



Contribution ID: 158

Type: **not specified**

XDP 1.5 Years In Production. Evolution and Lessons Learned.

Wednesday, 14 November 2018 14:35 (35 minutes)

Today every packet which is reaching Facebook's network is being processed by XDP enabled application. We have been using it for more than 1.5 years and this talk is about evolution of XDP and BPF which has been driven by our production needs. I'm going to talk about history of changes in core BPF components as well as will show why and how it was done. What performance improvements did we get (with synthetics and real world data) and how it was implemented. Also I'm going to talk about issues and shortcomings of BPF/XDP which we have found during our operations, as well as some gotchas and corner cases. In the end we are going to discuss on what is still missing and which part could be improved.

Topics and areas of existing BPF/XDP infrastructure which are going to be covered in this talk:

- why helpers such as `bpf_adjust_head/bpf_adjust_tail` has been added
- unittesting and microbenchmarking with `bpf_prog_test_run`: how to add test coverage of you BPF program and track the regression (we are going to cover how spectre affected BPF kernel infrastructure and what tweaks has been made to get some performance back)
- how map-in-map helps us to scale and make sure that we don't waste memory
- NUMA aware allocation for BPF maps
- inline lookups for BPF arrays/map-in-map

Lessons which we have learned during operation of XDP:

- BPF instruction counts vs complexity
- How to attach more than one XDP program to the interface
- when LLVM and verifier are not the same: some tricks to force LLVM to generate proper BPF
- we will briefly discuss HW limitation: NIC's bandwidth vs packet per second performance

Missing parts: what and why could be added:

- the need for hardware checksumming offload
- bounded loops: what they would allow us to do

I agree to abide by the anti-harassment policy

Presenter: SHIROKOV, Nikita V. (Facebook)

Session Classification: Networking Track