Lei Yan

Swiss permit B: Formation avec activité
l.yan@epfl.ch • Github • Linkedin
(Created using *pandoc_resume*)

I am a PhD student at EPFL, working with George Candea, Sanidhya Kashyap, and Diyu Zhou on systems and concurrency. My thesis involves designing systems that can automatically parallelize single-threaded systems software while achieving high and predictable performance. Previously, I built a threading library with fast user-level context switch for the Astriflash architecture and worked on optimizing pointer-intensive in-memory database operations (e.g., hash index lookup) using coroutines. I also did an ML project during my internship at Oracle Labs, which involved clustering security anomalies for fast analysis.

I am seeking software engineer/researcher positions in systems/infrastructure, starting from June 2025 or later. I like hacking and building systems, as well as developing tools that enhance developer and operator productivity.

Skills

- Hacking: C/C++, concurrent programming, multicore systems, operating systems, Bash, Python, network functions, DPDK, VPP, networking, asynchronous programming (coroutines), computer architecture, Linux Kernel, Scrum, database systems, distributed systems, x86/Arm assembly, ML, Computer graphics.
- Human Languages: English (fluent), Chinese (fluent), German (mid-level)

Experience

Sep 2019 - May 2025

PhD Candidate; EPFL (Lausanne, Switzerland)

Doing research in systems, concurrency, and networking

Collaborators: Diyu Zhou, Yueyang Pan
Advisors: George Candea, Sanidhya Kashyap

- Built NFOS, a system with three components: (1) a programming model where one writes single-threaded network functions (e.g., load balancer), (2) a runtime that automatically scales network functions to multicore using tailored software transactional memory and concurrent data structures that leverage, e.g., operation commutativity, and (3) a profiler that identifies root causes of scalability bottlenecks and suggests fixes.
- Built a new NFOS runtime that enables performance clarity of network functions, i.e., we can predict how performance scales with CPU cores for all possible workloads and configurations. This simplifies resource provisioning and performance debugging.
- Taught courses on advanced (operating) systems topics, e.g., microkernel.
- Built the threading library of Astriflash, an architecture that uses flash as main memory for cost saving. The Astriflash architecture triggers a hardware exception upon a DRAM-cache miss. The threading library handles the exception and does a fast user-level thread switch to schedule other threads. This hides the long latency of flash access and enables Astriflash to achieve throughput similar to DRAM-based systems for server workloads.

Nov 2018 - May 2019

Research Intern; Oracle Labs (Zurich, Switzerland)

Adaptive clustering of security anomalies for fast analysis

Advisors: Matteo Casserini, Milos Vasic, Felix Schmidt

Built a tool that uses hierarchical clustering to extract the hierarchy of potential security anomalies found by a ML tool. Each node in the hierarchy is a cluster of anomalies and is split into more homogeneous clusters; for example, a cluster of "failed ssh auth" can be split into clusters of "failed ssh passwd auth" and "failed ssh key auth". This tool makes the life of security operators easier as they only need to look at one anomaly from each cluster. Moreover, the tool is adaptive by providing the hierarchy of anomalies; the security operators can walk the hierarchy until the clusters look homogeneous enough to them.

May 2018 - Oct 2018 **Research Assistant**; Parallel Systems Architecture Lab (PARSA), EPFL (Lausanne, Switzerland)

DSL for exploiting memory-level parallelism of in-memory database operations

Collaborator: Fengyun Liu
Advisors: Dmitrii Ustiugov, Babak Falsafi

Pointer-intensive operations of large in-memory databases (e.g., hash index lookup) have low IPCs since they involve sequences of dependent loads from main memory. We built a domain-specific language (DSL) that automatically turns the operations into coroutines and runs them asynchronously to hide memory load latency, similar to what warp scheduling achieves in GPUs. This achieves a similar speedup to AMAC [Kocberber, VLDB 15] while requiring minimal code changes to the operations.

Publications

[Ongoing work] Transparent Multicore Scaling of Single-Threaded Software with Performance Clarity

[EuroSys 2024] Transparent Multicore Scaling of Single-Threaded Network Functions
***Lei Yan**, Yueyang Pan, Diyu Zhou, George Candea, Sanidhya Kashyap*

[HPCA 2023] AstriFlash: A Flash-Based System for Online Services
*Siddharth Gupta, Yunho Oh, **Lei Yan**, Mark Sutherland, Abhishek Bhattacharjee, Babak Falsafi, and Peter Hsu*

Education

2019-2025 **PhD, Computer Science**; EPFL (Lausanne, Switzerland)

2017-2018 **Exchange master study, Computer Science**; EPFL (Lausanne, Switzerland); 5.7/6

2015-2019 **MSc, Electrical Engineering, Information Technology and Computer Engineering**; RWTH Aachen University (Aachen, Germany); 1.2/1

Major in Computer Engineering

2011-2015 **BEng, Electronic Science & Technology**; Zhejiang University (HangZhou, China); 3.8/4

Honors

- EPFL EDIC Fellowship (2019-2020)
- Scholarship of RWTH Aachen Education Fund (2016-2017)
- RWTH Dean's List (2015-2016)
- Samsung Scholarship, Zhejiang University (2012-2013)