

Adding features to perf using BPF

Tuesday, 21 September 2021 08:00 (45 minutes)

The availability of BPF allows the improvement of preexisting perf features or the addition of new ones without requiring kernel changes.

The first use of BPF to augment perf is to use BPF programs to profile other BPF programs with ‘perf stat’, this is already upstream and set the stage for further uses. This provides functionality similar to ‘bpftool prog profile’ while reusing lots of ‘perf stat’ features that were developed and improved by the perf tooling community.

Then we had bperf, to share hardware performance counters, aggregate data in BPF maps that then get read by ‘perf stat’ as if it was a normal perf event that then reuses all the perf tooling features.

Some improvements, such as scaling cgroups perf monitoring were first attempted by modifying the kernel. But after several attempts, one is being made using BPF with encouraging results. It works by hooking into cgroup scheduling and doing aggregation that is made available to ‘perf stat’ via bperf.

Such use of BPF for aggregating information in the kernel instead of changing the perf subsystem was well received by a perf kernel maintainer, which is encouraging.

Future work will use BPF to enable perf_events when some specific trigger condition takes place, so that only a window determined by two probes gets sampled.

Also being considered is the conversion of some perf subcommands that analyze tracepoints like perf sched/lock/etc to use BPF to aggregate things in the kernel instead of passing vast amounts of data for aggregation in userspace while keeping the existing, familiar tooling interface.

This shows how the perf and BPF communities are working together to improve Linux tooling, provide ways to scale profiling and to improve observability of BPF programs, it is expected that by presenting this talk we get suggestions for further improvements.

I agree to abide by the anti-harassment policy

I agree

Primary authors: CARVALHO DE MELO, Arnaldo (Red Hat Inc.); LIU, Song

Co-author: KIM, Namhyung

Presenters: CARVALHO DE MELO, Arnaldo (Red Hat Inc.); LIU, Song; KIM, Namhyung

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track (Closed)