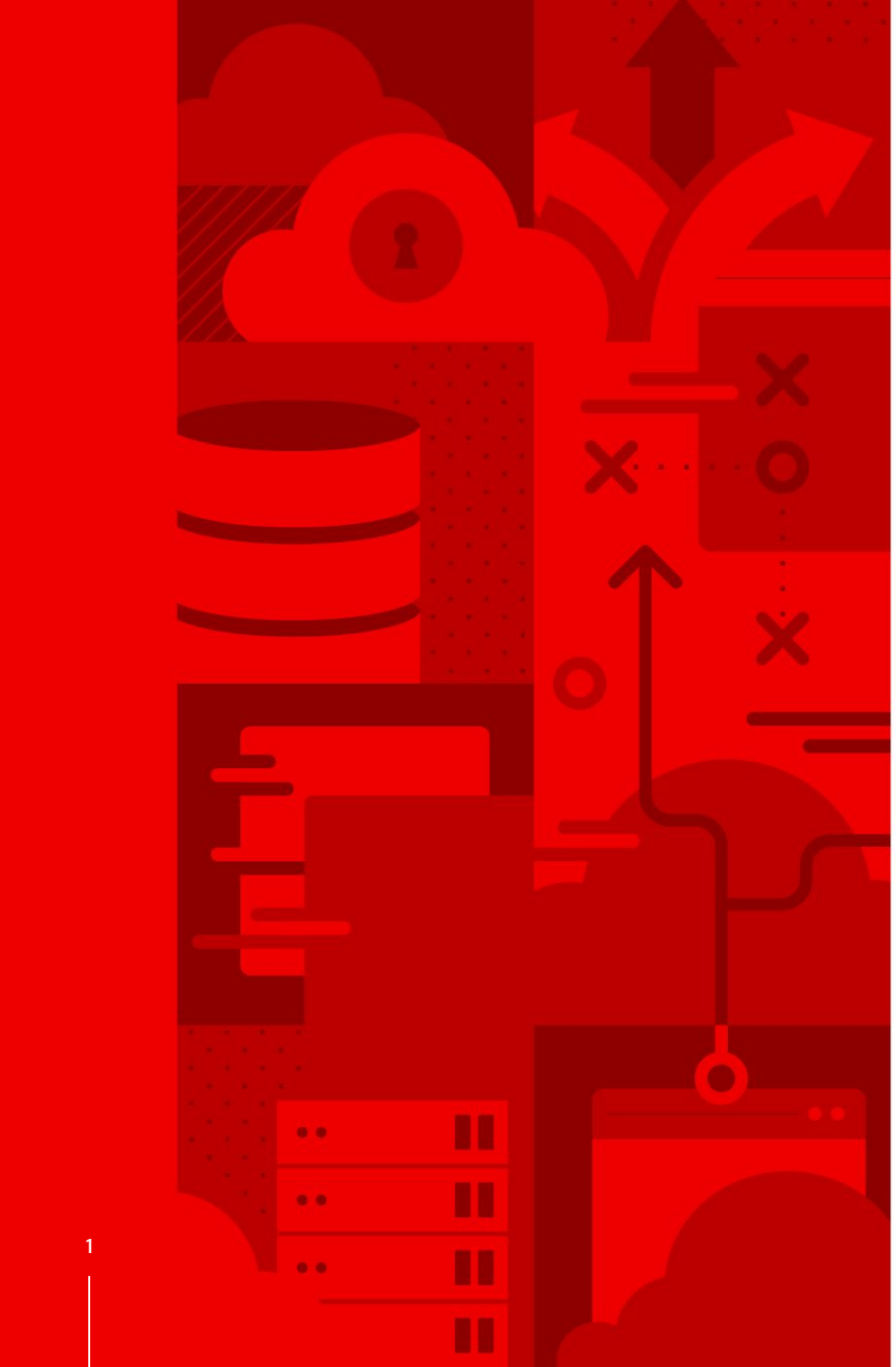


Using bpftrace with Performance Co-Pilot & Grafana

Andreas Gerstmayr
Software Engineer

Red Hat

Performance Co-Pilot

system performance analysis toolkit

Grafana

analytics and monitoring solution

bpftrace

high-level tracing language for eBPF

Source:
https://pcp.io
https://grafana.com
https://bpftrace.org

Performance Co-Pilot

Toolkit for collecting, analyzing, visualizing and responding to the status and performance of servers, applications and networks.

92

Agents

Performance Metrics Domain Agents export performance data from the kernel, services (e.g. PostgreSQL) and other instrumented applications

6000+

Metrics

Metrics have associated metadata:

Semantics: instant, counter, discrete

Unit: Kilobytes, bits/sec, etc.

Type: float, (un)signed int, string, ...

Instances: e.g. sda1, sda2

Grafana

Visualization
Visualization of time series, heatmaps, tabular data, etc.

Customizable
Customizable dashboards, drag & drop, live preview

Multiple data sources
PCP, Elasticsearch, Prometheus, PostgreSQL, ...

Extensible
Custom data sources & panels

bpftrace

tracing language for eBPF

```
// read.bt file
tracepoint:syscalls:sys_enter_read
{
    @start[tid] = nsecs;
}

tracepoint:syscalls:sys_exit_read / @start[tid] /
{
    @times = hist(nsecs - @start[tid]);
    delete(@start[tid]);
}
```

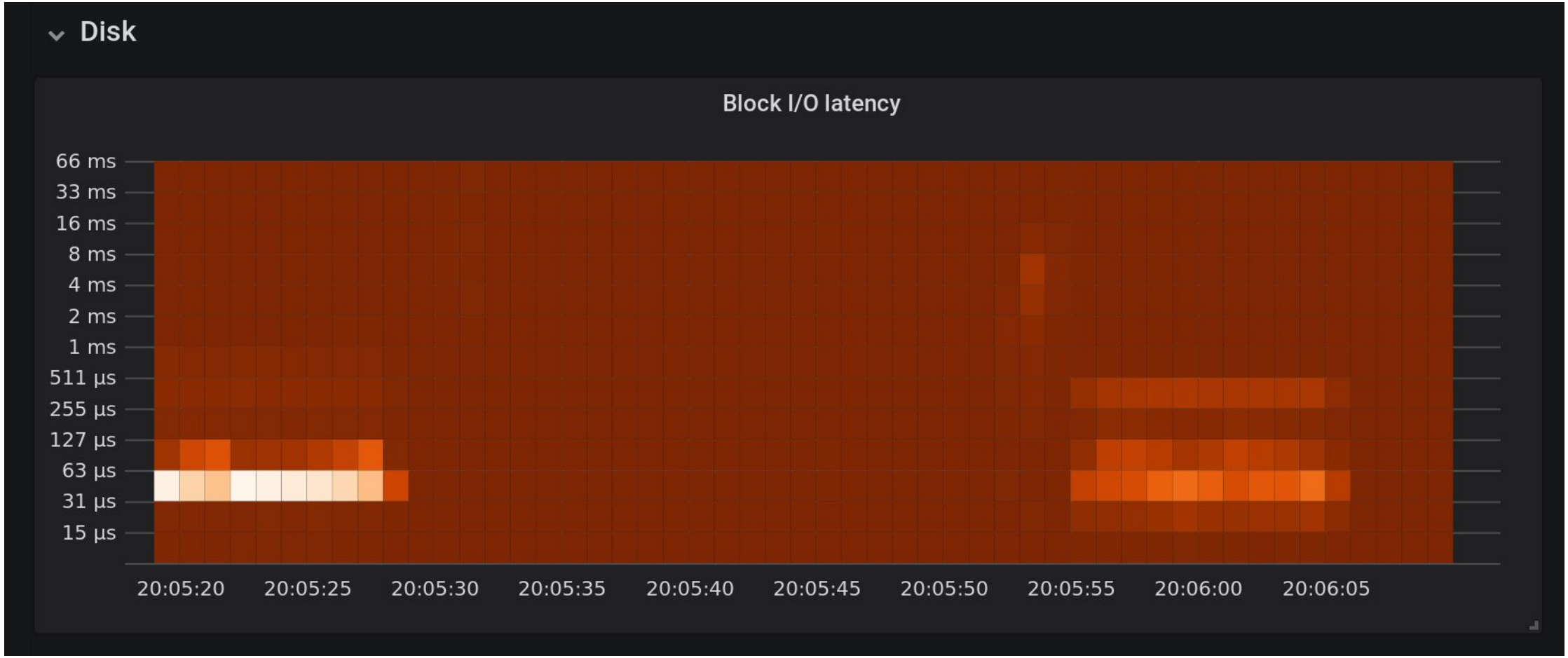

Source:
https://github.com/iovisor/bpftrace/blob/master/README.md

bpftrace

tracing language for eBPF

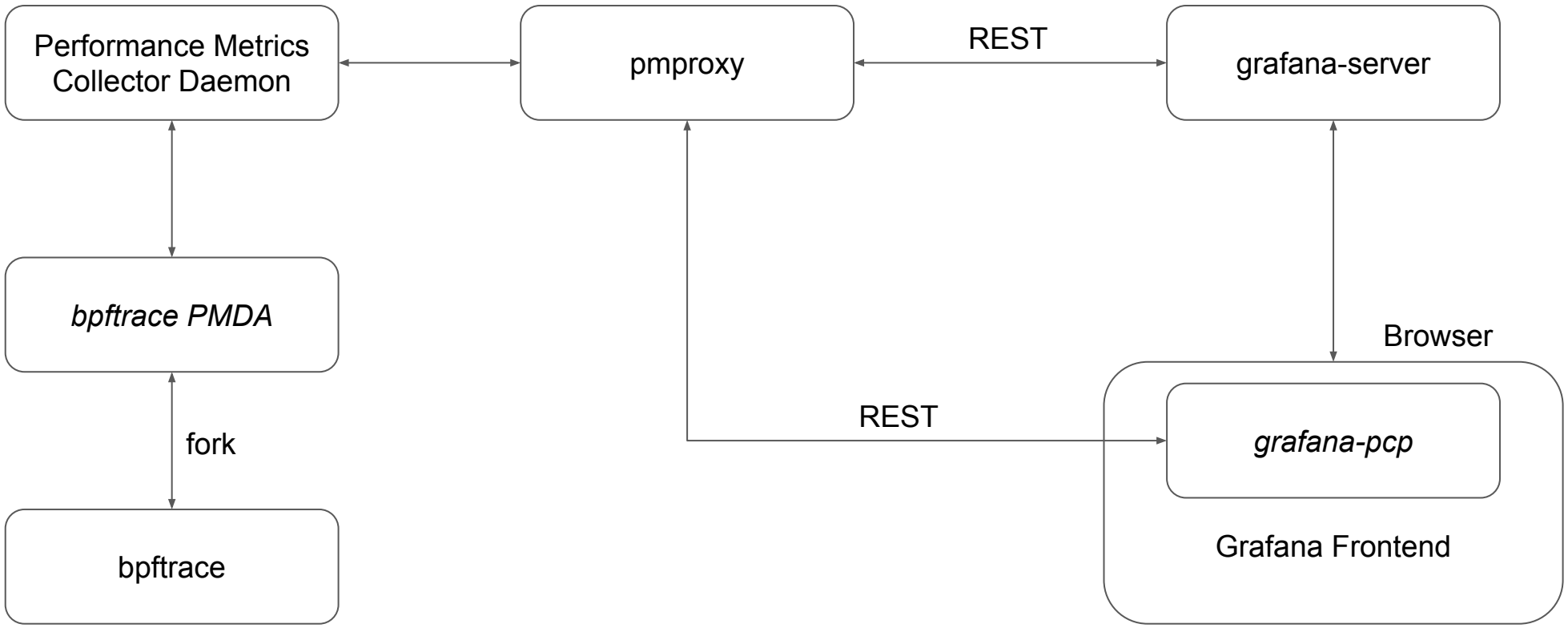
```
$ bpftrace read.bt
Attaching 2 probes...
^C

@times:
[256, 512)             326 |@                                                   |
[512, 1k)             7715 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@|
[1k, 2k)             15306 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@|
[2k, 4k)               609 |@@                                                  |
[4k, 8k)               611 |@@                                                  |
[8k, 16k)              438 |@                                                   |
[16k, 32k)              59 |                                                    |
[32k, 64k)              36 |                                                    |
[64k, 128k)              5 |                                                    |
```

PCP + Grafana + bpftrace

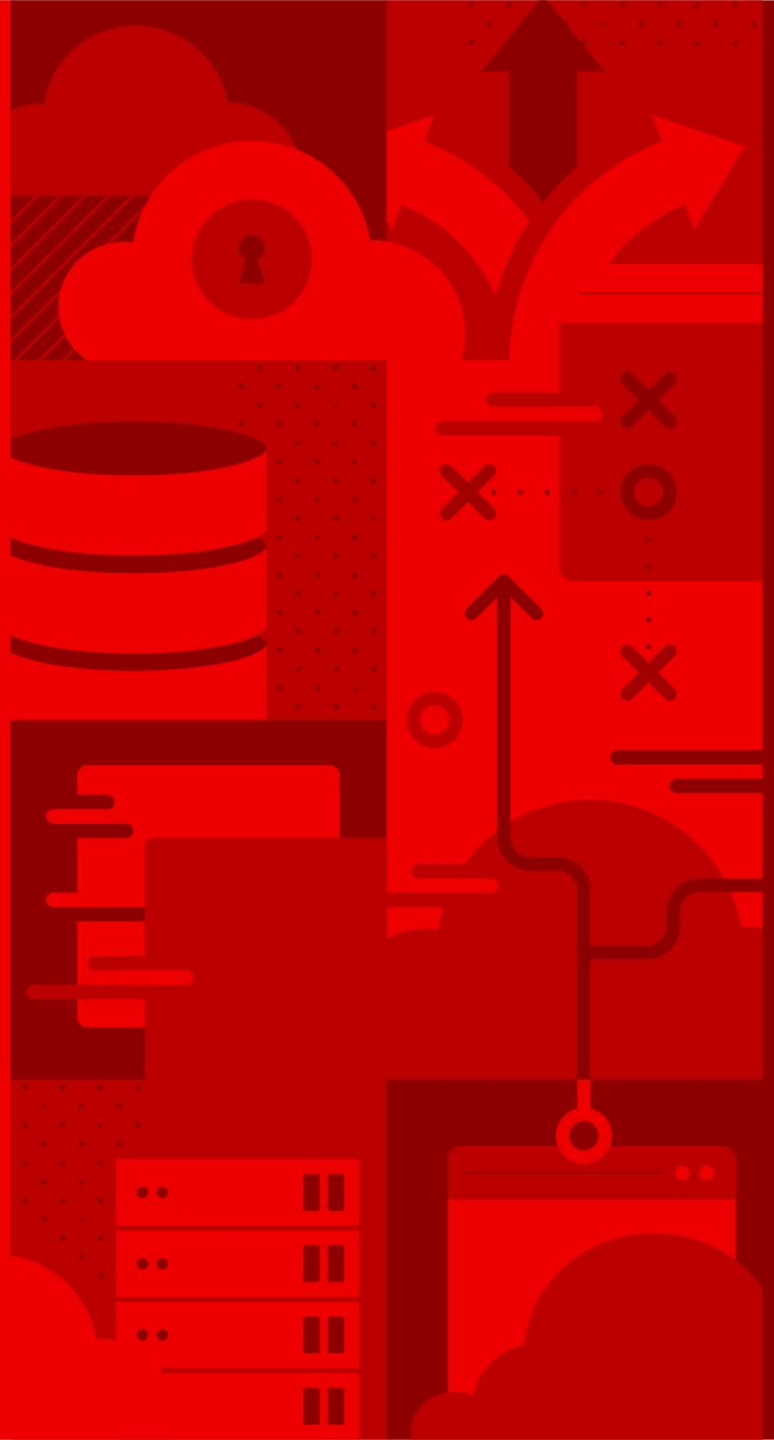


Red Hat

Architecture



Red Hat



Installation

Red Hat

Installation on Fedora 31

```
# bpftrace version with support for kernel 5.4+
$ dnf copr enable agerstmayr/bpftrace

$ sudo dnf install pcp pcp-pmda-bpftrace grafana grafana-pcp bpftrace
$ sudo systemctl start pmproxy grafana-server
```

bpftrace PMDA Configuration

Configure authentication

```
$ sudo vim /var/lib/pcp/pmdas/bpftrace/bpftrace.conf
```

Install/Enable the PMDA

```
$ cd /var/lib/pcp/pmdas/bpftrace
$ sudo ./Install
```

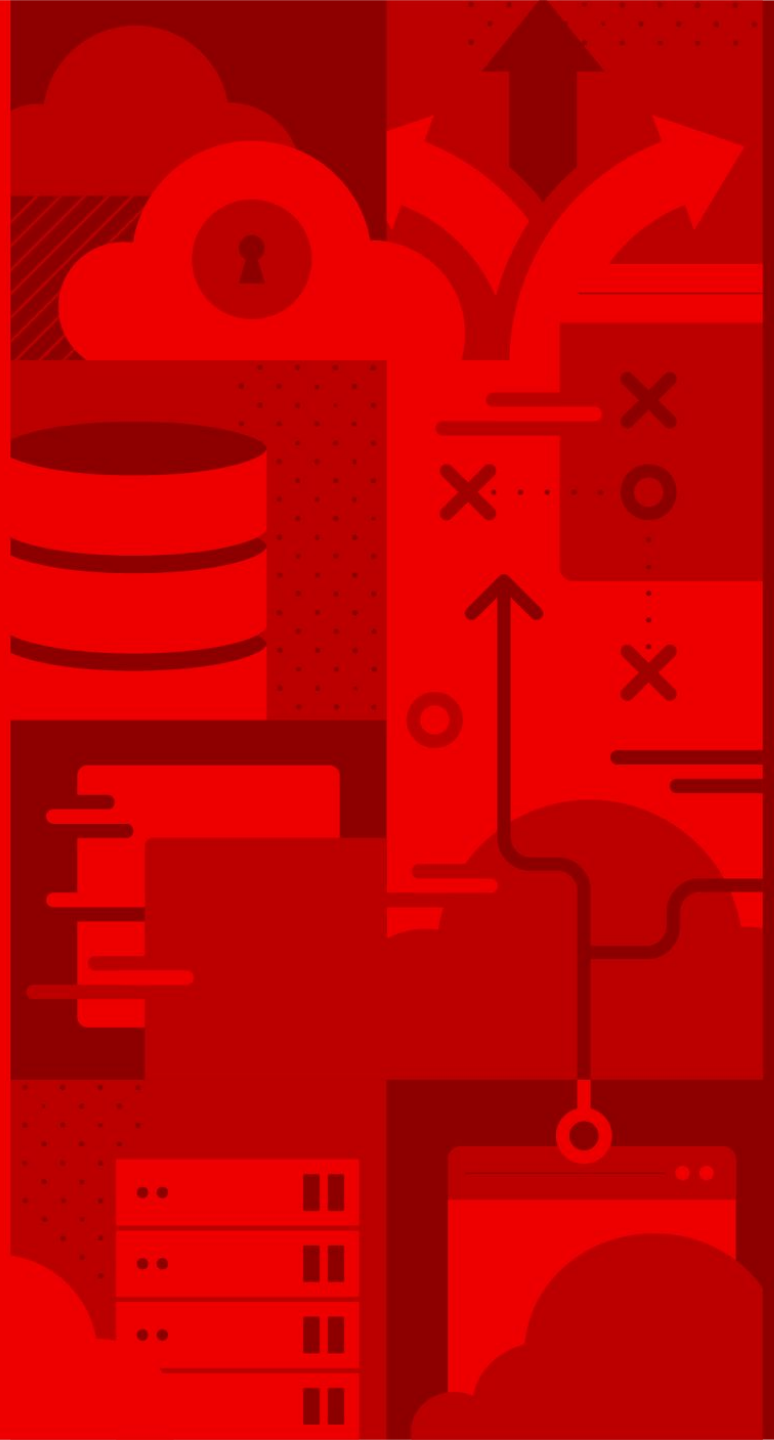
Grafana Configuration

Enable plugin

Go to Configuration / Plugins and enable *Performance Co-Pilot*

Configure data source

Go to Configuration / Data Sources / Add data source, select *PCP bpftrace*, enter URL of pmproxy, click Save

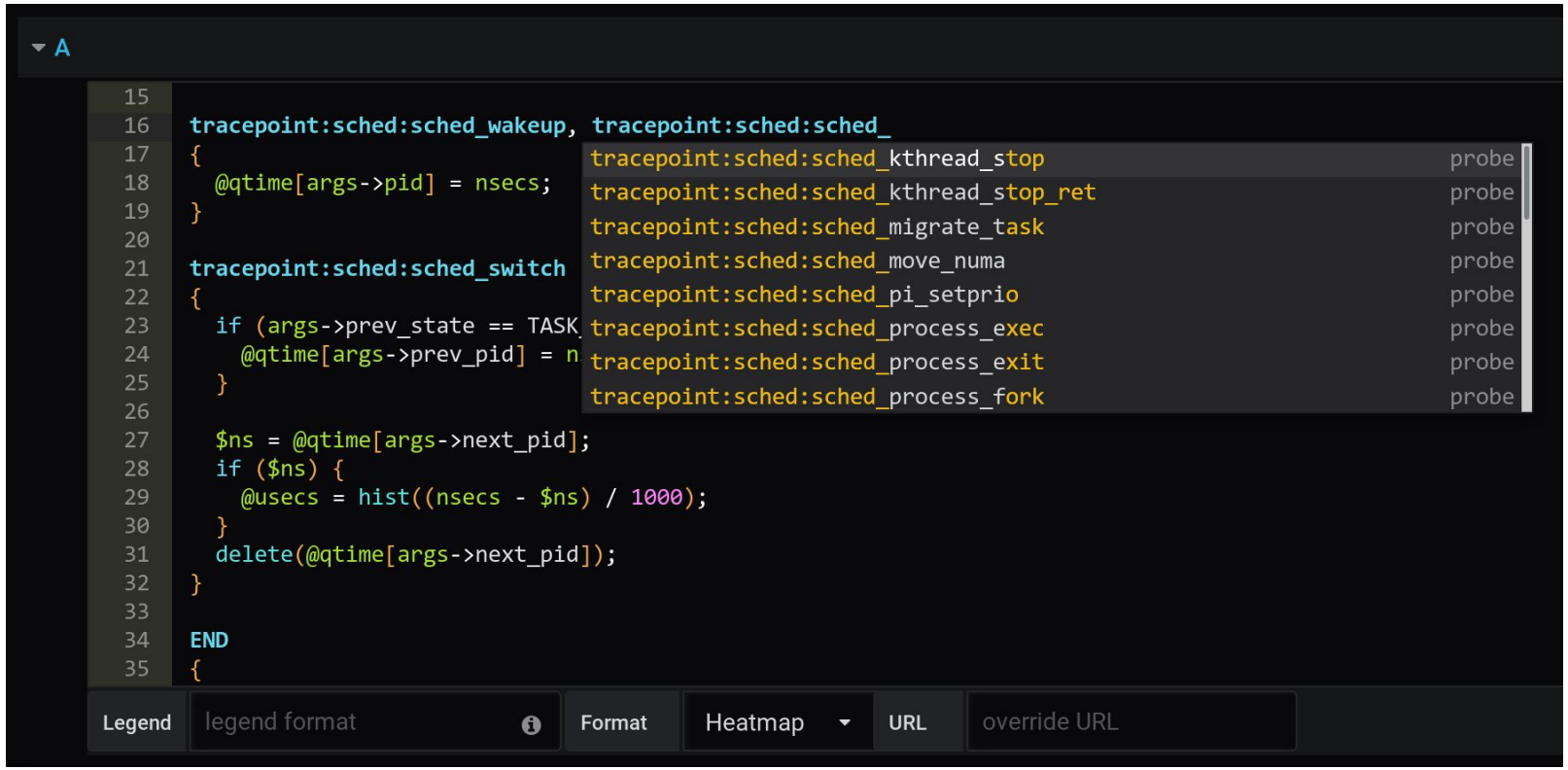


bpftrace tools

Red Hat

runqlat

CPU scheduler run queue latency



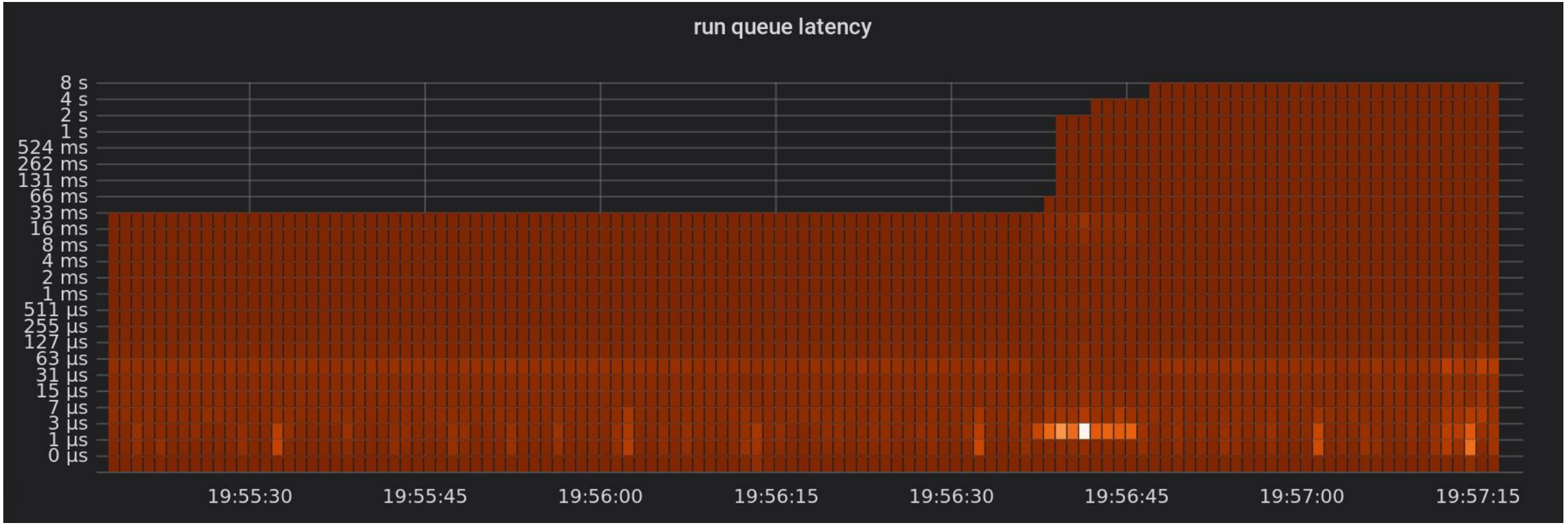
runqlat

```
1  #include <linux/sched.h>
2  tracepoint:sched:sched_wakeup,
3  tracepoint:sched:sched_wakeup_new
4  {
5      @qtime[args->pid] = nsecs;
6  }
7  tracepoint:sched:sched_switch
8  {
9      if (args->prev_state == TASK_RUNNING) {
10         @qtime[args->prev_pid] = nsecs;
11     }
12     $ns = @qtime[args->next_pid];
13     if ($ns) {
14         @usecs = hist((nsecs - $ns) / 1000);
15     }
16     delete(@qtime[args->next_pid]);
17 }
```


Source: https://github.com/iovisor/bpftrace/blob/master/tools/runqlat.bt

runqlat

CPU scheduler run queue latency



Red Hat

tcpconnect

Trace TCP connect()

```
$ sudo /usr/share/bpftrace/tools/tcpconnect.bt
Attaching 2 probes...
Tracing tcp connections. Hit Ctrl-C to end.
TIME     PID    COMM           SADDR  SPORT  DADDR                      DPORT
18:17:38 22576  curl           -      45666  1.1.1.1                    80
18:17:53 22598  curl           -      50966  8.8.4.4                    80
18:20:40 23429  wget           -      40606  2a02:26f0:10e:1b8::d44     443
18:21:15 1746   syncthing      -      44096  2a03:b0c0:0:1010::bb:4001  443
^C
```

tcpconnect

Trace TCP connect()

```
// table-retain-lines: 10

BEGIN
{
    printf("%s,%s,%s,", "TIME", "PID", "COMM");
    printf("%s,%s,%s,%s\n", "SADDR", "SPORT", "DADDR",
"DPORT");
}
```


Adapted from:
https://github.com/iovisor/bpftrace/blob/master/tools/tcpconnect.bt

```
1  kprobe:tcp_connect
2  {
3      $sk = ((struct sock *) arg0);
4      $inet_family = $sk->__sk_common.skc_family;
5
6      if ($inet_family == AF_INET || $inet_family == AF_INET6)
7      {
8          // [...]
9          $lport = $sk->__sk_common.skc_num;
10         $dport = $sk->__sk_common.skc_dport;
11         $dport = ($dport >> 8) | (($dport << 8) & 0x00FF00);
12
13         time("%H:%M:%S,");
14         printf("%d,%s,", pid, comm);
15         printf("%s,%d,%s,%d\n", $saddr, $lport, $daddr, $dport);
16     }
17 }
```


Adapted from:
https://github.com/iovisor/bpftrace/blob/master/tools/tcpconnect.bt

tcpconnect

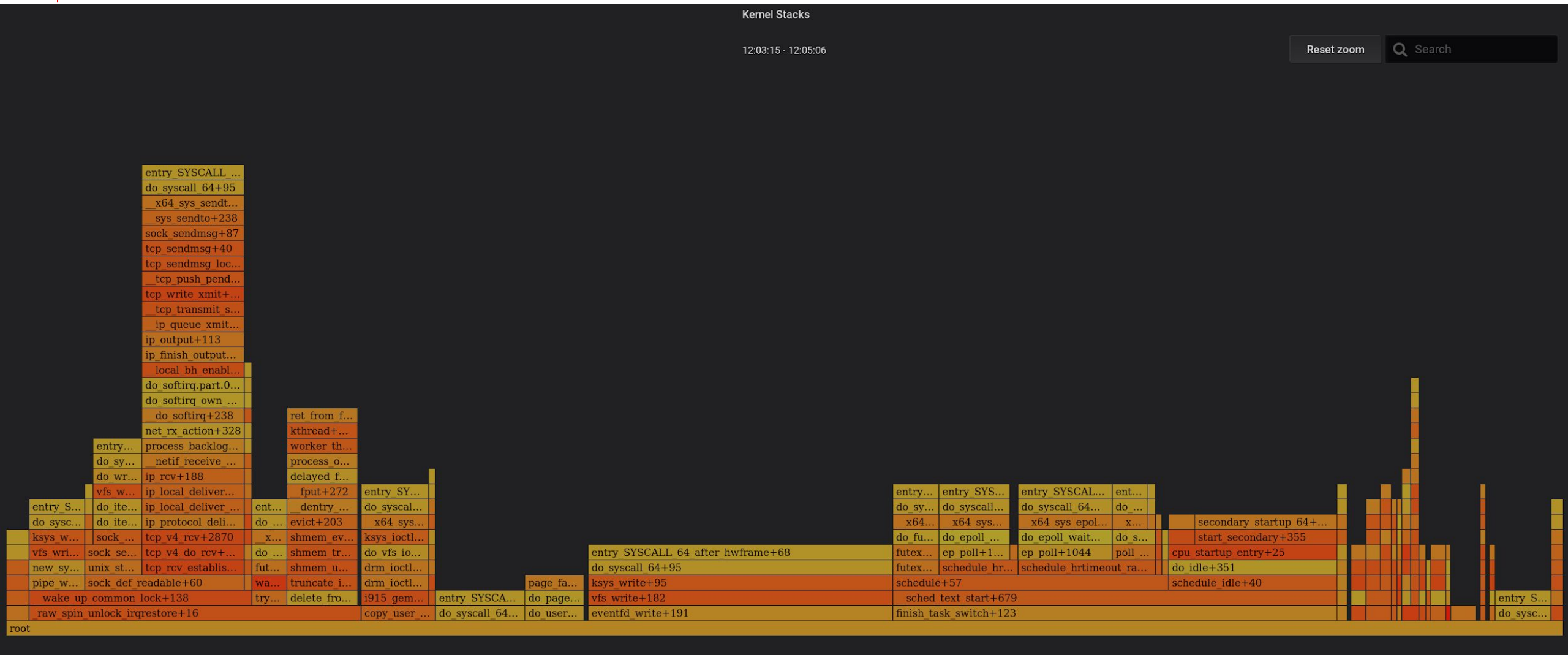
Trace TCP connect()

trace TCP connect()

TIME ▾	PID	COMM	SADDR	SPORT	DADDR	DPORT
18:21:15	1746	syncthing	2a02:8388:xxx:xxxx:xxxx:xxxx:xxxx:xxxx	44096	2a03:b0c0:0:1010::bb:4001	443
18:20:40	23429	wget	2a02:8388:xxx:xxxx:xxxx:xxxx:xxxx:xxxx	40606	2a02:26f0:10e:1b8::d44	443
18:17:53	22598	curl	192.168.0.185	50966	8.8.4.4	80
18:17:38	22576	curl	192.168.0.185	45666	1.1.1.1	80

Sampling kernel stacks

```
profile:hz:99
{
    @stacks[kstack] = count();
}
```


Kernel Stacks
12:03:15 - 12:05:06
Reset zoom
Search
entry_SYSCALL_...
do_syscall_64+95
__x64_sys_sendt...
__sys_sendto+238
sock_sendmsg+87
tcp_sendmsg+40
tcp_sendmsg_loc...
__tcp_push_pend...
tcp_write_xmit+...
__tcp_transmit_s...
__ip_queue_xmit...
ip_output+113
ip_finish_output...
__local_bh_enabl...
do_softirq.part.0...
do_softirq_own_...
__do_softirq+238
net_rx_action+328
process_backlog...
__netif_receive_...
ip_rcv+188
ip_local_deliver...
ip_local_deliver_...
ip_protocol_deli...
tcp_v4_rcv+2870
tcp_v4_do_rcv+...
tcp_rcv_establis...
sock_def_readable+60
__wake_up_common_lock+138
__raw_spin_unlock_irqrestore+16
ret_from_f...
kthread+...
worker_th...
process_o...
delayed_f...
__fput+272
__dentry_...
evict+203
shmem_ev...
shmem_tr...
shmem_u...
truncate_i...
delete_fro...
entry_S...
do_sysc...
ksys_w...
vfs_wri...
new_sy...
pipe_w...
entry...
do_sy...
do_wr...
vfs_w...
do_ite...
do_ite...
sock_...
sock_se...
unix_st...
ent...
do_...
__x...
do_...
fut...
wa...
try...
entry_SY...
do_syscal...
__x64_sys...
ksys_ioctl...
do_vfs_io...
drm_ioctl...
drm_ioctl...
i915_gem...
copy_user_...
page_fa...
entry_SYSCA...
do_syscall_64...
do_page...
do_user...
entry_SYSCALL_64_after_hwframe+68
do_syscall_64+95
ksys_write+95
vfs_write+182
eventfd_write+191
entry...
do_sy...
__x64...
do_fu...
futex...
futex...
schedule+57
__sched_text_start+679
finish_task_switch+123
entry_SYS...
do_syscall...
__x64_sys...
do_epoll_...
ep_poll+1...
schedule_hr...
entry_SYSCAL...
do_syscall_64...
__x64_sys_epol...
do_epoll_wait...
ep_poll+1044
schedule_hrtimeout_ra...
ent...
do_...
__x...
do_s...
poll_...
secondary_startup_64+...
start_secondary+355
cpu_startup_entry+25
do_idle+351
schedule_idle+40
entry_S...
do_sysc...
root

Red Hat

Preconfigured container

```
$ sudo -H podman run -d \
    --privileged \
    -v /lib/modules:/lib/modules:ro \
    -v /usr/src:/usr/src:ro \
    -p 127.0.0.1:3000:3000 \
    quay.io/performancecopilot/pcp-preview
```

Live Demo

github.com/andreasgerstmayr

linkedin.com/in/andreasgerstmayr

Red Hat