Can we blame the Kernel instead of Open vSwitch?

OVS/OVN Conference 2023

Eelco Chaudron
Principal Software Engineer

- ❏ Explaining the problem
- ❏ Introduction to the *kernel_delay.py* tool
- ❏ Demo debugging the revalidator
- ❏ Questions?

Explaining the problem

- *ovs-vswitchd* can generate weird log messages:
  - `...|WARN| Unreasonably long 1259ms poll interval (0ms user, 692ms system)`
  - `...|WARN| blocked 1001 ms waiting for handler340 to quiesce`
- Is this *ovs-vswitchd* or an overloaded/misbehaving kernel?
- Specific purpose tools exist
  - Coordinating and utilizing these tools efficiently can be a hurdle.
- Can we have a single tool to rule out the kernel?

Introduction to the *kernel_delay.py* tool

- Python based script
- Uses eBPF hooks/programs to gather information
  - Uses the BCC framework[1]
- Can be triggered on demand
- Will report on the following
  - SYSCALL statistics
  - THREAD RUN statistics
  - THREAD READY statistics
  - HARD IRQ statistics
  - SOFT IRQ statistics
- See blog[2] and documentation[3] for more information

[1] https://github.com/iovisor/bcc
[2] https://developers.redhat.com/articles/2023/07/24/troubleshooting-open-vswitch-kernel-blame
[3] https://github.com/openvswitch/ovs/blob/master/utilities/usdt-scripts/kernel_delay.rst

Run the kernel_delay.py tool

```
$ sudo ./kernel_delay.py
# Start sampling @2023-06-08T12:17:22.725127 (10:17:22 UTC)
# Stop sampling @2023-06-08T12:17:23.224781 (10:17:23 UTC)
# Sample dump @2023-06-08T12:17:23.224855 (10:17:23 UTC)
TID        THREAD           <RESOURCE SPECIFIC>
---------- ---------------- ----------------------------------------------------------------------------
     31741 revalidator122   [SYSCALL STATISTICS]
                            NAME                     NUMBER         COUNT           TOTAL ns            MAX ns
                            poll                          7             5        184,193,176       184,191,520
                            recvmsg                      47           494        125,208,756           310,331
                            ...
                            TOTAL( - poll):                           519        144,405,334

                            [THREAD RUN STATISTICS]
                            SCHED_CNT             TOTAL ns             MIN ns             MAX ns
                                 6       136,764,071              1,480       115,146,424

                            [THREAD READY STATISTICS]
                            SCHED_CNT             TOTAL ns             MAX ns
                                 7            11,334              6,636

                           [HARD IRQ STATISTICS]
                            NAME                           COUNT           TOTAL ns            MAX ns
                            eno8303-rx-1                       1              3,586             3,586
                            TOTAL:                             1              3,586

                            [SOFT IRQ STATISTICS]
                            NAME                     VECT_NR         COUNT           TOTAL ns            MAX ns
                            net_rx                         3             1             17,699            17,699
                            sched                          7             6             13,820             3,226
                            rcu                            9            16             13,586             1,554
                            timer                          1             3             10,259             3,815
                            TOTAL:                                      26             55,364
```

- *kernel_delay.py* has two modes of operation
  - In time mode, the tool runs for a specific time and collects the information.
  - In trigger mode, event collection can be started and/or stopped based on a specific eBPF probe.
    Supported trigger probes:
    - USDT probes
    - Kernel tracepoints
    - kprobe
    - kretprobe
    - uprobe
    - uretprobe

Red Hat

- Additional *sample* options exist:
  - `--sample-count`; How many measurements you would like to perform.
  - `--trigger-delta`; Ignore measurements if the delta is less than configured.
  - `--sample-interval`; Delay the start of a new measurement.

- ❏ Supports start and stop triggers in any combination
  - ❏ Start only example:

```
# ./kernel_delay.py --start-trigger up:bridge_run --sample-time 4 \
                    --sample-count 2 --sample-interval 1
```

  - ❏ Stop only example:

```
# ./kernel_delay.py --stop-trigger upr:bridge_run \
                    --sample-count 4 --sample-interval 1
```

  - ❏ Start and stop example:

```
# ./kernel_delay.py --start-trigger up:bridge_run \
                    --stop-trigger upr:bridge_run \
                    --sample-count 4 --sample-interval 1 \
                    --trigger-delta 50000
```

- Supported trigger probes and syntax[1]
    - USDT probes[2]; `[u]:{provider}:{probe}`
    - Kernel tracepoint; `[t:trace]:{system}:{event}`
    - Kprobe; `[k:kprobe]:{kernel_function}`
    - Kretprobe; `[kr:kretprobe]:{kernel_function}`
    - Uprobe; `[up:uprobe]:{function}`
    - Uretprobe; `[upr:uretprobe]:{function}`

[1] https://github.com/iovisor/bcc/blob/master/docs/reference_guide.md#events--argumentsc
[2] https://github.com/openvswitch/ovs/blob/master/Documentation/topics/usdt-probes.rst

What will it report?

- ❑ Currently has five statistics it’s gathering
  - ❑ SYSCALL statistics
  - ❑ THREAD RUN statistics
  - ❑ THREAD READY statistics
  - ❑ HARD IRQ statistics
  - ❑ SOFT IRQ statistics
- ❑ More can be easily added if needed in the future

- Report all syscalls per thread for the measurement duration
- Per type; count of calls, total duration, and worst case duration
- It only counts syscall that are started AND stopped during the interval

```
217258 ovs-vswitchd       [SYSCALL STATISTICS]
                          NAME                    NUMBER          COUNT            TOTAL ns            MAX ns
                          poll                         7              4       1,494,094,595       500,838,810
                          ioctl                       16             14           7,105,005         3,284,088
                          read                         0             10           1,088,265           225,845
                          accept                      43             15              38,603             9,978
                          socket                      41             14              36,769             6,520
                          openat                     257              5              33,489            16,825
                          sendmsg                     46              1              31,454            31,454
                          recvmsg                     47             26              19,587             7,536
                          futex                      202             12              10,003             3,649
                          close                        3             19               4,885               531
                          readv                       19             10               4,404               854
                          recvfrom                    45              5               3,056             1,215
                          getrusage                   98              5               2,728             1,004
                          TOTAL( - poll):                           136           8,378,248
```

THREAD RUN STATISTICS

- ❏ Report how long the thread was running on a CPU
- ❏ Counts time the thread was scheduled on and off
- ❏ Records total, minimum and maximum CPU time
- ❏ It only counts events that started AND stopped during the interval
- ❏ Note PMD threads might show nothing for this statistic due to the above

```
217258 ovs-vswitchd       [SYSCALL STATISTICS]
                          ...

                          [THREAD RUN STATISTICS]
                          SCHED_CNT            TOTAL ns              MIN ns               MAX ns
                                 230          1,459,544              1,381              155,609
```

THREAD READY STATISTICS

- ❏ Report how long the thread was waiting for CPU time
- ❏ Records total and maximum schedule delay
- ❏ It only counts events that started AND stopped during the interval
- ❏ Note PMD threads might show nothing for this statistic due to the above

```
217258 ovs-vswitchd     [SYSCALL STATISTICS]
                        ...

                        [THREAD RUN STATISTICS]
                        SCHED_CNT           TOTAL ns           MIN ns           MAX ns
                              230        1,459,544            1,381          155,609

                        [THREAD READY STATISTICS]
                        SCHED_CNT           TOTAL ns           MAX ns
                              230           66,745            2,984
```

HARD IRQ STATISTICS

- Report time spent servicing hard interrupts during the threads run time
- Records per irq vector count, total duration, and worst case duration

```
217331 revalidator48     [SYSCALL STATISTICS]
                         ...
                         ...

                         [HARD IRQ STATISTICS]
                         NAME                       COUNT           TOTAL ns            MAX ns
                         eno8303-rx-1                   1              3,586             3,586
                         TOTAL:                         1              3,586
```

- Report time spent servicing soft interrupts during the threads run time
- Records per irq vector count, total duration, and worst case duration

```
217331 revalidator48    [SYSCALL STATISTICS]
                        ...
                        ...

                        [SOFT IRQ STATISTICS]
                        NAME                 VECT_NR         COUNT           TOTAL ns             MAX ns
                        sched                      7             1              2,149              2,149
                        rcu                        9             1                890                890
                        TOTAL:                                   2              3,039
```

- ❑ The *`--syscall-events`* option will report individual syscalls
- ❑ Has an optional argument to only report call taking more than x ns
- ❑ Does support backtraces, but are not that useful[1], can be disabled with *`--stack-trace-size 0`*
- ❑ Skip poll() system calls with *`--skip-syscall-poll-events`*

```
# ./kernel_delay.py  --syscall-events 50000 --skip-syscall-poll-events
...
...
# SYSCALL EVENTS:
       ENTRY (ns)           EXIT (ns)        TID COMM             DELTA (us)  SYSCALL
  ------------------- ------------------- ---------- ---------------- ----------  ----------------
     2161821694935486    2161821695031201    3359699 revalidator14            95   futex
      syscall_exit_to_user_mode_prepare+0x161 [kernel]
      syscall_exit_to_user_mode_prepare+0x161 [kernel]
      syscall_exit_to_user_mode+0x9 [kernel]
      do_syscall_64+0x68 [kernel]
      entry_SYSCALL_64_after_hwframe+0x72 [kernel]
      __GI___lll_lock_wait+0x30 [libc.so.6]
      ovs_mutex_lock_at+0x18 [ovs-vswitchd]
      [unknown] 0x696c003936313a63
     2161821695276882    2161821695333687    3359698 revalidator13            56   futex
      syscall_exit_to_user_mode_prepare+0x161 [kernel]
      ...
      ovs_mutex_lock_at+0x18 [ovs-vswitchd]
      [unknown] 0x696c003134313a63
```

[1] https://github.com/chaudron/perf_scripts/blob/master/analyze_perf_pmd_syscall.py

Demo debugging the revalidator

Questions?

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat