Yuxuan Zhang

Last update on August 30, 2023

zyuxuan@seas.upenn.edu • 734-846-1069 • Levine 561, 3330 Walnut Street, Philadelphia, PA (19104)

Education

University of Pennsylvania, Philadelphia, PA
PhD in Computer Science, 2019 -
Advisor: Joseph Devietti
GPA: 4.0/4.0

University of Michigan, Ann Arbor, MI
MS in Electrical Engineering, 2015 - 2017

Harbin Institute of Technology, Harbin, China
BS in Electrical Engineering, 2008 - 2012
GPA: 89.4/100.0

Employment History

VMWare Corp., Boston, MA
Software Engineer Intern, May 2022 - Aug 2022

Microsoft Research Asia, Beijing, China
Research Intern, Jan 2018 - July 2019
Mentor: Yongqiang Xiong

NVIDIA Corporation, Beijing, China
Software Engineer Intern, May 2017 - Sep 2017

Publications

OCOLOS: Online COde Layout OptimizationS.
Y. Zhang, T. A. Khan, G. Pokam, B. Kasikci, H. Litz, J. Devietti.
Proc. International Symposium on Microarchitecture (***MICRO***), Oct. 2022.

Online COde Layout OptimizationS via Ocolos
Y. Zhang, T. A. Khan, G. Pokam, B. Kasikci, H. Litz, J. Devietti.
IEEE Micro Volume 43, Issue 4, "Top Picks From the 2022 Computer Architecture Conferences", July. 2023.

Honor & Awards

Paper selected for IEEE Micro Top Picks in Computer Architecture from 2023

Outstanding Graduates Awards, *Harbin Institute of Technology, 2012*

Fuji Xerox Scholarship, *Harbin Institute of Technology, 2011*
Undergraduate GPA ranking top 1 for one academic year

Suzhou Industry Park Scholarship, *Harbin Institute of Technology, 2010*
Undergraduate GPA ranking top 2 for one academic year

Talk

OCOLOS: Online COde Layouy OptimizationS. Oct. 2022
International Symposium on Microarchitecture (MICRO), Chicago, IL.

Professional Service

Journal Reviewer
IEEE Transactions on Computers (TOC), 2023

Teaching

Teaching Assistant for CIS505 Software Systems, University of Pennsylvania, Fall 2020, Fall 2021

Projects

RPG^2: Robust Profile-Guided Runtime Prefetch Generation

Architecture+Compilers group, University of Pennsylvania

Built an online data cache prefetching system that can profile and analyze the behavior of data memory accesses and then make the decision of whether and where to insert the prefetch instructions into the running process. After prefetches inserted, RPG^2 can monitor and tune the prefetches to maximize performance. RPG2 can provide speedup up to 2.15× across all graph inputs from Stanford Network Analysis Platform on CRONO workloads.

OCOLOS: Online COde Layout Optimization System

Architecture+Compilers group, University of Pennsylvania

Built a code layout optimization tool that can optimize the code layout of datacenter applications at runtime by first profiling and analyzing the application, then producing an optimized binary and finally inserting the machine code from the optimized binary to the target application process. OCOLOS is written in C++ and utilizes the functionality provided by Meta's post-link optimizer BOLT. OCOLOS can accelerate complex datacenter workloads like MySQL and MongoDB by up to 1.41×.

Build Accelerator

Architecture+Compilers group, University of Pennsylvania

Built a system to accelerate large software builds by optimizing the code layout of the compiler binary as the build progresses. The Build Accelerator is written in Rust and also is based on Meta's BOLT. Build Accelerator can provide a 1.14× speedup, without any changes to the source code or build scripts.

Glane on GPU

Networking Research Group, Microsoft Research Asia

Built a Linux module that can expose an NVIDIA GPU's physical memory for direct data transfer via C++, and a hardware stack in System Verilog for GPUs in a device-centric cluster to buffer and transfer data. Prototyped CUDA code to perform GPU computation and data transfer in parallel without host CPU involvement.

Pre-validation during pre-copy on VMotion

Monitor team, VMWare Corp.

Offloaded the pre-validation of the destination virtual machine (VM)'s page table from Virtual Machine Monitor(VMM) space to ESXi (VMKernel, the hypervisor) after a VM is migrated from source to destination (VMotion), in order to reduce the contention of updating page tables in different VMs. Built pre-validation during the pre-copy of memory pages during VMotion to reduce the time spent on pre-validation. On the tested self-VMotion machine, the results showed that offloading pre-validation from VMM to VMKernel achieves a 2× speedup.

Skills

Programming: C/C++, Python, JavaScript, Rust, System Verilog

Language: Chinese (native), English (professional working proficiency), Japanese (intermediate level)