RUI GU

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

Experience

Cloud Infrastructure, Distributed System, Concurrency, Operating System, Compiler Infrastructure

Skills

- C++, Python, Go
- Docker, Kubernetes, Microservices Infrastructure
- LLVM

Work Experience

2017.7–Present Software Engineer, VMware Inc — Seattle

Working on **Cascade**, a cloud service controller providing Kubernetes Integration. Previously working on **Photon OS**, a minimal Linux container host, optimized to run on VMware cloud platforms.

2016.2–2016.5 Research Intern, Tencent Inc — Guangzhou

Comprehensive assessment on robustness, scalability and efficiency of WeChat's **microservice traffic control platform**.

Education

M.S., Columbia University
Computer Science, May, 2017

B.S., Purdue University-West Lafayette
Computer Science, May, 2013

Research Publications

[1] Heming Cui, **Rui Gu**, Cheng Liu, Tianyu Chen and Junfeng Yang. ”Paxos Made Transparent”. *In Proceedings of the ACM Symposium on Operating Systems Principles*, **SOSP'15**, Oct 2015

[2] **Rui Gu**, Bo Gan, Jason Zhao, Ning Yi, Heming Cui and Junfeng Yang. ”OWL: Understanding and Detecting Concurrency Attacks”. **Under Review**

[3] **Rui Gu**, Guoliang Jin, Linhai Song, Linjie Zhu and Shan Lu. ”What Change History Tells Us About Thread Synchronization”. *In Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering,* **FSE'15**, Sep 2015

Research Projects

2015.1–2015.7 Crane: Transparent State Machine Replication System

Collaboratively design and implement Crane [1], a state machine replication system that provides transparent replication service to general server applications. The system includes a distributed input consensus(paxos) sidecar server and a deterministic multithreading library.

2015.7–2016.11 Owl: Concurrency Vulnerability Detector

Design and implement Owl [2], a program analysis system 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 components of large open-source software.

January 23, 2018