Luke **Hsiao**

Backend Development · Systems & Networking · Consulting

Remote · No visa sponsorship required to work in the US

cv@luke.hsiao.dev · luke.hsiao.dev · lukehsiao · lukehsiao · Luke Hsiao

Skills

Programming	Python, Rust, C
Areas	backend, systems, tools, open source, networking, infrastructure, cloud

Industry experience

Member of Technical Staff — *Salt Lake City, UT*
Maritime Stealth Startup — *2026-02–present*

Staff Software Engineer — *Salt Lake City, UT*
Alation — *2025-05–2026-01*

- Built the technology stack and foundations of Alation's AI platform.
- Maintained technical standards: security, reliability, scalability.
- Defined and refined team processes (testing, documentation, tooling, etc.) for efficiency.
- Mentored other engineers to accelerate career-growth and skill development.

Principal Engineer — *Salt Lake City, UT*
Numbers Station — *2021-11–2025-05*

- Helped build company from day 1 through acquisition by Alation.
- Led infrastructure: CI/CD, IaC, k8s, security, compliance (SOC-2), VPC deploys, bare-metal servers.
- Designed and implemented fundamental system architecture and engineering processes.
- Created and maintained internal developer tooling and technical documentation.

Software Engineer — *Sunnyvale, CA*
Google — *2021-06–2021-11*

- Led the TCP rebase effort for Project Icebreaker to help Google move towards the mainline Linux kernel.

Software Engineer Intern — *Sunnyvale, CA*
Google — *2020-06–2020-09*

- Added support for TCP tx zerocopy (tx0cp) using `io_uring` in the Linux kernel.
- Profiled and optimized benchmarks to demonstrate an 18% improvement in CPU efficiency for tx0cp via `io_uring`.

Research Intern — *New York City, NY*
Google — *2019-06–2019-09*

- Tuned BBRv2 on many-to-one datacenter traffic; experiments showed >30% latency and >80% retransmit reductions.
- Open-sourced Transperf, a transport protocol performance tool for testing TCP over emulated network scenarios.

Software Engineering Intern — *Santa Clara, CA*
NVIDIA — *2017-06–2017-09*

- Worked with the drivers team to develop a new system-level Windows driver for gaming laptops.
- Designed and implemented secure APIs in kernel-space C code.

Education

2015–2021	**Ph.D. in Electrical Engineering**, Stanford University	*Stanford, CA*
2015–2017	**M.S. in Electrical Engineering**, Stanford University	*Stanford, CA*
2010–2015	**B.S. in Computer Engineering**, Brigham Young University · *Summa Cum Laude*	*Provo, UT*

Research experience

Ph.D. Research Assistant *Stanford, CA*

Stanford University, *Advisors: Phil Levis and Keith Winstein* *2015-09–2021-06*

- Area: Systems and Networking
- Saved 80% network bandwidth by lowering latency to <15 ms for foveated video compression.
- Generated hardware component knowledge bases using training data generation and multitask learning.

Undergraduate Research Assistant *Provo, UT*

Brigham Young University, *Advisor: Mike Wirthlin* *2014-04–2015-06*

- Area: Embedded Systems, FPGA Reliability, Fault Injection
- Assisted in validation and development of Xilinx V5QV fault injection infrastructure.
- Designed and optimized VHDL components for use in FPGA reliability experiments.
- Developed standalone JTAG fault injection system for radiation testing using C/C++.

Teaching experience

W2019 **Graduate CA**, Introduction to Computer Networking (CS 144), Stanford University *Stanford, CA*

W2016 **Graduate Grader**, Program Analysis and Optimizations (CS 243), Stanford University *Stanford, CA*

W2014 **Undergraduate TA**, Data Structures and Algorithms (CS 235), Brigham Young University *Provo, UT*

Publications

2022 **Towards Retina-Quality VR Video Streaming: 15ms Could Save You 80% of Your Bandwidth** *ACM CCR*
L. Hsiao, B. Krajancich, P. Levis, G. Wetzstein, and K. Winstein
📄 cs.stanford.edu/ keithw/sigcomm-ccr-paper523.pdf · </> github.com/lukehsiao/fvideo

2020 **Creating Hardware Component Knowledge Bases with Training Data Generation and Multi-task Learning** *ACM TECS*
L. Hsiao, S. Wu, N. Chiang, C. Ré, and P. Levis
📄 sing.stanford.edu/site/assets/publications/tecs20hack.pdf · </> github.com/lukehsiao/tecs-hardware-kbc

2019 **Automating the Generation of Hardware Component Knowledge Bases** *LCTES*
L. Hsiao, S. Wu, N. Chiang, C. Ré, and P. Levis
📄 sing.stanford.edu/site/assets/publications/hack-lctes19.pdf · </> github.com/lukehsiao/lctes-p27

2018 **Smart Contracts for Machine-to-Machine Communication: Possibilities and Limitations** *IOTAIS*
Y. Hanada, **L. Hsiao**, and P. Levis
📄 arxiv.org/abs/1806.00555

2018 **Fonduer: Knowledge Base Construction from Richly Formatted Data** *SIGMOD*
S. Wu, **L. Hsiao**, X. Cheng, B. Hancock, T. Rekatsinas, P. Levis, and C. Ré
📄 sing.stanford.edu/site/assets/publications/fonduer-sigmod18.pdf · </> github.com/HazyResearch/fonduer

2015 **Estimating Soft Processor Soft Error Sensitivity through Fault Injection** *FCCM*
N. Harward, M. Gardiner, **L. Hsiao**, and M. Wirthlin
📄 ieeexplore.ieee.org/document/7160058

2014 **A Fault Injection System for Measuring Soft Processor Design Sensitivity on Virtex-5 FPGAs** *FASA*
N. Harward, M. Gardiner, **L. Hsiao**, and M. Wirthlin
📄 link.springer.com/chapter/10.1007%2F978-3-319-14352-1_5