Contribution ID: **166** Type: **not specified**

Efficient JIT to 32-bit architectures through data flow analysis

Thursday, November 15, 2018 11:00 AM (20 minutes)

eBPF has 64-bit general purpose registers, therefore 32-bit architectures normally need to use register pair to model them and need to generate extra instructions to manipulate the high 32-bit in the pair. Some of these overheads incurred could be eliminated if JIT compiler knows only the low 32-bit of a register is interested. This could be known through data flow (DF) analysis techniques. Either the classic iterative DF analysis or "path-sensitive" version based on verifier's code path walker.

In this talk, implementations for both versions of DF analyser will be presented. We will see how a def-use chain based classic eBPF DF analyser looks first, and will see the possibility to integrate it with previous proposed eBPF control flow graph framework to make a stand-alone eBPF global DF analyser which could potentially serve as a library. Then, another "path-sensitive" DF analyser based on the existing verifier code path walker will be presented. We will discuss how function calls, path prune, path switch affect the implementation. Finally, we will summarize pros and cons for each, and will see how could each of them be adapted to 64-bit and 32-bit architecture back-ends.

Also, eBPF has 32-bit sub-register and ALU32 instructions associated, enable them (-mattr=+alu32) in LLVM code-gen could let the generated eBPF sequences carry more 32-bit information which could potentially easy flow analyser. This will be briefly discussed in the talk as well.

Presenter: WANG, Jiong (Netronome)

Session Classification: BPF MC