Linsong Guo

No.800 Dongchuan Road ◇ Shanghai, China 200240
gls1196@sjtu.edu.cn

EDUCATION

Shanghai Jiao Tong University (SJTU) *2018 - 2022 (Expected)*
B.S. in Computer Science
Member of ACM Class, an elite CS program for the top 5% talented students.
GPA: 88/100

RESEARCH EXPERIENCE

Database System Group, Pennsylvania State University
Supervised by Prof. Xie Dong *Jun 2021 - Present*

Serverless Functions Placement for Disaggregated Architecture on Distributed Environment
I concentrate on the two issues of serverless computing under disaggregated storage architecture:

- Functions running on the compute server need several data transmissions including *get()/put()* with the storage server, which hurts end-to-end **latency** of functions.
- These *get()/put()* requests on the storage server waste some CPU dispatching requests from NIC to userspace and unpacking requests, which hurts the **throughput** of the storage server.

Shipping functions to the storage server could reduce their e2e latency. Also, this can save the storage server's CPU, thus improving throughput. However, the throughput is hurt if the saved CPU is not enough to handle compute introduced by these functions on the storage server. I focus on **the tradeoff** and explore it on **the distributed environment** to make it more practical.

Emerging Parallel Computing Center (EPCC Lab), SJTU
Supervised by Prof. Quan Chen *Jul 2020 - Jun 2021*

Eliminating Cold Startup in Serverless Computing by Sharing Containers between Functions

- Proposed an effective **inter-function container sharing policy based on startup frequency**, which helped our system to alleviate 87.9% of cold startup.
- Implemented most of the system, designed and ran experiments, especially **a large-scale evaluation in cloud environment**.

Optimizing Data Communication across Functions within Serverless Applications

- Co-proposed **an efficient workflow scheduling mechanism**, which mitigates the data transmission overhead by 50.1%.
- Designed **an adaptive storage library** which automatically chooses the most appropriate storage service between local memory and remote database for user functions.
- Designed a parser which could parse hierarchy application workflows into DAGs for better scheduling.

PUBLICATIONS

Zijun Li, Yushi Liu, **Linsong Guo**, Quan Chen, Jiagan Cheng, Wenli Zheng, Minyi Guo. *FaaSFlow: Enable Efficient Workflow Execution for Function-as-a-Service.* **ASPLOS 2022 (To Appear)**.

2nd author in *The Serverless Computing Survey: A Technical Primer for Design Architecture.* Under Review in **ACM Computing Surveys**.

OTHER EXPERIENCE

Teaching Assistant of C++ Programming Course, SJTU
instructed by Prof. Huiyu Weng *Sep 2019 - Jan 2020*

Designed some assignments, gave a lecture about the introduction to C++ programming, guided a group of students in programming and algorithms, and contributed one programming problem to the final exam.

Member in ACM-ICPC Team, SJTU
guided by Prof. Yong Yu *Jun 2018 - Jul 2019*

I was a member of a team named *Quasar*. In this team, I practiced programming and algorithms with two other members at least twice each week. And we won three gold medals (one 1st runner-up) in ACM-ICPC Asia regional contests and one gold medal in China Collegiate Programming Contest. Therefore, my programming and algorithmic abilities have been improved in the ACM-ICPC team.

PROJECTS

Java-and-C-like Language Compiler (∼16K lines in Java) [github]

Given a piece of code, the compiler could convert it into an AST, then LLVM IR, and finally RISC-V assembly. Due to my interest in exploring the compiler back-end, I added some optimizations to it, including mem2reg, inlining, CSE(Common SubExpression Elimination), LICM(Loop Invariant Code Motion), SCCP(Sparse Conditional Constant Propagation), and so on.

Replicated KV Store Based on Raft Consensus Protocol (∼1.5K lines in C++) [github]

The store could run on a cluster of servers communicated via gRPC and support basic operations, including get and put.

RISC-V CPU with 5-Stage Pipeline (∼3.7K lines in Verilog) [github]

To explore more about computer architecture, I add some components like d-cache, i-cache, and branch predictor combining BTB and BHT. The CPU could run successfully on an FPGA board.

Playing Atari Games [github]

To explore reinforcement learning, I trained several DQN models, including DoubleDQN, DuelingDQN, RainbowDQN, and so on, to play some Atari games.

HONORS AND AWARDS

1st Runner-Up, ACM-ICPC Asia Regional Contest, Nakhon Pathom Site *2018*
Gold Medal, ACM-ICPC Asia Regional Contest, Qingdao Site *2018*
Gold Medal, China Collegiate Programming Contest, Guilin Site *2018*
Silver Medal, China Collegiate Programming Contest, Final *2018*
Gold Medal, ACM-ICPC invitational Contest, Xi'an Site *2019*
Zhiyuan Honorary Scholarship, Award for top 5% students *2018, 2019, 2020*
Excellence Scholarship for Undergraduates *2019, 2020*

SKILLS

Programming Languages: C/C++ > Python > Java > Rust, x86 and RISC-V assembly

Hardware: Verilog, PC assembly