Yunxin Sun

Email: yunsun@student.ethz.ch Website: https://yunxin-sun.com

Research Interests

I am interested in **computer systems** and their interactions with computer architecture and core database systems. I am especially passionate about the following three topics: ***(i)*** machine learning for systems, ***(ii)*** next-generation storage systems, and ***(iii)*** multi-objective data centers.

Education Background

M.Sc. in Computer Science Zurich, Switzerland
ETH Zurich Sep. 2020 - Aug. 2022 (expected)

– Major in Data Management Systems, Minor in Programming Languages and Software Engineering
– Courses: Big Data, Data Management Systems, Advanced Systems Lab, Cloud Computing, Distributed Systems Lab
– GPA: 5.37/6

B.Eng. in Software Engineering Shanghai, China
Tongji University Sep. 2015 - Jul. 2020

– GPA: 88.88/100
– Ranking: Top 5%

Research Experiences

A Learning-Based Flash Translation Layer (FTL) for Solid-State Drives
Advisor: Prof. Dr. Jian Huang, PlatformX Lab *UIUC*

– We built a learning-based FTL that could learn address mapping at runtime. We coordinated this process with garbage collection (GC) so that it incurred negligible overhead. This learning-based FTL saves the memory footprint of the mapping table by up to 2.9x and improves the tail latency by 1.6x on average.
– I proposed and implemented an LSM-tree-like data structure to tolerate machine learning misprediction.
– I implemented 40% of the project.
– This work is submitted to USENIX FAST 2022 and is currently under review.

Fair and Deterministic I/O Scheduling in Modern NVMe Solid State Drives
Advisor: Prof. Dr. Onur Mutlu & Dr. Jisung Park, SAFARI Research Group *ETH Zurich*

– The key idea is to divide program and erase (P/E) operations into multiple substeps. Therefore, the SSD controller could alternately schedule each P/E substep and read request, preventing a program or erase operation from blocking a read request for too long. The write tail latency improves up to 30%.
– I implemented all of the project.
– We plan to submit this work to USENIX ATC 2022.

Prediction and Estimation of Serverless Functions' Execution Time
Advisor: Prof. Dr. Ana Klimovic, EASLab *ETH Zurich*

– We used several machine learning models including decision tree, deep neural networks to estimate and predict the execution time of serverless functions. We used collected information in the model such as the CPU usage, the memory footprint, the network traffic, and the input function parameters such as the size of an image. The predicted value could be useful for resource allocation and scheduling. On average, our model achieves 81% prediction accuracy.
– I implemented 60% of the project.

Publications

- **Yunxin Sun**, Tamar Shinar, and Craig Schroeder. ”Effective time step restrictions for explicit MPM simulation.” In Computer Graphics Forum, vol. 39, no. 8, pp. 55-67. 2020.
- Josh Urban Davis, Jun Gong, **Yunxin Sun**, Parmit Chilana, and Xing-Dong Yang. ”Circuitstyle: A system for peripherally reinforcing best practices in hardware computing.” In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, pp. 109-120. 2019.
- Guokai Zhang, Ye Luo, Dandan Zhu, Yixuan Xu, **Yunxin Sun**, and Jianwei Lu. ”Spatial pyramid dilated network for pulmonary nodule malignancy classification.” In 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3911-3916. IEEE, 2018.

Other Manuscripts

- ”A Learning-Based Flash Translation Layer for Solid-State Drives”. In submission to USENIX FAST 2022.
- ”Execution Time Prediction For Serverless Workload”. Technical Report.

Teaching and Mentoring Experiences

- Mentored two undergraduate students in the course ”*Seminar on Modern SSDs*”. Met with them weekly to check the progress of the course project.
- Gave a lecture about *MQSim*, a popular simulator for Solid State Drives, in the course ”*Seminar on Modern SSDs*”.

Skills

- **Programming Languages**: C, C++, SQL, Rust, Python, Assembly Language, Verilog, HLS, OCaml.
- **Tools & Software**: Git, CMake, perf, eBPF, PyTorch, GDB, Intel SGX, Shell, gem5, FPGA, LaTeX.