking routes, ensuring the most efficient and shortest paths are identified.

Coding Projects

Tiny Web Server

- Devised the concurrency model of the web server by leveraging **IO multiplexing** with epoll (utilizing both **Level-Triggered** (LT) and **Edge-Triggered** (ET) modes), a **thread pool** for efficient resource management, and **non-blocking** socket operations.
- Developed a sophisticated **finite-state machine** to parse HTTP GET and POST requests, and implemented **SQL** queries to support user sign-up and log-in.
- Constructed a **synchronous logging system** and a timing wheel mechanism to close timeout connections.

Bustub Database

- Established a Buffer Pool Manager based on **LRU-K Cache**, and achieved a PageGuard class to ensure **RAII**, enhancing system reliability by automatically cleaning pinned pages and releasing latches.
- Processed SQL queries following the **volcano model**, enabling support for common SQL operations such as INSERT, DELETE, UPDATE, JOIN, and AGG.
- Applied a **Two-Phase locking (2PL)** mechanism in the lock manager, achieving three distinct isolation levels: **read uncommitted, read committed, and repeatable read**.

TCP/IP Protocol

- Completed a **TCP/IP Protocol, IP Router, and ARP Protocol** based on the architecture of Standford CS144.
- Created byte streams and stream reassembly techniques utilizing **deque and vector** in C++, supporting TCP sliding window and overtime retransmission.
- Accomplished TCP **3-way handshake** and **4-way handshake** via ACK, SYN, and FIN.