Jianping Zeng

Department of Computer Science
Purdue University
LWSN 3133-6, 305 N. University Street
West Lafayette, Indiana 47906

zeng207@purdue.edu
https://www.cs.purdue.edu/homes/zeng207

Research Interests

My research interests generally lie on designing more reliable computing systems, e.g., server-class systems and energy-harvesting systems, against errors such as soft errors and power failure. In the past, I have already proposed several solutions to providing near-zero run-time overhead soft error resilience and high-performance crash-consistency. In particular, I usually co-design compiler and architecture to make the solutions efficient while maintaining their minimal hardware complexity. My works are usually published in top-tier system venues, e.g., PLDI, MICRO, HPDC, ISCA, and RTSS.

Education Background

- Purdue University — West Lafayette, IN
  Ph.D. in Computer Science — *Aug. 2019 – 2024 (Expected)*
- Virginia Tech — Blacksburg, VA
  Ph.D in Computer Science; Transferred to Purdue University — *Aug. 2018 – May. 2019*
- Huazhong University of Science and Technology — Wuhan, China
  Master of Science in Computer Science and Technology — *Sept. 2014 – June. 2017*
- Central South University — Changsha, China
  Exchange Student — *Sept. 2012 – June. 2013*
- Jiaxing University — Jiaxing, China
  Bachelor of Science in Computer Science and Technology — *Sept. 2010 – June. 2014*

Publication

- **RTSS'23**: RTailor: Parameterizing Soft Error Resilience for Mixed-Criticality Real-Time Systems, Shao-Yu Huang, **Jianping Zeng**, Xuanliang Deng, Sen Wang, Ashrarul Haq Sifat, Burhanuddin Bharmal, Jia-Bin Huang, Ryan Williams, Haibo Zeng, and Changhee Jung
- **MICRO'23**: Persistent Processor Architecture, **Jianping Zeng**, Jungi Jeong, and Changhee Jung
- **MICRO'23**: SweepCache: Intermittence-Aware Cache on the Cheap, Yuchen Zhou, **Jianping Zeng**, Jungi Jeong, Jongouk Choi, and Changhee Jung
- **ISCA'23**: WL-Cache: Write-Light Cache for Energy Harvesting Systems, Jongouk Choi, **Jianping Zeng**, Dongyoon Lee, Changwoo Min, and Changhee Jung
- **HPDC'22**: Capri: Compiler and Architecture Support for Whole-System Persistence, Jungi Jeong, **Jianping Zeng**, and Changhee Jung
- **MICRO'21**: ReplayCache: Enabling Volatile Caches for Energy Harvesting Systems, **Jianping Zeng**, Jongouk Choi, Xinwei Fu, Ajay P. Shreepathi, Dongyoon Lee, Changwoo Min, and Changhee Jung
- **MICRO'21**: Turnpike: Lightweight Soft Error Resilience for In-Order Cores, **Jianping Zeng**, Hongjune Kim, Jaejin Lee, and Changhee Jung
- **PLDI'20**: Compiler-Directed Soft Error Resilience for Lightweight GPU Register File Protection, Hongjune Kim, **Jianping Zeng**, Qingrui Liu, Mohammad Abdel-Majeed, Jaejin Lee, Changhee Jung

Working Experience

- Purdue University — West Lafayette, IN
  Graduate Research Assistant — *Aug 2019 - Present*
  Devoted to creating a series of novel compiler optimizations and architectural techniques to improve the reliability (Turnpike;MICRO'21), performance (PPA;MICRO'23, ReplayCache;MICRO'21) of general-purpose and intermittent computing systems.
- Samsung Memory Solutions Lab, USA — San Jose, CA
  Research Intern — *May 2023 - Aug 2023*
  - **Proposed to improve the energy-efficiency of ECC DRAM**
- Alibaba DAMO Academy, USA — Sunnyvale, CA
  Research Intern — *May 2022 - Aug 2022*
  - **Optimizing memory fences for ARM server cores**: Worked on optimizing the ARM memory fence instructions, which is particular of importance for the server processors to support up to 256 cores efficiently.
- Virginia Tech — Blacksburg, VA
  Graduate Research Assistant — *Aug 2018 - Aug 2019*
- Alibaba T-Head Semiconductor — Hangzhou, China
  Compiler Engineer — *Dec 2017 - July 2018*
  - **C-Sky backend of GCC/LLVM**: Engaged in improving GCC backend and developing LLVM backend for C-Sky processors, and tuning GCC/LLVM optimizations as well.
- Alibaba Taobao BU — Hangzhou, China
  Senior Software Engineer — *June 2017 - Dec 2017*
  - **Static Program Analyzer**: Finished a static analyzer for C/C++/Objective-C/Objective-C++ based on Clang, which is widely used for statically validating the correctness and ensuring the robustness of Taobao App (the most popular online shopping application in China).
- Alibaba Group — Hangzhou, China
  Software Engineer Intern — *June 2016 - Aug 2016*
  - **Static-JavaScript based-on Google V8 virtual machine**: Tuned the fore-end of Google V8 virtual machine to add static type annotation for JavaScript language, which would be beneficial to the following compiler optimizations and code generations.
- Huazhong University of Science and Technology — Wuhan, China
  Teaching Assistant — *Sept 2014 - June 2015*
  - **Teaching assistant for compiler course**

Services

- **Organization Committee**

  Languages, Compilers, Tools and Theory of Embedded Systems (LCTES): 2020 (Web Chair).

- **Journal Reviewer**

  Computer Architecture Letter (CAL): 2022.

- **Sub-reviewer**

  Architectural Support for Programming Languages and Operating Systems (ASPLOS): 2020, 2022-2023.

  USENIX Annual Technical Conference (ATC): 2020.

  International Conference on Compilers, Architectures, and Synthesis for Embedded Systems (CASES): 2020, 2023.

  International Conference on Compiler Construction (CC): 2020.

  International Symposium on Computer Architecture (ISCA): 2023.

  Principles and Practice of Parallel Programming (PPoPP): 2020-2023.

  International Symposium on Microarchitecture (MICRO): 2020, 2022.

  International Symposium on High-Performance Parallel and Distributed Computing (HPDC): 2020, 2022.

  International Symposium on Code Generation and Optimization (CGO): 2020, 2022.

Achievements

- NVMW Travel Award: 2022, 2023.

Projects

- **XCC Research Compiler**: An open source C/C++/Fortran Compiler written in Java for research purpose, which accepts LLVM IR as input and produces ARM/x86 assembly code. The main motivation of this project is greatly leveraging the safe programming features provided by managed language, e.g., Java, to free compiler engineer so that they can focus on problem-specific coding instead of being distracted by other irrelevant bugs, e.g., segmentation faults.