Kernel Tracing Tools

字节跳动系统部STE团队 - 宋牧春

ByteDance字节跳动

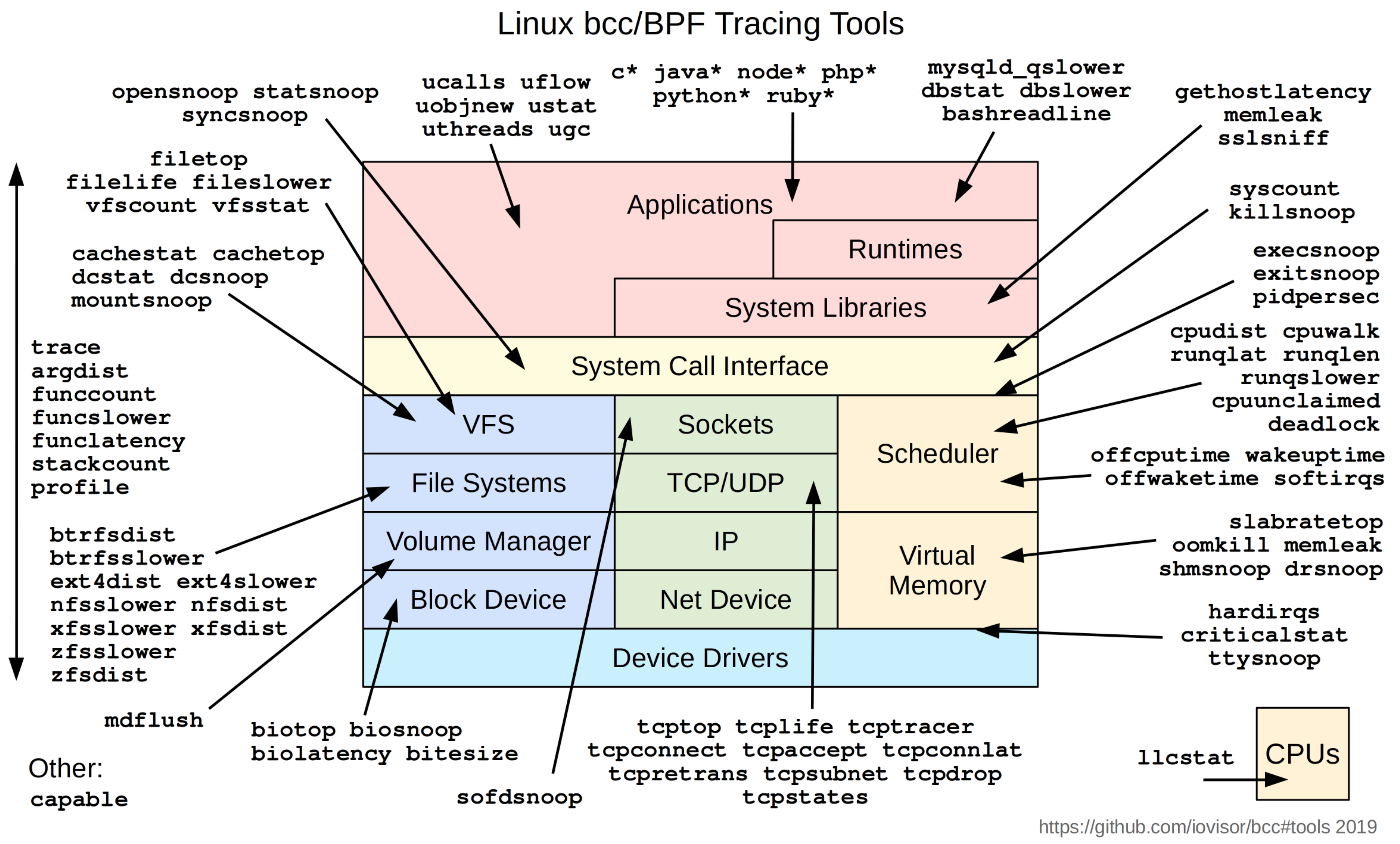
Agenda

- Lightweight and simple kernel memory leak detector
- Hard/soft irqs off latency tracing
- Non-scheduled thread in kernel space tracing

Lightweight and simple kernel memory leak detector

Which tools can troubleshooting memory leak

- [memleak.py](#) which in the BCC tools
- [kmemleak](#) which in the kernel development tools


Linux bcc/BPF Tracing Tools
opensnoop statsnoop syncsnoop
ucalls uflow uobjnew ustat uthreads ugc
c* java* node* php* python* ruby*
mysqld_qslower dbstat dbslower bashreadline
gethostlatency memleak sslsniff
filetop filelife fileslower vfscount vfsstat
cachestat cachetop dcstat dcsnoop mountsnoop
trace argdist funccount funcslower funclatency stackcount profile
btrfsdist btrfsslower ext4dist ext4slower nfsslower nfsdist xfsslower xfsdist zfsslower zfsdist
mdflush
syscount killsnoop
execsnoop exitsnoop pidpersec
cpudist cpuwalk runqlat runqlen runqslower cpuunclaimed deadlock
offcputime wakeuptime offwaketime softirqs
slabratetop oomkill memleak shmsnoop drsnoop
hardirqs criticalstat ttysnoop
Applications
Runtimes
System Libraries
System Call Interface
VFS
Sockets
Scheduler
File Systems
TCP/UDP
Volume Manager
IP
Virtual Memory
Block Device
Net Device
Device Drivers
biotop biosnoop biolatency bitesize
sofdsnoop
tcptop tcplife tcptracer tcpconnect tcpaccept tcpconnlat tcpretrans tcpsubnet tcpdrop tcpstates
llcstat
CPUs
Other: capable
https://github.com/iovisor/bcc#tools 2019

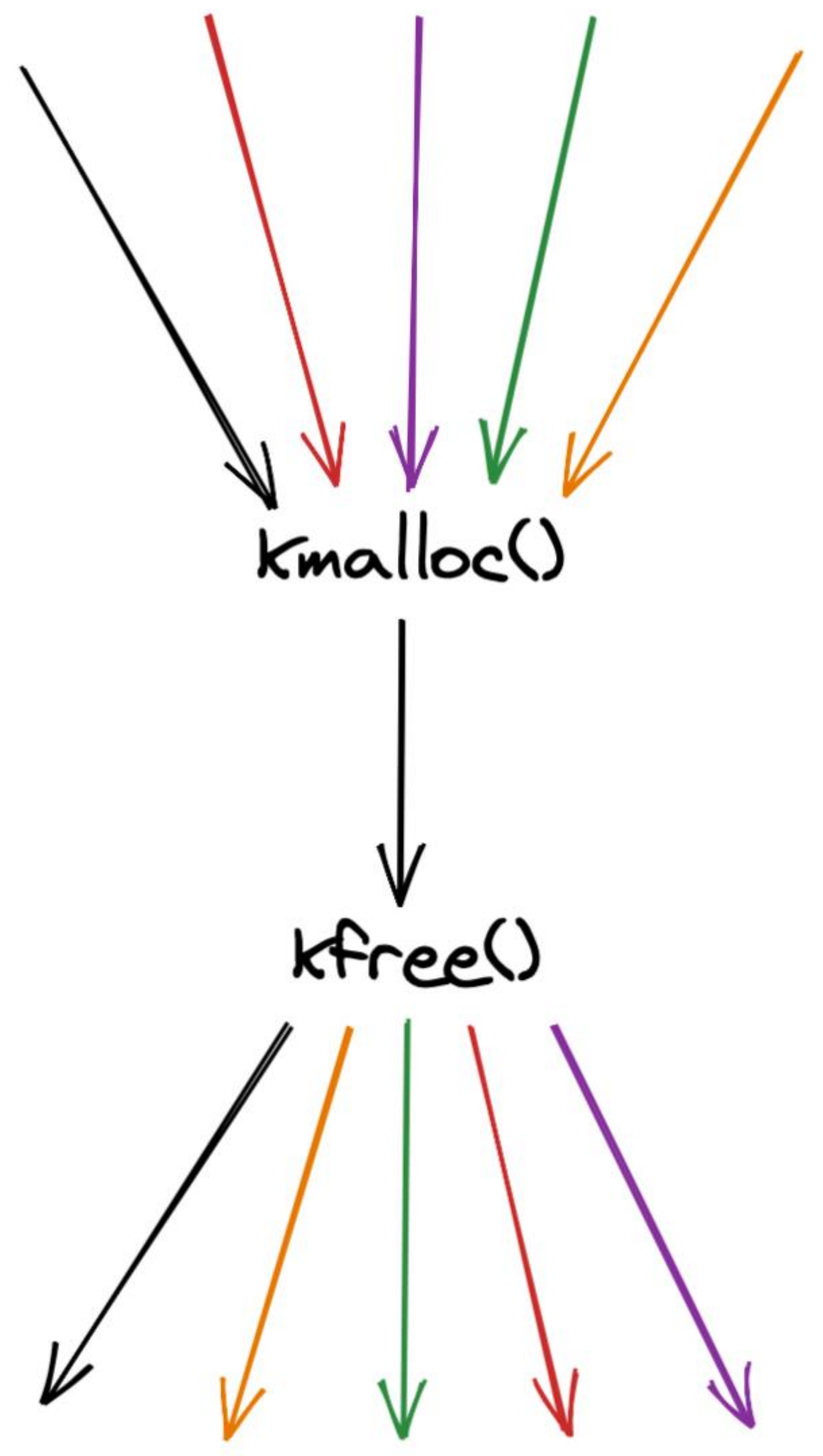
Why do not we choose BCC and kernel memory leak tools

Why do not choose the memleak.py

- Save all memory allocation call traces, increase memory usage.
- Too much noise call traces, disturbing our analysis.

Why do not choose the kmemleak

- Need enable CONFIG_DEBUG_KMEMLEAK and compile the kernel.
- Need reboot the server and install the debug kernel image.
- Waste a lot of memory to maintain the metadata.
- Need to reproduce the issue.



How to troubleshoot memory leaks

The steps to debug memory leak

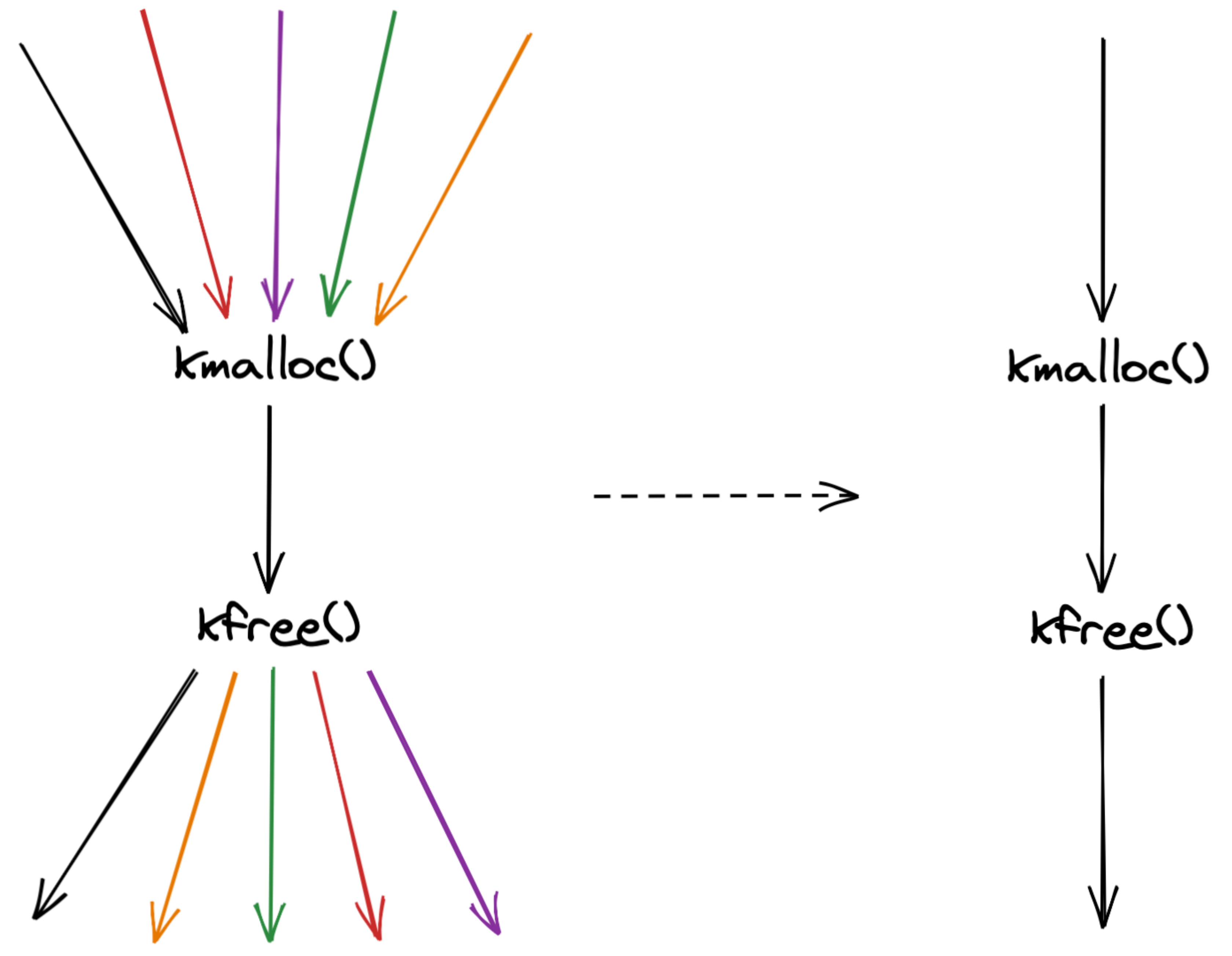
- Use the slabtop to determine which kmem_cache may leak.
- Trace the special kmem_cache memory allocation
- Troubleshoot call trace where memory leaks may occur

The characteristics of memory leak

- Generally speaking, there is only one memory leak in the system.
- Most of the memory allocation will be freed in a short time.

Lightweight and simple kernel memory leak detector

- Only trace one allocated address and save the address and call trace.
- Assume that this address will be freed soon.
- If it is freed for a short time, the next address is randomly selected for tracking.
- If not, assume that this address is leaked and print the call trace periodically.



Lightweight and simple kernel memory leak detector demo

memory alloc

```
tracepoint:kmem:kmalloc,
tracepoint:kmem:kmalloc_node,
tracepoint:kmem:kmem_cache_alloc,
tracepoint:kmem:kmem_cache_alloc_node
{
        if (args->bytes_alloc == 1024 && !@kmem_addr) {
                @alloc_stack = kstack;
                @kmem_addr = args->ptr;
        }
}
```

memory free

```
tracepoint:kmem:kfree,
tracepoint:kmem:kmem_cache_free
{
        if (@kmem_addr && @kmem_addr == args->ptr) {
                delete(@kmem_addr);
                delete(@alloc_stack);
        }
}
```

print leak addr

```
interval:s:100
{
        if (@kmem_addr) {
                printf("kmem_addr: 0x%lx\n", @kmem_addr);
                printf("%s\n", @alloc_stack);
        }
}
```

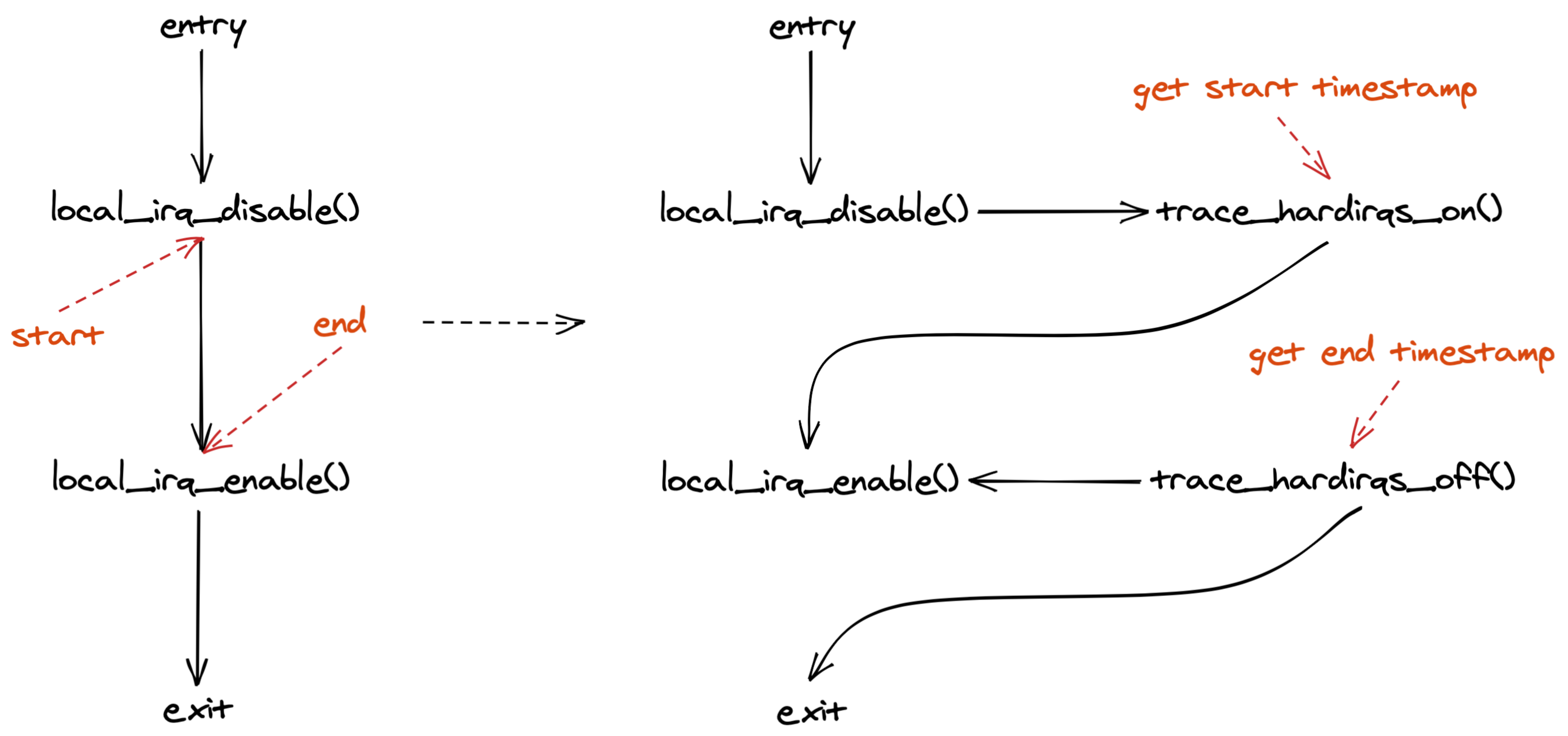
Lightweight and simple kernel memory leak detector demo

Advantage:

- No need to reboot the server.
- The output information is simple and easy to analyze.
- Simple but effective. At present, three memory leaks in the internal kernel of ByteDance have been discovered by this tool.

Hard/soft irqs off latency tracing

How to trace hard/soft irqs off latency



How to trace hard/soft irqs off latency

Advantages:

- Simple but accurate.
- The tool is readily available.

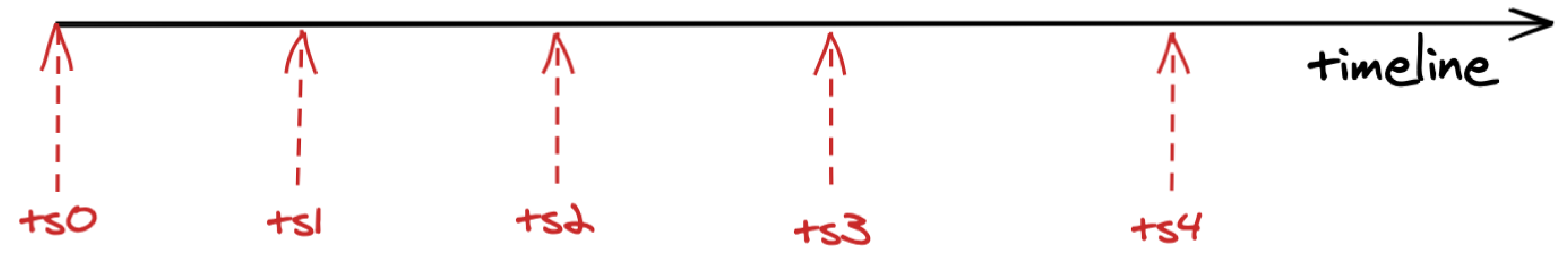
Disadvantages:

- Need enable CONFIG_TRACE_IRQFLAGS and compile the kernel.
- Need reboot the server and install the debug kernel image.
- Need to reproduce the issue.
- The overhead is high.

Hard/soft irqs off latency traceing

Is it necessary to accurately measure the irq off latency?

Hard/soft irqs off latency traceing



- Use a periodically hrtimer to record timestamp.
- If the interval between two timestamps is greater than 2 times the hrtimer period, we think that the hardirqs off latency is hrtimer period.

Hard/soft irqs off latency traceing

Advantages:

- No need to compile kernel, just need insmod.
- No need to reboot server.
- The overhead is low.

Disadvantages:

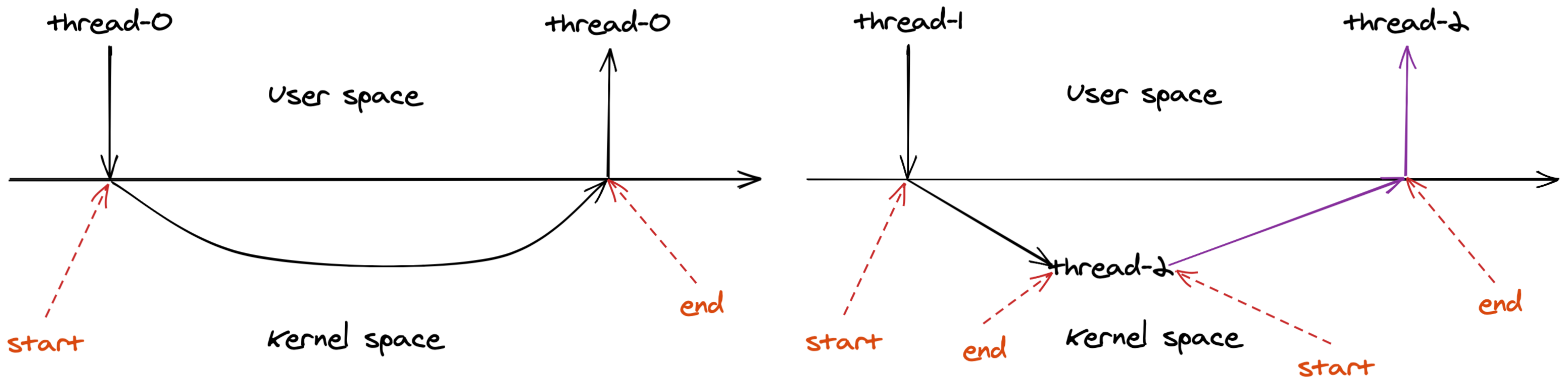
- The accuracy depends on the timer sampling period, so the accuracy is not high. But debugging the issue is enough.

Non-scheduled thread in kernel space tracing

Non-scheduled thread in kernel space tracing

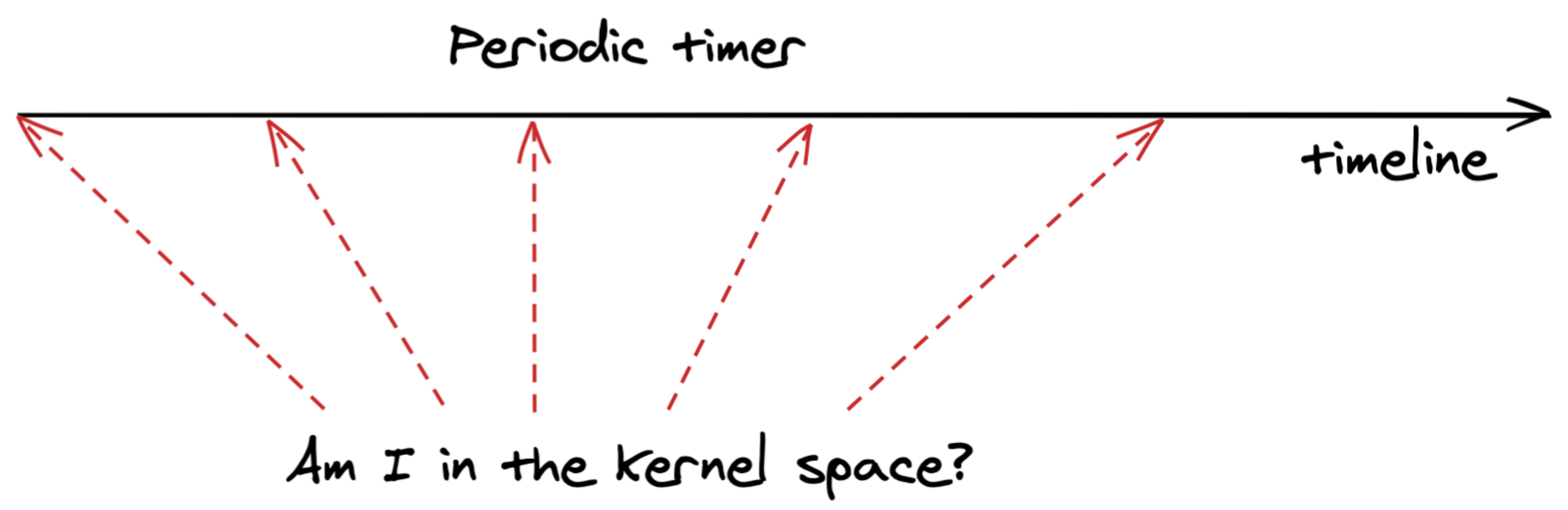
What problems may we face in the non-preemptible kernel?

The time spent in the kernel space

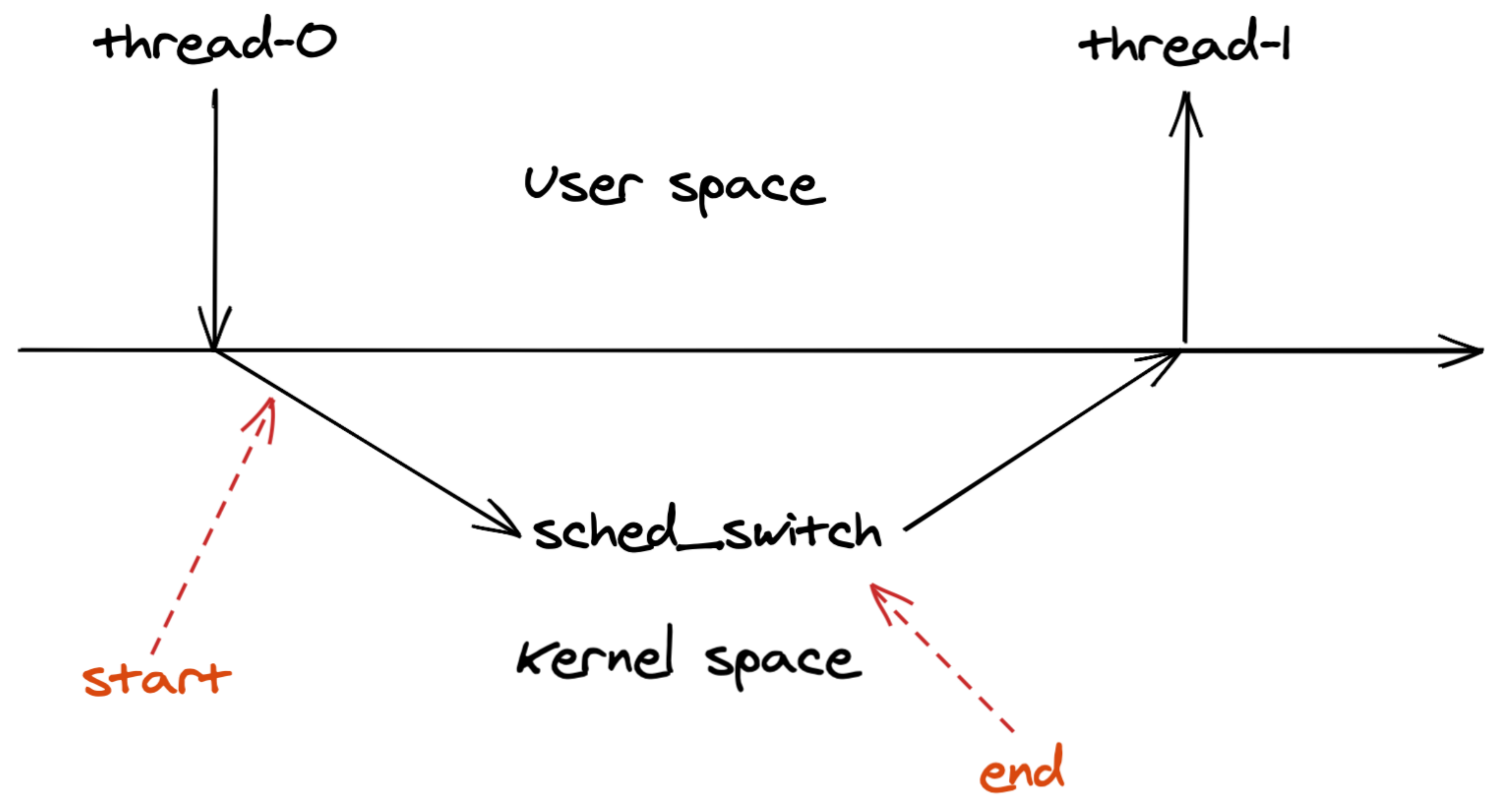


Non-scheduled thread in kernel space tracing

Get the start timestamp(hrtimer)



Get the end timestamp(sched tracepoint)



THANKS

ByteDance 字节跳动

More tools and open source

https://github.com/bytedance/trace-irqoff

https://github.com/bytedance/trace-noschedule

https://github.com/bytedance/trace-runqlat

kernel trace tools讨论

该二维码7天内(10月31日前)有效，重新进入将更新