# Jiacheng Ma

2100 Logic Dr
San Jose, CA 95124, USA

i@jcma.me
https://jcma.me

## Research Interests

My research is at the intersection of hardware and software. I am interested in improving the programmability, reliability, debuggability, and deployability of heterogeneous systems by building systems support such as hypervisors, compilers, debuggers, and runtimes. I am also interested in system virtualization and software-hardware co-design.

## Education

**University of Michigan** *Ann Arbor, MI, USA*

*Ph.D. in Computer Science and Engineering* Sept. 2018 – May 2024
Thesis: Systems and Debugging Supports for Hardware Designs
Advisor: Prof. Baris Kasikci

*M.S.E. in Computer Science and Engineering* Sept. 2018 – Apr. 2021

**Shanghai Jiao Tong University** *Shanghai, China*

*B.E. in Software Engineering* Sept. 2014 – June 2018
Thesis: Efficient GPU Live Migration Optimized by Software Dirty Page for Full Virtualization
Advisor: Prof. Zhengwei Qi

## Employment

**Advanced Micro Devices** *San Jose, CA, USA*

*Sr. Software Engineer* Jan. 2024 – Present
Development of AMD's next generation XDNA NPUs

**University of Michigan** *Ann Arbor, MI, USA*

*Graduate Student Research Assistant & Graduate Student Instructor* Sept. 2018 – Dec. 2023
Research on hardware security and reliability

- I designed and developed **Vega** [1], a system to construct concise software testsuite that detects aging-related silent data corruptions (SDCs) inside a chip. Vega empowers frequent and routine SDC detection at application runtime, thereby improving the effectiveness of transistor aging failure detection.
- I conducted an FPGA bug study and explored debugging techniques for FPGA [3]. In this work, I performed a comprehensive study on real-world bugs in FPGA projects, classified these bugs based on their root causes and symptoms, and proposed techniques to help bug localization.
- I participated in the development of **Dolma** [5], the first secure processor that provide automatic, comprehensive protection against all known variants of transient execution attacks.
- I participated in the development of **Vidi** [2], a record/replay system for FPGA applications. Vidi can not only record and replay a hardware execution on an FPGA, but also enable other developers to build more sophisticated development tools.

Research on systems supports for FPGAs

- I designed and developed **Optimus** [8, 4], the first hypervisor for shared-memory FPGA platforms. With supports for both spatial and temporal multiplexing, Optimus helps data center operators to deploy multiple shared-memory accelerators on the same FPGA, thus maximizing resource utilization.

Research on other systems-related topics

- I participated in the development of **Execution Reconstruction** [7], a hardware-assisted bug reproduction technique. This work combines online recording and offline symbolic execution to recover failing program executions with low overhead and high accuracy.

**Alibaba Group (U.S.)** *Sunnyvale, CA, USA*
*Research Intern* May 2022 – Aug. 2022
Research on remote memory

- I prototyped a hardware-assisted remote memory system that allows one computer to borrow the system memory from others in a data center. Supervisor: Dr. Dimin Niu.

**VMware Research** *Remote*
*Intern* May 2021 – Aug. 2021
Research on remote memory

- I prototyped an RDMA-based remote memory system enabling multiple computers to access shared data using the same pointer. Supervisors: Dr. Marcos K. Aguilera & Dr. Irina Calciu.

**Intel Lab** *Remote*
*Graduate Technical Intern* June 2020 – Aug. 2020
Research on far memory

- I explored the architecture supports for cold data on far memory. Supervisor: Dr. Sanjay Kumar.

**Intel Asia-Pacific Research & Development Ltd** *Shanghai, China*
*Software Developer Intern* July 2016 – June 2018
Research on system virtualization

- I developed **gMig** [11, 10], the first system that enables GPU live migration for full virtualization. gMig enables seamlessly migrating vGPUs for cloud applications such as virtual desktops, cloud gaming farms, cloud transcoding services, etc. Supervisor: Dr. Eddie (Yaozu) Dong.
- I created an infrastructure to pack OS image with pre-installed GPU virtualization products.

**Shanghai Jiao Tong University** *Shanghai, China*
*Research Assistant & Teaching Assistant* Jan. 2016 – June 2018
Research on GPU virtualization

- I participated in the development of **gScale** [13, 12, 14], which increased the scalability of Intel's GPU virtualization solution for 5× with minimum overhead. gScale makes GPU virtualization more consolidated, since more VMs with vGPU can be depolyed on one machine. Joint work with Intel.
- I participated in the design of **gRemote** [6, 9], which enables API-forwarding based cloud rendering for OpenGL applications on a resource pool.

## Peer-Reviewed Publications

[1] Proactive Runtime Detection of Aging-Related Silent Data Corruptions: A Bottom-Up Approach. Jiacheng Ma, Majd Ganaiem, Madeline Burbage, Theo Gregersen, Rachel McAmis, Freddy Gabbay, and Baris Kasikci. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2024.

[2] Vidi: Record Replay for Reconfigurable Hardware. Gefei Zuo, Jiacheng Ma, Andrew Quinn, and Baris Kasikci. *Proceedings of the 28th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2023.

[3] Debugging in the Brave New World of Reconfigurable Hardware. Jiacheng Ma, Gefei Zuo, Kevin Loughlin, Haoyang Zhang, Andrew Quinn, and Baris Kasikci. *Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022.

[4] MEGATRON: Software-Managed Device TLB for Shared-Memory FPGA Virtualization. Yanqiang Liu, Jiacheng Ma, Zhengjun Zhang, Linsheng Li, Zhengwei Qi, and Haibing Guan. *The 58th Design Automation Conference*, 2021.

[5] DOLMA: Securing Speculation with the Principle of Transient Non-Observability. Kevin Loughlin, Ian Neal, Jiacheng Ma, Elisa Tsai, Ofir Weisse, Satish Narayanasamy, and Baris Kasikci. *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[6] gRemote: Cloud rendering on GPU resource pool based on API-forwarding. Dongjie Tang, Linsheng Li, Jiacheng Ma, Xue Liu, Zhengwei Qi, and Haibing Guan. *Journal of Systems Architecture*, 116:102055, 2021.

[7] Execution reconstruction: Harnessing failure reoccurrences for failure reproduction. Gefei Zuo, Jiacheng Ma, Andrew Quinn, Pramod Bhatotia, Pedro Fonseca, and Baris Kasikci. *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 1155–1170, 2021.

[8] A Hypervisor for Shared-Memory FPGA Platforms. Jiacheng Ma, Gefei Zuo, Kevin Loughlin, Xiaohe Cheng, Yanqiang Liu, Abel Mulugeta Eneyew, Zhengwei Qi, and Baris Kasikci. *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020.

[9] gRemote: API-Forwarding Powered Cloud Rendering. Dongjie Tang, Yun Wang, Linsheng Li, Jiacheng Ma, Xue Liu, Zhengwei Qi, and Haibing Guan. *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, pages 197–201, 2020.

[10] gMig: Efficient vGPU Live Migration with Overlapped Software-based Dirty Page Verification. Qiumin Lu, Xiao Zheng, Jiacheng Ma, Yaozu Dong, Zhengwei Qi, Jianguo Yao, Bingsheng He, and Haibing Guan. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[11] gMig: Efficient GPU Live Migration Optimized by Software Dirty Page for Full Virtualization. Jiacheng Ma, Xiao Zheng, Yaozu Dong, Wentai Li, Zhengwei Qi, Bingsheng He, and Haibing Guan. *Proceedings of the 14th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 31–44, 2018.

[12] Scalable GPU Virtualization with Dynamic Sharing of Graphics Memory Space. Mochi Xue, Jiacheng Ma, Wentai Li, Kun Tian, Yaozu Dong, Jinyu Wu, Zhengwei Qi, Bingsheng He, and Haibing Guan. *IEEE Transactions on Parallel and Distributed Systems*, 29(8):1823–1836, 2018.

[13] gScale: Scaling up GPU Virtualization with Dynamic Sharing of Graphics Memory Space. Mochi Xue, Kun Tian, Yaozu Dong, Jiacheng Ma, Jiajun Wang, Zhengwei Qi, Bingsheng He, and Haibing Guan. *Proceedings of the 2016 USENIX Conference on Usenix Annual Technical Conference*, pages 579–590, 2016.

## U.S. Patent

[14] Jiacheng Ma, Haibing Guan, Zhengwei Qi, and Yongbiao Chen. System, apparatus, and method for optimizing a scalable GPU virtualization, October 1 2019. US Patent 10,430,991.

## Teaching

**Advanced Operating Systems (EECS 582)** *Ann Arbor, MI, USA*
GSI for Prof. Baris Kasikci at the University of Michigan. Sept. 2021 – Dec. 2021

**Programming and Data Structure (SE 117)** *Shanghai, China*
TA for Prof. Zhengwei Qi at Shanghai Jiao Tong University. Feb. 2016 – June 2016

## Academic Mentoring

| | |
|---|---|
| Xiaohe Cheng (HKUST BSc → Google) | *2019* |
| Abel Mulugeta Eneyew (Addis Ababa Institute of Technology) | *2019* |
| Haoyang Zhang (UMich BSc → UIUC PhD) | *2021 – 2022* |
| Wentao Zhang (SJTU MSc → UIUC PhD) | *2022* |
| Yin Yuan (UMich BSc/MSc) | *2022* |
| Madeline Burbage (UW PhD) | *2023 – 2024* |

## Professional Services

| | |
|---|---|
| Reviewer for the Journal of Supercomputing | *2024* |
| Reviewer for IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems | *2023-2024* |
| Artifact Evaluation Committee for ACM SIGPLAN Conference on Programming Language Design and Implementation | *2023* |
| External Reviewer for International Conference on Architectural Support for Programming Languages and Operating Systems | *2022* |
| Artifact Evaluation Committee for Journal of Systems Research | *2021* |
| Artifact Evaluation Committee for Symposium on Operating Systems Principles | *2021* |
| Reviewer for IEEE Transactions on Parallel and Distributed Systems | *2018* |

## Technical Skills

Programming Language: C, C++, Verilog, System Verilog

Computer Architecture: PCIe, Performance Modeling & Projection

OS & Virtualization: Linux Kernel, Device Drivers, KVM, QEMU, Mediated Pass-Through

Compiler & Program Analysis: LLVM, Klee, Yosys, Pyverilog

Last edit: Apr. 15, 2025