

HAOHUA LIU

howardlau1999@hotmail.com · <https://liuhaohua.com> · GitHub @howardlau1999

EDUCATION

Sun Yat-sen University, Software Engineering, **B.Eng**

Sep. 2017 - Jun. 2021

GPA: 4.2/5.0 Rank 8/175 (Top 5%)

SKILLS

- **Programming Languages:** C, C++, Python, Golang, JavaScript, TypeScript, SQL
- **Operating System, Databases and Engineering Skills:** Linux, MySQL, TiDB, Git
- **Keywords:** Kubernetes, Docker, Angular, PyTorch
- **Course Projects:** MIT 6.824 2021 Raft Lab, MIT 6.824 2015 Paxos Lab, CMU 15-445 2020 Bustub RDBMS Lab, Berkeley CS186 2021 RookieDB Lab, [SJTU chCore OS Labs](#), THU uCore OS Labs

WORK EXPERIENCE

PingCAP, R&D Intern

Jul. 2020 - Oct. 2020

- Implemented the auto-scaling functionality for the cloud-native database TiDB, which is aimed to provide users with a easy way to improve machine resource utilization of their database deployment. It helps shutdown under-utilized instances of the database to reduce users' costs and start up new instances if the database is overloaded to deal with burst traffic.
- The auto-scaling framework is implemented in both Placement Driver and TiDB Operator side. The Placement Driver is responsible to query metrics from Prometheus to calculate the current resource utilization and compare them against the target utilization specified by users. Afterwards, the Placement Driver will send a scaling plan to the TiDB Operator.
- Enabled TiDB Operator to reconcile with the scaling plan by mutating CRDs in the Kubernetes cluster.
- Wrote end-to-end tests to make sure both the Placement Driver and TiDB Operator function correctly.
- Promoted as Reviewer for the Placement Driver and TiDB Operator projects.

WeChat, Tencent, Backend Engineer Intern

May. 2020 - Jul. 2020

- Implemented a persistent Bitcask KV storage server from scratch as a demo project.
- Developed an access control module for WeChat Miniprogram System. Users can control the available miniprograms and use time on an admin portal.
- Refactored the module that allow multiple customer service talking using one public account using modern APIs.

PROJECTS

YatCPU

Nov. 2021 - Jun. 2022

- Implemented an open-source three-stage RISC-V CPU with interrupt and bus functionality in Chisel 3, open-sourced at <https://github.com/howardlau1999/yatcpu>.
- Ported RISC-V compliance test and CoreMark benchmarks to the CPU for verification. In addition, a Tetris game is also written served as an interesting application.
- Wrote a step-by-step tutorial with automatic unit tests on implementing a CPU from scratch as well as running applications on it, which is used as labs of Computer Organization Principles course.

Autograder

Dec. 2021 - Feb. 2022

- Designed and developed a light-weight autograding system aimed at running automatic tests and giving timely feedback for students doing lab projects, featuring easy self-hosted deployment and low resource demanding. It uses Docker for execution, Angular for web front-end and gRPC for communication between server and graders.
- Served for automatic grading labs of Computer Organization Principles and Compiler Principles.
- <https://github.com/howardlau1999/autograder-server>

SeaweedFS Operator

Nov. 2020 - Dec. 2020

- Implemented basic functionality for SeaweedFS Operator (<https://github.com/seaweedfs/seaweedfs-operator>) that automatically sets up several components and deploys a minimal SeaweedFS (<https://github.com/chrislusf/>

[seaweedfs](#)) object storage cluster on a Kubernetes cluster. This helps users to automatically manage a complicated object storage cluster on servers and helps promote the open-source project.

Matrix Online Judge System

Jan. 2019 - Jul. 2020

- Designed, led developing and maintained an online judge system named Matrix which supports the daily teaching activities of the School of Computer Science and Engineering. It is used by over 600 students and 10 professors, autograding over 2,000 submissions per day. It is also used in final exams which demand extreme stability and performance of the system.
- Led the development and contributed most code on backends including APIs implemented in Golang and the autograder system implemented in C++ using control groups and namespace of Linux.
- Deployed the whole system on a Kubernetes cluster which runs on virtual machines managed by Proxmox Virtual Environment. This helps avoid single point of failure and enables high availability.
- Deployed monitoring stack using Prometheus for metrics and alerting and Elasticsearch for log processing.
- The autograding system is open-sourced at <https://github.com/VMatrixTeam/judge-system>

Single-threaded FPTree Implementation

May 2019

- Group project for the Database Management System course. Simulated an NVM device using Linux kernel and implemented the basic algorithm of the paper *FPTree: A Hybrid SCM-DRAM Persistent and Concurrent B-Tree for Storage Class Memory*.
- Responsible for writing the deletion algorithm for the B+-Tree and optimizing the performance of the whole implementation. Our performance ranked the 1st place among 30 groups in the class.
- <https://github.com/sysu-2019-dbms/fp-tree>

FlaPGA Mario

Apr. 2018 - Jun. 2018

- A final project for the Digital Circuit Design course. Designed and implemented a mini-game like Flappy Bird on Basys 3 FPGA board using Verilog HDL. Just for fun.
- <https://github.com/howardlau1999/flapga-mario>