Strace.eBPF

Fast syscall's tracing

Vitalii Chernookyi

Why we need new tool

- regular system tracing tools are slow
- regular tools slowdown traced application for few orders
- output of regular tools is human-oriented and don't assume automated processing
- overcoming above problems in regular way require:
  - kernel hacking (sysdig)
  - special HW (Lauterbach).

Used technologies

- eBPF
- KProbe
- Perf Event Circular Buffer
- event-loop

System requirements

- libbcc
- Linux Kernel 4.4 (for Perf Event Circular Buffer)
- CAP_SYS_ADMIN capability for bpf() syscall
- mounted tracefs

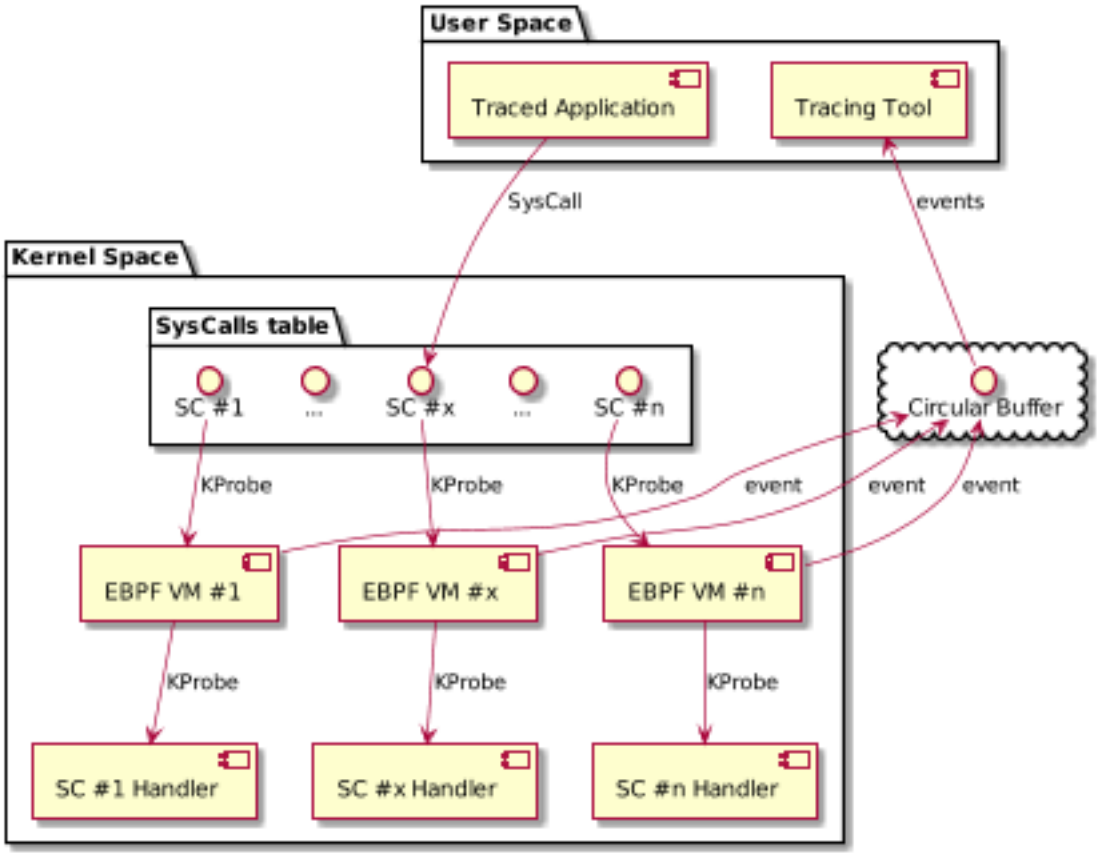
Pros

- Used combination of technologies allow tool to be about one order faster than regular system strace.
- This tool consume much less amount of CPU resource
- Output of this tool is designed to be suiteable for processing with classical tools and technologies, like awk.
- Could trace syscalls system-wide.

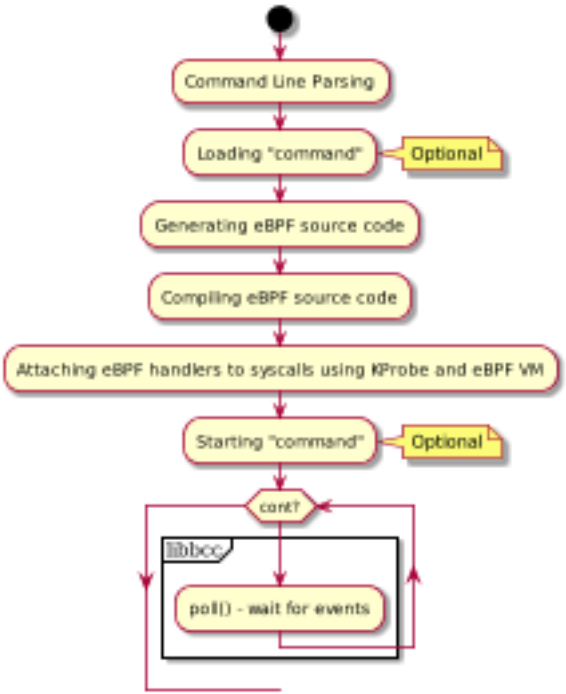
Cons

- Limited functionality
- Slow attaching and detaching
- Asyncronity. If user will not provide enough system resources for performace tool will skip some calls. Tool does not assume to try any work-around behind the scene.

Structural Component Diagram



Behavioral Activity Diagram



Conclusion

- we reached performance about 1000000 syscalls per second.
- there is places for future optimization.