

Automatically optimizing BPF programs using program synthesis

Friday, 24 September 2021 07:50 (40 minutes)

This talk will present K2, an optimizing compiler that uses program synthesis to automatically produce both safe, compact, more performant BPF bytecode. K2 compresses BPF bytecode by 6-26%, improves throughput by 0-4.75%, and reduces average latency by 1.36-55.03%, across benchmarks from Cilium, Facebook Katran, hXDP, and the Linux kernel. We designed several domain-specific techniques to make synthesis practical by accelerating equivalence-checking of BPF programs by 6 orders of magnitude.

The talk will consist of the following parts:

1. A discussion of the challenges in designing optimizing compilers for BPF
2. A description of why and how to utilize program synthesis to find performant BPF bytecode which can pass the kernel checker
3. Techniques for fast equivalence and safety checking
4. Optimizations discovered by K2 for realistic benchmarks
5. Limitation of K2 and future work
6. A discussion of how we think K2 might benefit the community, seeking feedback to improve, more benchmarks, and opportunities to work together

You may find more information including K2's source code, the full technical paper on K2, and responses to some FAQs at <https://k2.cs.rutgers.edu>

I agree to abide by the anti-harassment policy

I agree

Primary authors: XU, Qiongwen (Rutgers University); WONG, Michael (Princeton University); WAGLE, Tanvi (Rutgers University); NARAYANA, Srinivas (Rutgers University); SIVARAMAN, Anirudh (New York University)

Presenters: XU, Qiongwen (Rutgers University); WONG, Michael (Princeton University); WAGLE, Tanvi (Rutgers University); NARAYANA, Srinivas (Rutgers University); SIVARAMAN, Anirudh (New York University)

Session Classification: BPF & Networking Summit

Track Classification: Networking & BPF Summit (Closed)