

RUI GU

rui.gu3@gmail.com
425-623-0088

Experience	Microservices, Service Mesh, Cloud Infrastructure, Distributed System, Concurrency, Operating System, Compiler aided Program Analysis
Skills	<ul style="list-style-type: none"> • Go, C++, Python • Kubernetes, Istio, Docker, GCP, AWS • Microservices Architecture, LLVM
Work Experience	<p><i>2019.10-present</i> Software Engineer, GOOGLE, Sunnyvale, CA</p> <p>Working on Anthos Service Mesh, a dedicated infrastructure layer for making service-to-service communication safe, fast, and reliable.</p> <p><i>2018.10-2019.9</i> Software Engineer, ISOVALENT, Sunnyvale, CA</p> <p>Worked on Isovalent Platform, a microservice-based SaaS offering provides management and visibility of Kubernetes networking using Cilium(cilium.io). Collaboratively design and implement features including network flow extraction and visualization, network policy management and continuous compliance for workloads.</p> <p><i>2017.7-2018.10</i> Software Engineer, VMWARE, Bellevue, WA</p> <p>Worked on VMware Kubernetes Engine(VKE), a fully managed Kubernetes as a service on the public cloud. Design and implement features related to cluster life-cycle management, cluster resource monitoring. Also worked on Photon OS, an open-sourced Linux distribution optimized to run on modern cloud platforms.</p>
Education	<p>M.S.(PhD Dropout), Columbia University Computer Science, May, 2017 Research Topics: Software Reliability, Distributed System, Cloud Infrastructure, Concurrency Advisor: Junfeng YANG</p> <p>B.S., Purdue University Computer Science, May, 2013</p>
Research Publications	<p>[1] Hao Zhou, Ming Chen, Qian Lin, Yong Wang, Xiaobin She, Sifan Liu, Rui Gu, Beng Chin Ooi, Junfeng Yang. "Overload Control for Scaling WeChat Microservices". In <i>Proceedings of the ACM Symposium on Cloud Computing 2018</i>, SoCC'18, Oct 2018</p> <p>[2] Rui Gu, Shixiong Zhao, Haoran Qiu, Tsz On Li, Yuexuan Wang, Heming Cui, and Junfeng Yang. "OWL: Understanding and Detecting Concurrency Attacks". In <i>Proceedings of the 48th IEEE/IFIP International Conference on Dependable Systems and Networks DSN'18</i>, Jun 2018</p> <p>[3] Heming Cui, Rui Gu, Cheng Liu, Tianyu Chen and Junfeng Yang. "Paxos Made Transparent". In <i>Proceedings of the ACM Symposium on Operating Systems Principles</i>, SOSP'15, Oct 2015</p> <p>[4] Rui Gu, Guoliang Jin, Linhai Song, Linjie Zhu and Shan Lu. "What Change History Tells Us About Thread Synchronization". In <i>Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering</i>, FSE'15, Sep 2015</p> <p>[5] Heming Cui, Rui Gu, Cheng Liu and Junfeng Yang. "RepFrame: An Efficient and Transparent Framework for Dynamic Program Analysis". In <i>Proceedings of the ACM SIGOPS Asia-Pacific Workshop on Systems</i>, APSys 2015, July 2015</p> <p>[6] Wei-Chiu Chuang, Bo Sang, Sunghwan Yoo, Rui Gu, Charles Killian and Milind Kulkarni. "Programming Model and Runtime Support for Tightly-Coupled Elastic Cloud Applications". In <i>Proceedings of the ACM Symposium on Cloud Computing</i>, SoCC'13, Oct 2013</p>
Research Projects	<p><i>2015.1-2015.7</i> CRANE: TRANSPARENT STATE MACHINE REPLICATION SYSTEM</p> <p>Collaboratively design and implement Crane[2], a state machine replication runtime system that provides transparent replication capability to general server applications. The system includes</p>

a distributed input consensus(paxos) sidecar server and a deterministic multithreading runtime library.

2015.7–2016.11 OWL: CONCURRENCY VULNERABILITY DETECTOR

Design and implement Owl[1], a program analysis tool that detects concurrency bug related vulnerabilities. Owl first detects and verifies data races using dynamic program analysis. Then it adopts static program analysis(LLVM) to rank the severeness of the real races based on their vulnerability features.

2014.4–2014.8 REPOSITORY MINER

Design and implement a repository miner[3] to automatically analyze and categorize the evolution of synchronization related code of large open source software.

July 26, 2020