Contribution ID: **281** Type: **not specified**

bpf tracing multi-link supporting

For now, tracing-type BPF programs and BPF trampolines adopt a per-function design, requiring the creation of independent instances for each kernel function to be traced. This leads to significant inefficiencies in large-scale tracing scenarios (e.g., monitoring hundreds or thousands of kernel functions): not only do redundant instances consume substantial additional memory, but the program attachment process also takes an excessively long time, severely limiting overall efficiency. The tracing multi-link feature draws inspiration from the design of kprobe multi-link, enabling efficient binding of a single BPF program to multiple target functions and fundamentally addressing this critical pain point.

The BPF global trampoline is introduced. Unlike bpf trampolines, which hardcode program addresses into instructions (relying on direct calls and thus resulting in per-function binding), the global trampoline constructs a function metadata hash table using rhashtable. This table centrally stores key information such as BPF programs, cookies, and the number of function parameters. When triggered, it queries the hash table in real time based on the target function address to retrieve the corresponding metadata and invokes the associated program. The entire implementation is built using naked functions and C code, greatly simplifying the logic. Practical tests show that this feature reduces the loading time in large-scale tracing scenarios from tens of minutes to within 1 second (even when tracing 50k kernel functions), delivering a significant performance improvement. For detailed information, please refer to: https://lore.kernel.org/bpf/20250703121521.1874196-1-dongml2@chinatelecom.cn/

Primary author: Mr DONG, Menglong (ChinaTelecom)

Presenter: Mr DONG, Menglong (ChinaTelecom)

Session Classification: eBPF Track

Track Classification: eBPF Track