Tracepoints that Allow Faults

Mathieu Desnoyers
Michael Jeanson
EfficiOS Inc.
Problem and Goals

- Reliably capture data from user-space memory when tracing system call entry and exit,
- Similar to `strace`, but without the overhead associated with scheduling threads from another process and `ptrace` peek,
- The major issue is handling page faults from tracer callbacks because it requires code to be sleepable and to take the `mmap` semaphore:
  - Typically happens when system call input arguments are located in the binary data segment immediately after an `exec()` or a `dlopen()`,
  - Can also happen on a system under high memory pressure.
- Tracepoints allow in-kernel tracers to hook on system call entry/exit,
- Tracepoints disable preemption around the entire callback invocation,
- Prevents kernel tracers from handling page faults,
- eBPF and LTTng attached to tracepoints allow reading user-space data pointed to by system call arguments, but use a zero-padding strategy when a page fault would be required,
- eBPF since 5.10 supports sleepable programs, but those cannot currently attach to tracepoints.
Proposed Solution

- Extend Tracepoints and TRACE_EVENT APIs to allow defining a faultable tracepoint which invokes its callbacks with preemption enabled:
  - TRACEPOINT_MAYFAULT flag.
- Extend Tracepoints probe registration APIs to allow registering a callback which is meant to be invoked with preemption enabled:
  - tracepoint_probe_register_mayfault().
- Use Task Trace RCU to synchronize read-side marshalling of the registered probes with respect to faultable probes unregistration and teardown.
References

- “Relief for insomniac tracepoints”, Linux Weekly News
  - https://lwn.net/Articles/835426/
- “Sleepable BPF programs”, Linux Weekly News
  - https://lwn.net/Articles/825415/
- “[RFC PATCH] Faultable tracepoints (v2)”
  - https://lore.kernel.org/lkml/20210218222125.46565-1-mjeanson@efficios.com/
- “[PATCH] rcu-tasks: Add an RCU Tasks Trace to simplify protection of tracing hooks”
  - https://lore.kernel.org/lkml/20200415181941.11653-15-paulmck@kernel.org/