facebook

XDP: 1.5 years in production. Evolution and lessons learned.

Nikita V. Shirokov

Facebook Traffic team

Goals of this talk:

Show how bpf infrastructure (maps/helpers) could be used for building networking application and what benefits/ gotchas exist.

Share operational experience of running XDP on large scale

Operational Experience

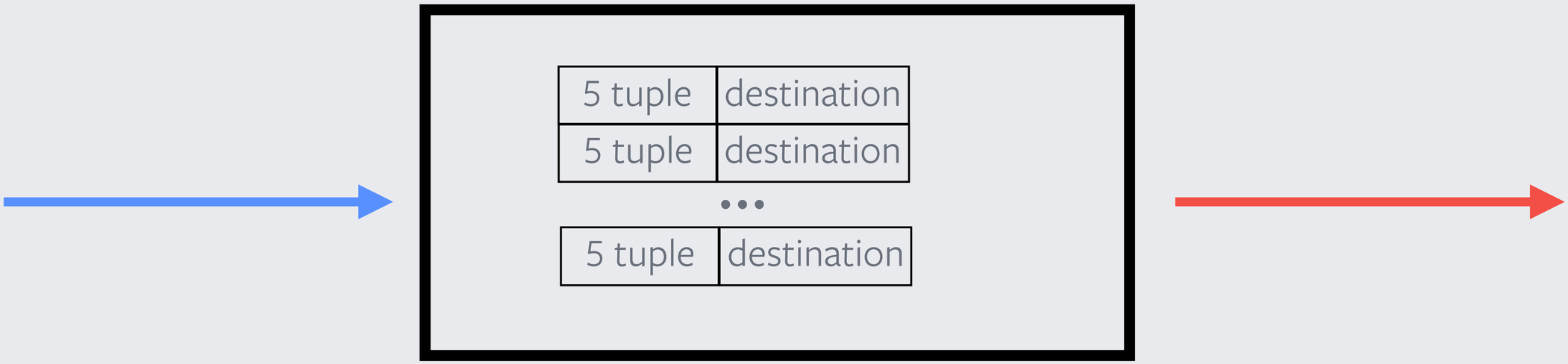
Every packet toward facebook.com has been processed by XDP enabled application since May, 2017

XDP enabled application

L4 load balancer:
https://github.com/facebookincubator/katran

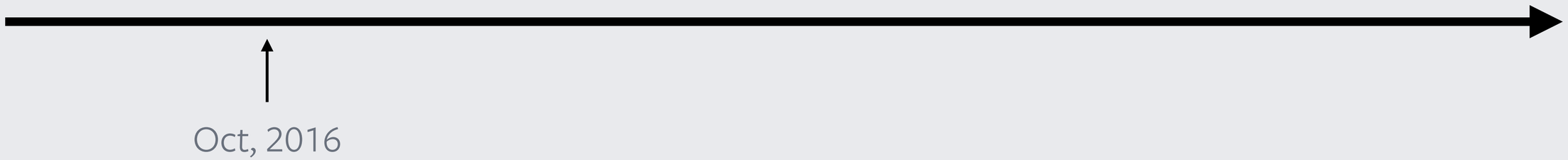
Reason for L4 load balancing:
https://atscaleconference.com/videos/networking-scale-2018-layer-4-load-balancing-at-facebook/

L4 load balancer at glance



Why?

Deployment timeline



IPVS under flood

```
top - 20:26:19 up 2 days, 23:42,  1 user,  load average: 10.99, 3.38, 1.79
Tasks: 542 total,  17 running, 525 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 ni, 97.4 id,  0.0 wa,  0.0 hi,  2.6 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu12 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.3 us,  0.0 sy,  0.0 ni, 99.3 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,100.0 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu27 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu28 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu29 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu30 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

IPVS under flood

```
Samples: 39K of event 'cycles:ppp', Event count (approx.): 28776205145
Overhead  Command      Shared Object      Symbol
  59.97%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_conn_in_get
   7.29%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_conn_expire
   4.97%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_ch_schedule
   3.32%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_conn_out_get
   2.82%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_conn_new
   2.21%  ksoftirqd/4  [kernel.kallsyms]  [k] skb_clone
   1.46%  ksoftirqd/4  [kernel.kallsyms]  [k] ipt_do_table
   1.25%  ksoftirqd/4  [kernel.kallsyms]  [k] _raw_spin_lock
   0.83%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_idents_reserve
   0.74%  ksoftirqd/4  [kernel.kallsyms]  [k] _raw_spin_lock_bh
   0.73%  ksoftirqd/4  [kernel.kallsyms]  [k] __inet_lookup_established
   0.64%  ksoftirqd/4  [kernel.kallsyms]  [k] memcpy_erms
   0.62%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
   0.53%  ksoftirqd/4  [kernel.kallsyms]  [k] cascade
   0.48%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_in.part.25
   0.47%  ksoftirqd/4  [kernel.kallsyms]  [k] netif_receive_skb_internal
   0.44%  ksoftirqd/4  [kernel.kallsyms]  [k] __slab_free
   0.44%  ksoftirqd/4  [kernel.kallsyms]  [k] __call_rcu.constprop.70
   0.44%  ksoftirqd/4  [kernel.kallsyms]  [k] fib_table_lookup
   0.43%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_conn_hashkey_param
   0.42%  ksoftirqd/4  [kernel.kallsyms]  [k] __local_bh_enable_ip
   0.35%  ksoftirqd/4  [kernel.kallsyms]  [k] ip_vs_schedule
   0.35%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_xmit
   0.31%  ksoftirqd/4  [kernel.kallsyms]  [k] kmem_cache_alloc
   0.28%  ksoftirqd/4  [kernel.kallsyms]  [k] __ip_vs_get_out_rt
   0.27%  ksoftirqd/4  [kernel.kallsyms]  [k] nf_iterate
   0.27%  ksoftirqd/4  [kernel.kallsyms]  [k] __bpf_prog_run
   0.26%  ksoftirqd/4  [kernel.kallsyms]  [k] queued_spin_lock_slowpath
```

road to XDP LB

Documentation matters

- Not everyone could have BPF maintainer in their company
- Lucky enough today we have: https://cilium.readthedocs.io/en/v1.2/bpf/

Encapsulation

what we want:

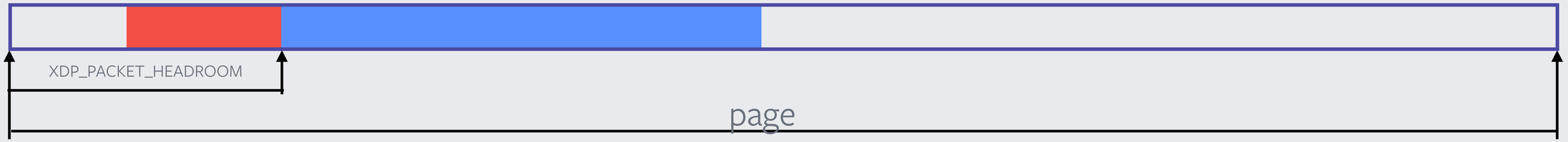
what we had:

```
commit 17bedab2723145d17b1408443074 3549e6943d03
Author: Martin KaFai Lau <kafai@fb.com>
Date:   Wed Dec 7 15:53:11 2016 -0800

    bpf: xdp: Allow head adjustment in XDP prog
```

Encapsulation

what we want:

what we had:



Encapsulation: example

```
if (bpf_xdp_adjust_head(xdp, 0 - (int)sizeof(struct ipv6hdr))) {
  return false;
}
data = (void *)(long)xdp->data;
data_end = (void *)(long)xdp->data_end;
new_eth = data;
ip6h = data + sizeof(struct eth_hdr);
```

Connection table

```
2874aa2e467d Martin KaFai Lau <kafai@fb.com>  bpf: Fix compilation warning in __bpf_lru_list_rotate_inactive
8f8449384ec3 Martin KaFai Lau <kafai@fb.com>  bpf: Add BPF_MAP_TYPE_LRU_PERCPU_HASH
29ba732acbee Martin KaFai Lau <kafai@fb.com>  bpf: Add BPF_MAP_TYPE_LRU_HASH
fd91de7b3c69 Martin KaFai Lau <kafai@fb.com>  bpf: Refactor codes handling percpu map
961578b63474 Martin KaFai Lau <kafai@fb.com>  bpf: Add percpu LRU list
3a08c2fd7634 Martin KaFai Lau <kafai@fb.com>  bpf: LRU List
```

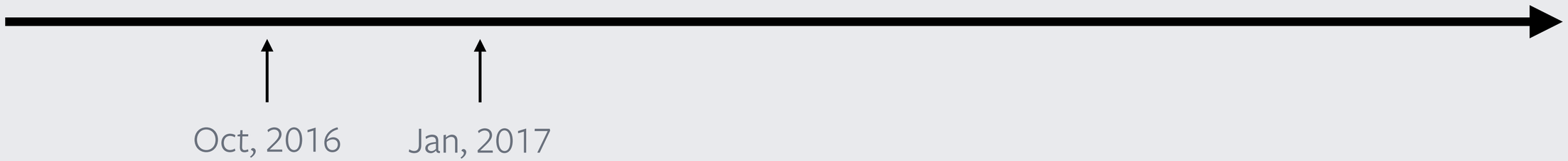
LRU types

BPF_MAP_TYPE_LRU_HASH
BPF_MAP_TYPE_LRU_PERCPU_HASH
TYPE + BPF_F_NO_COMMON_LRU

Lesson/Tip: know your environment

initial deployment + evaluation

Deployment timeline



XDP under flood

```
top - 21:27:18 up 3 days, 43 min,  2 users,  load average: 0.75, 0.58, 0.28
Tasks: 556 total,   2 running, 554 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni, 80.3 id,  0.0 wa,  0.0 hi, 19.7 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni, 84.2 id,  0.0 wa,  0.0 hi, 15.8 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni, 84.6 id,  0.0 wa,  0.0 hi, 15.4 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni, 84.4 id,  0.0 wa,  0.0 hi, 15.6 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni, 82.0 id,  0.0 wa,  0.0 hi, 18.0 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni, 82.8 id,  0.0 wa,  0.0 hi, 17.2 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni, 83.9 id,  0.0 wa,  0.0 hi, 16.1 si,  0.0 st
%Cpu11 :  0.0 us,  0.6 sy,  0.0 ni, 80.3 id,  0.0 wa,  0.0 hi, 19.1 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni, 53.8 id,  0.0 wa,  0.0 hi, 46.2 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni, 53.8 id,  0.0 wa,  0.0 hi, 46.2 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni, 52.5 id,  0.0 wa,  0.0 hi, 47.5 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni, 55.4 id,  0.0 wa,  0.0 hi, 44.6 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu27 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu28 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

Lesson: profile/measure everything that moves.

e.g. NIC's bps perf != NIC's pps perf (64 bytes)

or

v6 vs v4

or

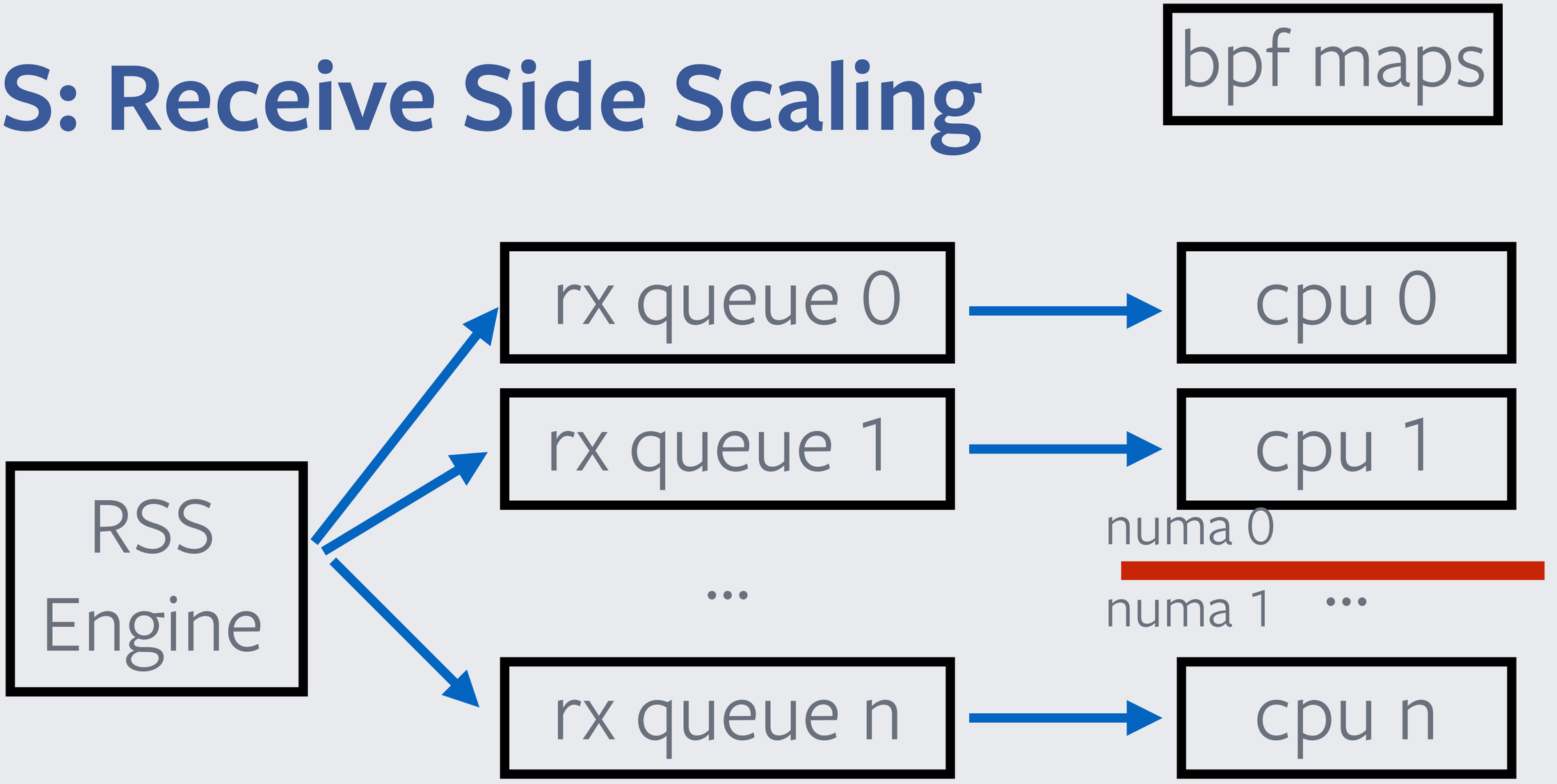
TCP vs UDP

XDP under flood

```
top - 21:27:18 up 3 days, 43 min,  2 users,  load average: 0.75, 0.58, 0.28
Tasks: 556 total,   2 running, 554 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni, 80.3 id,  0.0 wa,  0.0 hi, 19.7 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni, 84.2 id,  0.0 wa,  0.0 hi, 15.8 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni, 84.6 id,  0.0 wa,  0.0 hi, 15.4 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni, 84.4 id,  0.0 wa,  0.0 hi, 15.6 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni, 82.0 id,  0.0 wa,  0.0 hi, 18.0 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni, 82.8 id,  0.0 wa,  0.0 hi, 17.2 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni, 83.9 id,  0.0 wa,  0.0 hi, 16.1 si,  0.0 st
%Cpu11 :  0.0 us,  0.6 sy,  0.0 ni, 80.3 id,  0.0 wa,  0.0 hi, 19.1 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni, 53.8 id,  0.0 wa,  0.0 hi, 46.2 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni, 53.8 id,  0.0 wa,  0.0 hi, 46.2 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni, 52.5 id,  0.0 wa,  0.0 hi, 47.5 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni, 54.6 id,  0.0 wa,  0.0 hi, 45.4 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni, 55.4 id,  0.0 wa,  0.0 hi, 44.6 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu27 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu28 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

Our server's "topology"

RSS: Receive Side Scaling



NUMA memory allocation

```
[root@localhost katran2]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11
node 0 size: 128686 MB
node 0 free: 123862 MB
node 1 cpus: 14 15 16 17 18 19 20 21 2
node 1 size: 129019 MB
node 1 free: 128326 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@localhost katran2]#
```

XDP under flood

```
 13.82%  ksoftirqd/4     [kernel.kallsyms]  [k] __bpf_lru_list_rotate_inactive.isra.5
  9.87%  swapper         [kernel.kallsyms]  [k] __bpf_lru_list_rotate_inactive.isra.5
  8.92%  swapper         [kernel.kallsyms]  [k] cleanup_module
  7.98%  swapper         [kernel.kallsyms]  [k] intel_idle
  7.44%  swapper         [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
  7.37%  ksoftirqd/4     [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
  6.16%  ksoftirqd/4     [kernel.kallsyms]  [k] cleanup_module
  2.32%  swapper         [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
  1.82%  swapper         [kernel.kallsyms]  [k] mlx4_eq_int
  1.72%  swapper         [kernel.kallsyms]  [k] cpuidle_enter_state
  1.65%  ksoftirqd/4     [kernel.kallsyms]  [k] lookup_elem_raw
  1.43%  swapper         [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
  1.34%  swapper         [kernel.kallsyms]  [k] lookup_elem_raw
  1.06%  swapper         [kernel.kallsyms]  [k] poll_idle
  1.03%  ksoftirqd/4     [kernel.kallsyms]  [k] htab_lru_map_delete_node
  0.90%  ksoftirqd/4     [kernel.kallsyms]  [k] htab_lru_map_update_elem
  0.89%  swapper         [kernel.kallsyms]  [k] htab_lru_map_delete_node
  0.81%  swapper         [kernel.kallsyms]  [k] native_irq_return_iret
  0.78%  swapper         [kernel.kallsyms]  [k] tasklet_action
  0.77%  swapper         [kernel.kallsyms]  [k] bpf_map_lookup_elem
  0.76%  swapper         [kernel.kallsyms]  [k] htab_lru_map_update_elem
  0.71%  swapper         [kernel.kallsyms]  [k] mlx4_en_xmit_frame
  0.63%  swapper         [kernel.kallsyms]  [k] __alloc_pages_nodemask
  0.62%  swapper         [kernel.kallsyms]  [k] htab_map_lookup_elem
  0.59%  swapper         [kernel.kallsyms]  [k] menu_select
  0.58%  swapper         [kernel.kallsyms]  [k] memcmp
  0.55%  swapper         [kernel.kallsyms]  [k] napi_complete_done
  0.55%  ksoftirqd/4     [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
  0.54%  swapper         [kernel.kallsyms]  [k] mlx4_en_prepare_rx_desc
  0.47%  swapper         [kernel.kallsyms]  [k] mlx4_en_recycle_tx_desc
  0.43%  ksoftirqd/4     [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
```

output from cpu on the same NUMA node as bpf's map

For fun: LRU w/o BPF_F_NO_COMMON_LRU

```
Samples: 31K of event 'cycles:ppp', Event count (approx.): 22775786748
Overhead  Command      Shared Object       Symbol
  75.82%  ksoftirqd/4  [kernel.kallsyms]   [k] queued_spin_lock_slowpath
   6.59%  ksoftirqd/4  [kernel.kallsyms]   [k] cleanup_module
   4.61%  ksoftirqd/4  [kernel.kallsyms]   [k] _raw_spin_lock_irqsave
   2.19%  ksoftirqd/4  [kernel.kallsyms]   [k] lookup_elem_raw
   1.75%  ksoftirqd/4  [kernel.kallsyms]   [k] htab_lru_map_delete_node
   0.85%  ksoftirqd/4  [kernel.kallsyms]   [k] mlx4_en_process_rx_cq
   0.72%  ksoftirqd/4  [kernel.kallsyms]   [k] htab_lru_map_update_elem
   0.47%  swapper      [kernel.kallsyms]   [k] intel_idle
   0.42%  ksoftirqd/4  [kernel.kallsyms]   [k] htab_map_lookup_elem
   0.41%  swapper      [kernel.kallsyms]   [k] cleanup_module
   0.36%  ksoftirqd/4  [kernel.kallsyms]   [k] mlx4_en_poll_tx_cq
   0.35%  ksoftirqd/4  [kernel.kallsyms]   [k] _raw_spin_unlock_irqrestore
   0.35%  ksoftirqd/4  [kernel.kallsyms]   [k] __bpf_lru_list_shrink
   0.33%  ksoftirqd/4  [kernel.kallsyms]   [k] bpf_map_lookup_elem
```

XDP under flood

```
 13.82%  ksoftirqd/4     [kernel.kallsyms]  [k] __bpf_lru_list_rotate_inactive.isra.5
  9.87%  swapper         [kernel.kallsyms]  [k] __bpf_lru_list_rotate_inactive.isra.5
  8.92%  swapper         [kernel.kallsyms]  [k] cleanup_module
  7.98%  swapper         [kernel.kallsyms]  [k] intel_idle
  7.44%  swapper         [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
  7.37%  ksoftirqd/4     [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
  6.16%  ksoftirqd/4     [kernel.kallsyms]  [k] cleanup_module
  2.32%  swapper         [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
  1.82%  swapper         [kernel.kallsyms]  [k] mlx4_eq_int
  1.72%  swapper         [kernel.kallsyms]  [k] cpuidle_enter_state
  1.65%  ksoftirqd/4     [kernel.kallsyms]  [k] lookup_elem_raw
  1.43%  swapper         [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
  1.34%  swapper         [kernel.kallsyms]  [k] lookup_elem_raw
  1.06%  swapper         [kernel.kallsyms]  [k] poll_idle
  1.03%  ksoftirqd/4     [kernel.kallsyms]  [k] htab_lru_map_delete_node
  0.90%  ksoftirqd/4     [kernel.kallsyms]  [k] htab_lru_map_update_elem
  0.89%  swapper         [kernel.kallsyms]  [k] htab_lru_map_delete_node
  0.81%  swapper         [kernel.kallsyms]  [k] native_irq_return_iret
  0.78%  swapper         [kernel.kallsyms]  [k] tasklet_action
  0.77%  swapper         [kernel.kallsyms]  [k] bpf_map_lookup_elem
  0.76%  swapper         [kernel.kallsyms]  [k] htab_lru_map_update_elem
  0.71%  swapper         [kernel.kallsyms]  [k] mlx4_en_xmit_frame
  0.63%  swapper         [kernel.kallsyms]  [k] __alloc_pages_nodemask
  0.62%  swapper         [kernel.kallsyms]  [k] htab_map_lookup_elem
  0.59%  swapper         [kernel.kallsyms]  [k] menu_select
  0.58%  swapper         [kernel.kallsyms]  [k] memcmp
  0.55%  swapper         [kernel.kallsyms]  [k] napi_complete_done
  0.55%  ksoftirqd/4     [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
  0.54%  swapper         [kernel.kallsyms]  [k] mlx4_en_prepare_rx_desc
  0.47%  swapper         [kernel.kallsyms]  [k] mlx4_en_recycle_tx_desc
  0.43%  ksoftirqd/4     [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
```

output from cpu on the same NUMA node as bpf's map

XDP under flood (stateless)

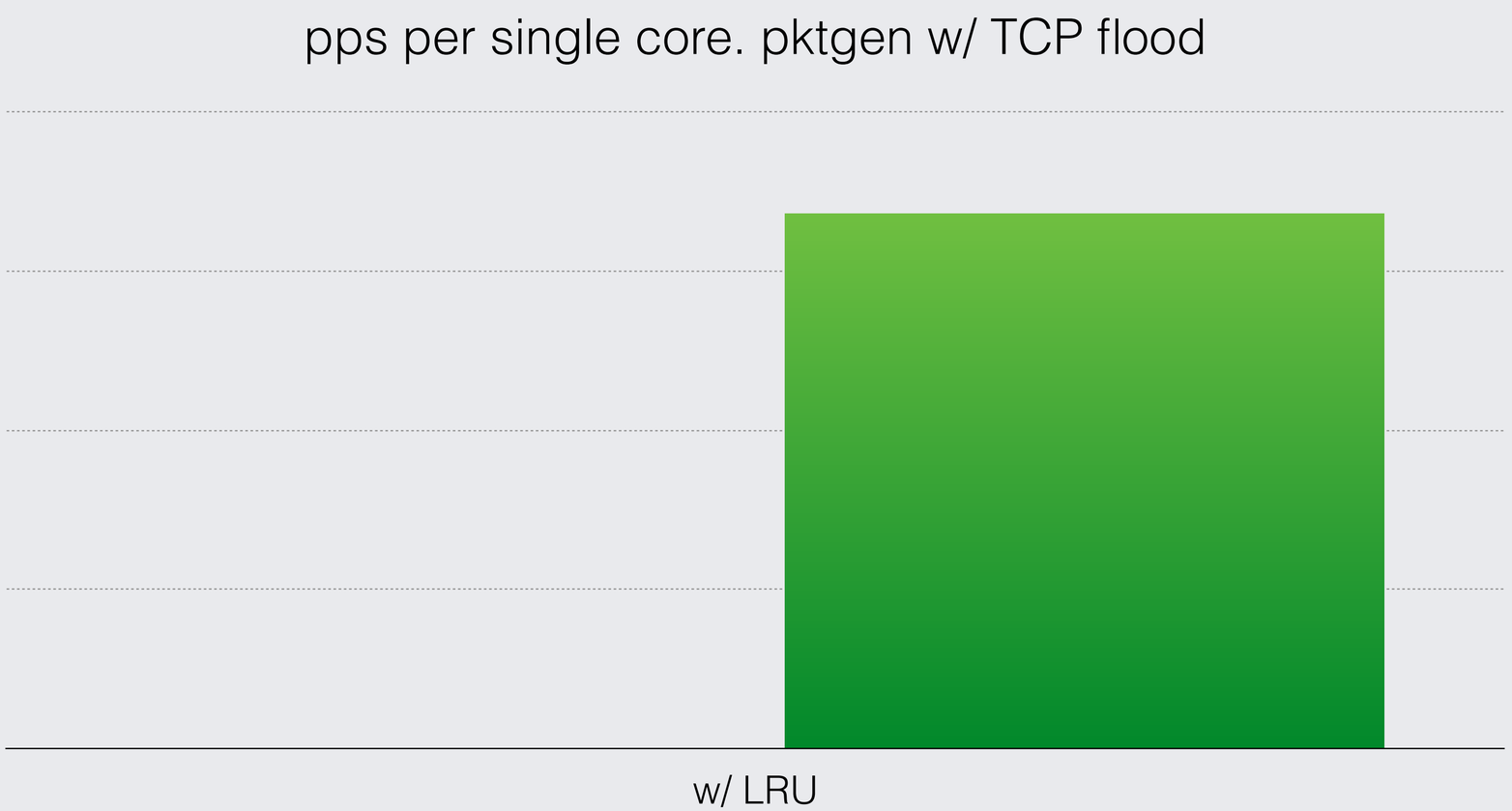
```
top - 16:49:31 up  2:01,  2 users,  load average: 3.29, 7.54, 5.90
Tasks: 551 total,   1 running, 550 sleeping,   0 stopped,   0 zombie
%Cpu0  :  2.0 us,  0.3 sy,  0.0 ni, 97.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.7 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu6  :  0.7 us,  0.7 sy,  0.0 ni, 97.3 id,  0.0 wa,  0.0 hi,  1.3 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni, 98.2 id,  0.0 wa,  0.0 hi,  1.8 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

XDP under flood (stateless)

```
Samples: 39K of event 'cycles:ppp', Event count (approx.): 32743868310
Overhead  Command      Shared Object      Symbol
  39.14%  ksoftirqd/4  [kernel.kallsyms]  [k] cleanup_module
   8.30%  ksoftirqd/4  [kernel.kallsyms]  [k] bpf_map_lookup_elem
   7.99%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
   7.97%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
   6.44%  ksoftirqd/4  [kernel.kallsyms]  [k] htab_map_lookup_elem
   5.96%  ksoftirqd/4  [kernel.kallsyms]  [k] memcmp
   5.28%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_xmit_frame
   2.75%  ksoftirqd/4  [kernel.kallsyms]  [k] array_map_lookup_elem
   2.39%  ksoftirqd/4  [kernel.kallsyms]  [k] ktime_get_mono_fast_ns
   2.39%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_prepare_rx_desc
   1.48%  ksoftirqd/4  [kernel.kallsyms]  [k] bpf_xdp_adjust_head
   1.17%  ksoftirqd/4  [kernel.kallsyms]  [k] percpu_array_map_lookup_elem
   1.15%  ksoftirqd/4  [kernel.kallsyms]  [k] lookup_elem_raw
   1.01%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_recycle_tx_desc
   0.97%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_en_rx_recycle
   0.78%  ksoftirqd/4  [kernel.kallsyms]  [k] mlx4_eq_int
   0.70%  ksoftirqd/4  [kernel.kallsyms]  [k] bpf_ktime_get_ns
   0.43%  ksoftirqd/4  [kernel.kallsyms]  [k] swiotlb_sync_single_for_cpu
   0.42%  ksoftirqd/4  [kernel.kallsyms]  [k] native_irq_return_iret
   0.37%  ksoftirqd/4  [kernel.kallsyms]  [k] read_tsc
   0.34%  ksoftirqd/4  [kernel.kallsyms]  [k] net_rx_action
   0.31%  ksoftirqd/4  [kernel.kallsyms]  [k] swiotlb_sync_single_for_device
```

output from cpu on the same NUMA node as bpf's map

Single core performance



JIT is your friend

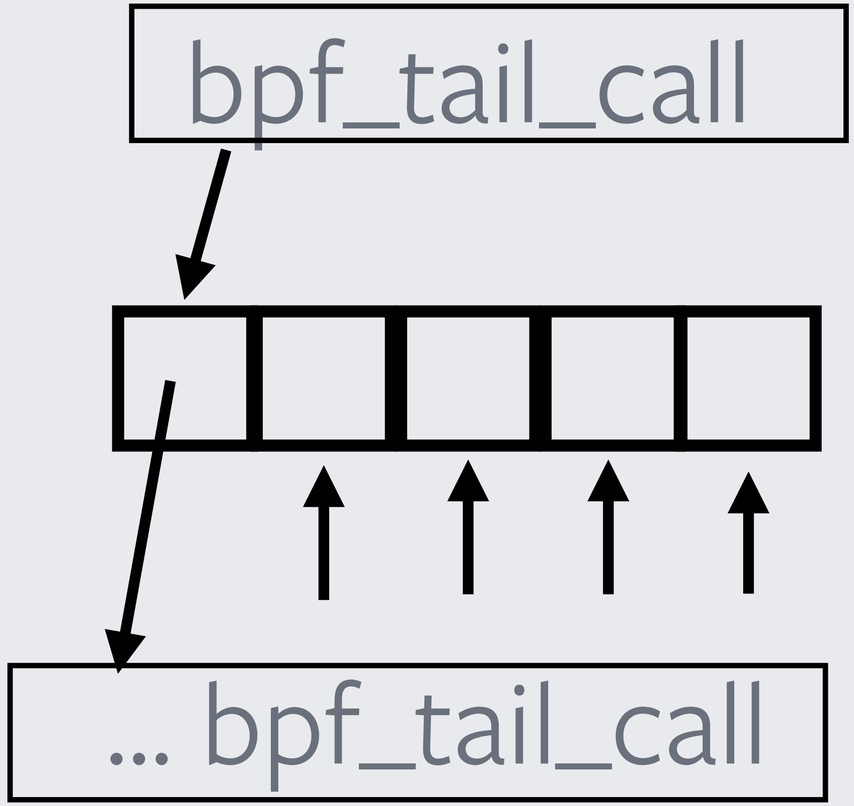
net.core.bpf_jit_enable = 1

kernel's interface counters wont show the whole picture

your application must count bytes and packets as well

XDP programs chaining

BPF_MAP_TYPE_PROG_ARRAY



BPF_MAP_TYPE_PROG_ARRAY

```
struct bpf_map_def SEC("maps") root_array = {
  .type = BPF_MAP_TYPE_PROG_ARRAY,
  .key_size = sizeof(__u32),
  .value_size = sizeof(__u32),
  .max_entries = ROOT_ARRAY_SIZE,
};

SEC("xdp-root")
int xdp_root(struct xdp_md *ctx) {
  __u32 *fd;
  #pragma clang loop unroll(full)
  for (__u32 i = 0; i < ROOT_ARRAY_SIZE; i++) {
    bpf_tail_call(ctx, &root_array, i);
  }
  return XDP_PASS;
}
```

```
  #pragma clang loop unroll(full)
  for (int i = 1; i < 16; i++) {
   jmp.call((void *)ctx, i);
  }
  return XDP_PASS;
}
```

tcpdump will stop to work

"xdpdump" for the rescue

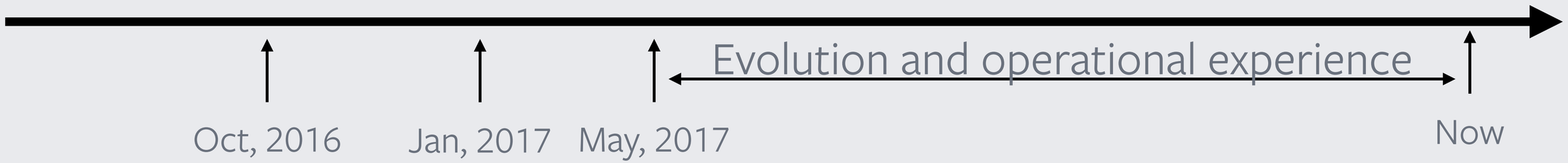
"light" version available in katran's github repo.

```
    output.sport = pckt.port16[0];
    output.dport = pckt.port16[1];
    output.proto = pckt.proto;
    output.pkt_size = data_end - data;
    __u16 data_len = output.pkt_size < MAX_LEN ? output.pkt_size : MAX_LEN;
    output.data_len = data_len;
    perf_event_map.perf_submit_skb(xdp, data_len, &output, sizeof(output));
    return;
```

```
 sudo ./build/tools/xdpdump/xdpdump -map_path /sys/fs/bpf/jmp_enp0s3 -proto 6 -dport 22 -src 10.0.2.2
src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 90 chunk size: 90

src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 60 chunk size: 60
src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 60 chunk size: 60
src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 60 chunk size: 60
src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 60 chunk size: 60
src: 10.0.2.2 dst: 10.0.2.15
proto: 6 sport: 52840 dport: 22 pkt size: 60 chunk size: 60
src: 10.0.2.2 dst: 10.0.2.15
```

```
tcpdump -ennnvvvr /tmp/out.pcap
reading from file /tmp/out.pcap, link-type EN10MB (Ethernet)
09:14:11.715640 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 90: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [P.], cksum 0x7060 (correct), seq 39355439:39355475, ack 1894993876
09:14:11.716611 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x8026 (correct), seq 36, ack 429, win 65535, length 0
09:14:11.717068 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7fe2 (correct), seq 36, ack 497, win 65535, length 0
09:14:11.717541 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7ed6 (correct), seq 36, ack 765, win 65535, length 0
09:14:11.717770 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7ea2 (correct), seq 36, ack 817, win 65535, length 0
09:14:11.718186 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7da6 (correct), seq 36, ack 1069, win 65535, length 0
09:14:11.718743 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7d0a (correct), seq 36, ack 1225, win 65535, length 0
09:14:11.719199 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7c6e (correct), seq 36, ack 1381, win 65535, length 0
09:14:11.719757 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7bd2 (correct), seq 36, ack 1537, win 65535, length 0
09:14:11.720345 52:54:00:12:35:02 > 08:00:27:24:44:6b, ethertype IPv4 (0x0800), length 60: (tos 0x0, ttl 64,
    10.0.2.2.52840 > 10.0.2.15.22: Flags [.], cksum 0x7b36 (correct), seq 36, ack 1693, win 65535, length 0
```

Deployment timeline



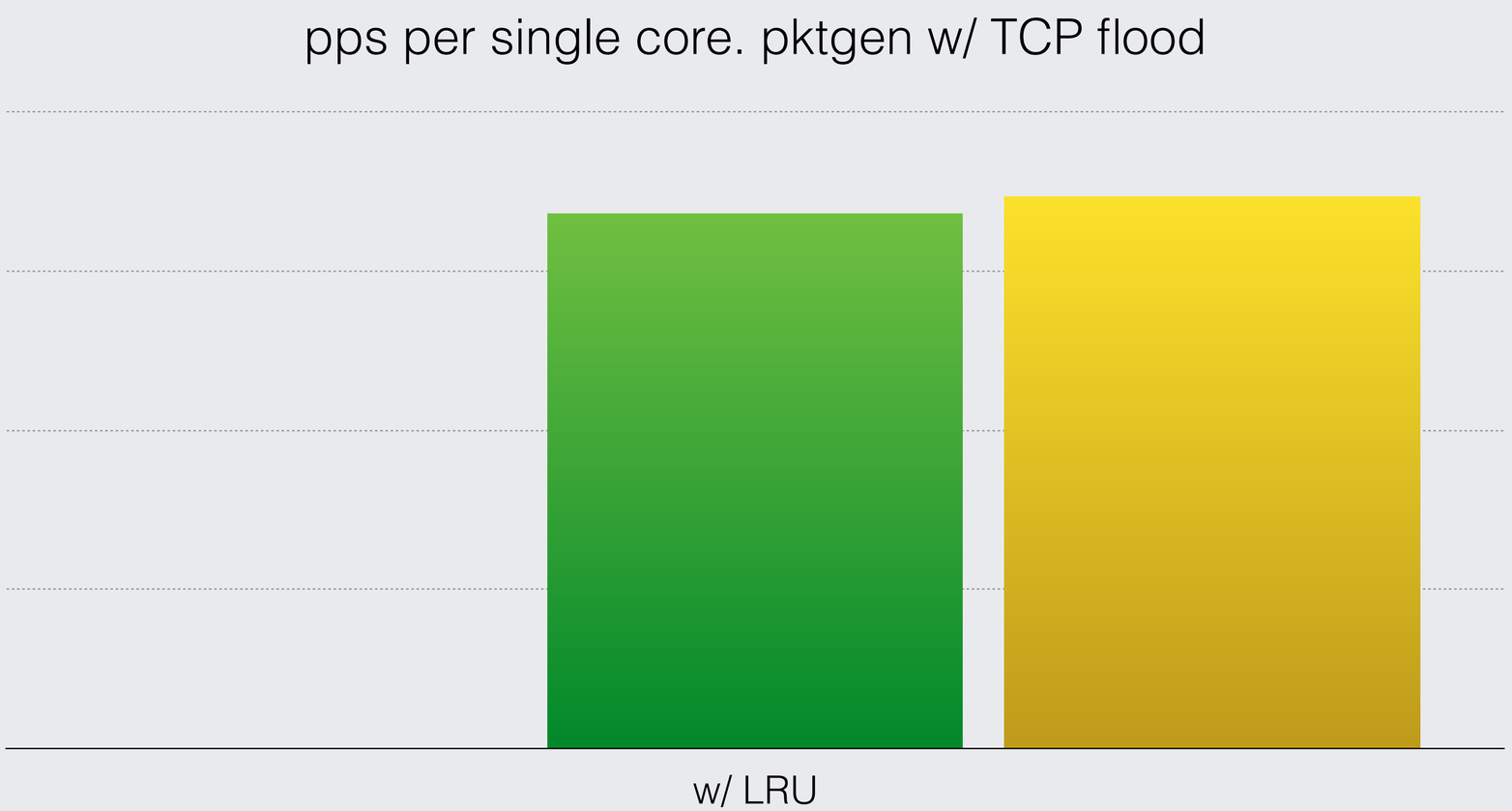
Evolution

Maps lookup

our code: ~6 arrays lookup ~3 hash lookups

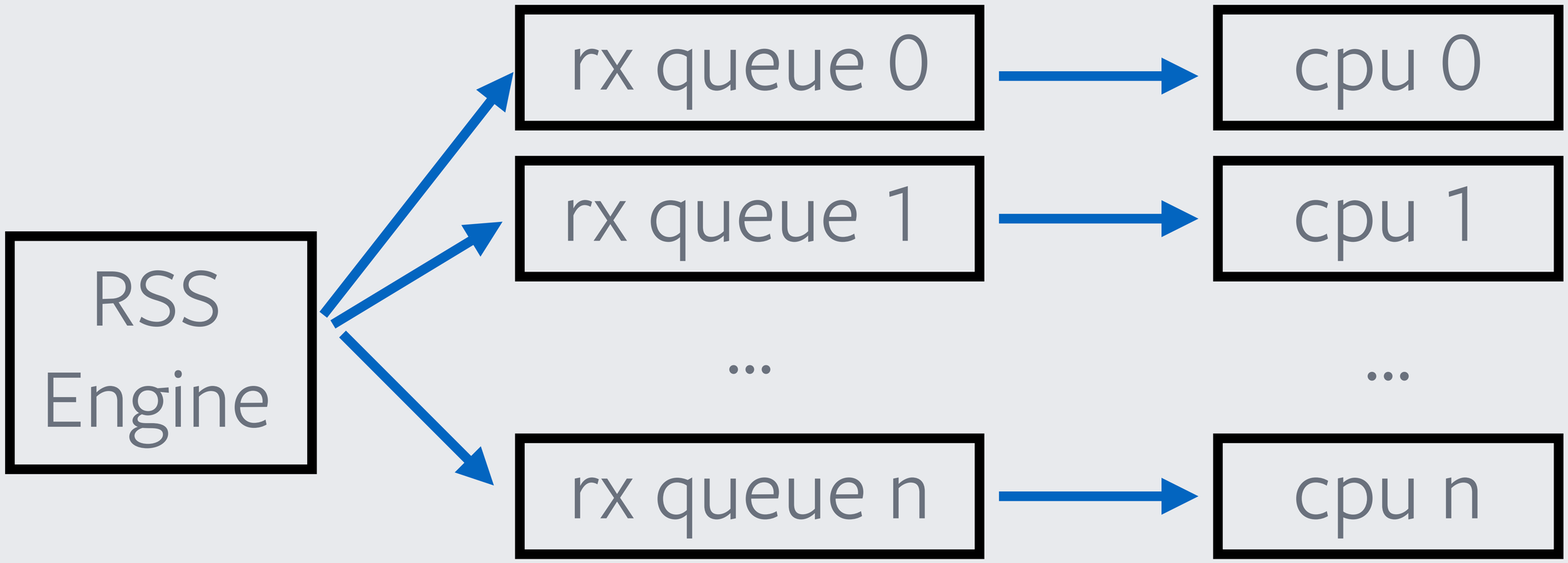
```
commit 9015d2f5953590e8273392b44c2b0f864350b427
Author: Alexei Starovoitov <ast@fb.com>
Date:   Wed Mar 15 18:26:43 2017 -0700

    bpf: inline htab_map_lookup_elem()

    Optimize:
    bpf_call
      bpf_map_lookup_elem
        map->ops->map_lookup_elem
          htab_map_lookup_elem
            __htab_map_lookup_elem
    into:
    bpf_call
      __htab_map_lookup_elem

    to improve performance of JITed programs.
```

Single core performance



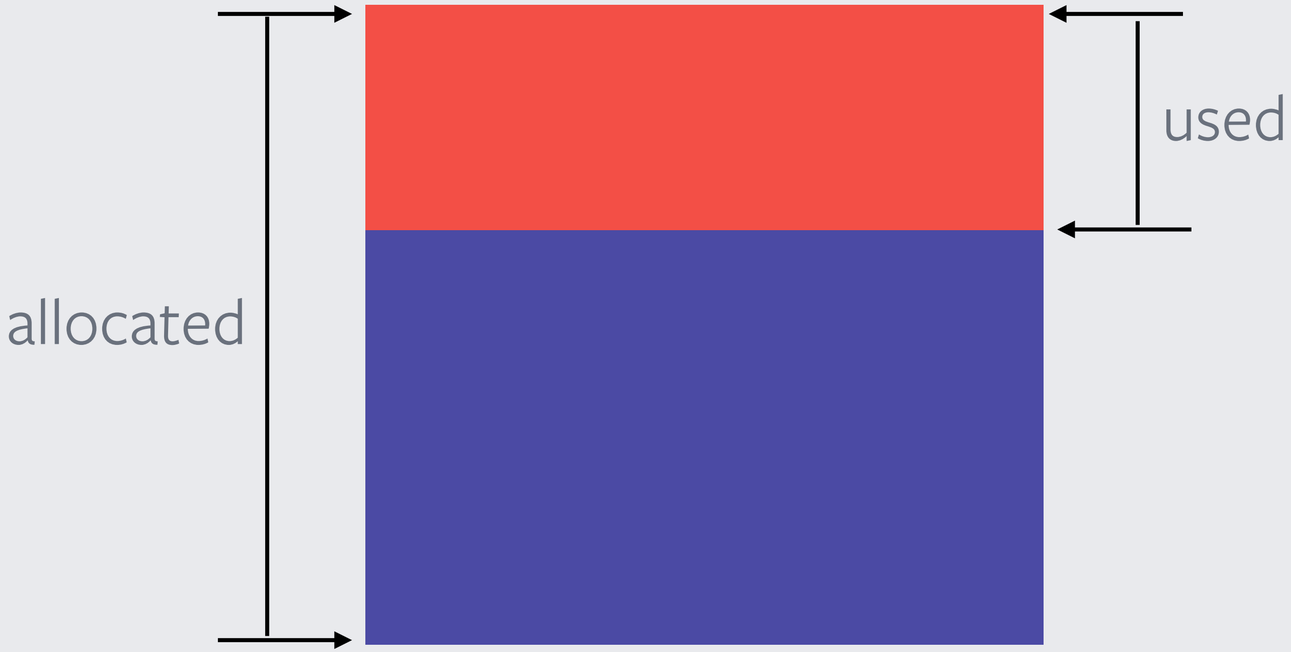
+3% perf

RSS: Receive Side Scaling



RSS and multiple CPUs

num queues < num cores



for us only ~30% of memory was used

what if we could allocate maps only for ”forwarding” cores

Map-in-Map support

```
bcc6b1b7ebf8 Martin KaFai Lau <kafai@fb.com>  bpf: Add hash of maps support
56f668dfe00d Martin KaFai Lau <kafai@fb.com>  bpf: Add array of maps support
fad73a1a35ea Martin KaFai Lau <kafai@fb.com>  bpf: Fix and simplifications on inline map lookup
```

Map-in-Map in action

```
__u32 cpu_num = bpf_get_smp_processor_id();
void *lru_map = bpf_map_lookup_elem(&lru_maps_mapping, &cpu_num);
```

Before Map-in-Map

```
[root@localhost katran2]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11
node 0 size: 128686 MB
node 0 free: 123862 MB
node 1 cpus: 14 15 16 17 18 19 20 21 2
node 1 size: 129019 MB
node 1 free: 128326 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@localhost katran2]#
```

After Map-in-Map

```
[root@localhost katran2]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 1
node 0 size: 128686 MB
node 0 free: 126259 MB
node 1 cpus: 14 15 16 17 18 19 20 21 
node 1 size: 129019 MB
node 1 free: 128607 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@localhost katran2]#
```

~3G

XDP under flood

```
top - 17:28:51 up 17 min,  2 users,  load average: 0.07, 0.03, 0.01
Tasks: 572 total,   1 running, 571 sleeping,   0 stopped,   0 zombie
%Cpu0  :  2.3 us,  1.0 sy,  0.0 ni, 96.0 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni, 94.6 id,  0.0 wa,  0.0 hi,  5.4 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni, 93.8 id,  0.0 wa,  0.0 hi,  6.2 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni, 95.4 id,  0.0 wa,  0.0 hi,  4.6 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni, 96.1 id,  0.0 wa,  0.0 hi,  3.9 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni, 96.1 id,  0.0 wa,  0.0 hi,  3.9 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni, 94.5 id,  0.0 wa,  0.0 hi,  5.5 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni, 94.7 id,  0.0 wa,  0.0 hi,  5.3 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 ni, 93.2 id,  0.0 wa,  0.0 hi,  6.8 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni, 63.4 id,  0.0 wa,  0.0 hi, 36.6 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni, 61.4 id,  0.0 wa,  0.0 hi, 38.6 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni, 61.3 id,  0.0 wa,  0.0 hi, 38.7 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni, 62.8 id,  0.0 wa,  0.0 hi, 37.2 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni, 62.9 id,  0.0 wa,  0.0 hi, 37.1 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni, 61.6 id,  0.0 wa,  0.0 hi, 38.4 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni, 63.2 id,  0.0 wa,  0.0 hi, 36.8 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni, 67.1 id,  0.0 wa,  0.0 hi, 32.9 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu27 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu28 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu29 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu30 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu31 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

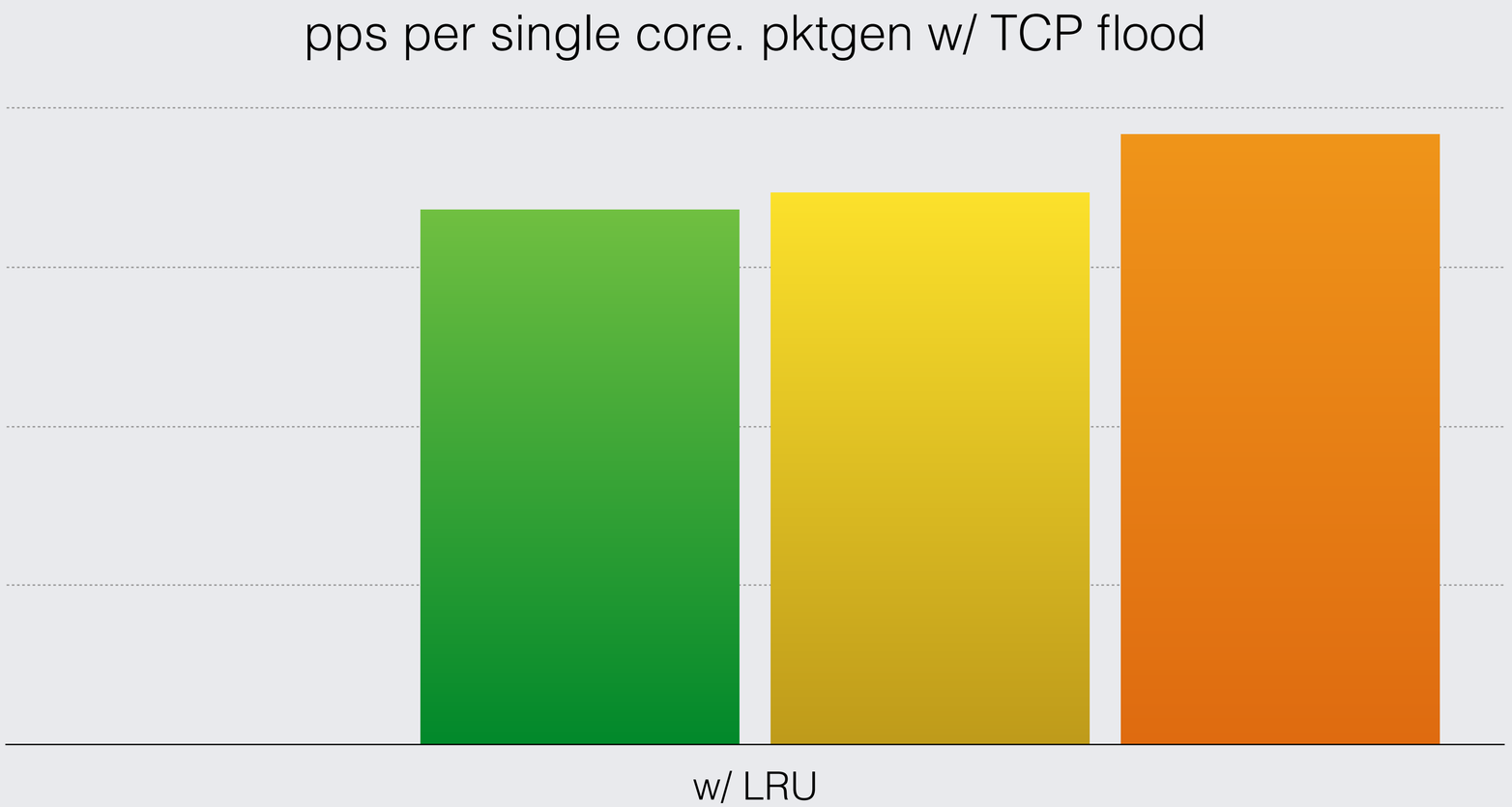
XDP under flood (local NUMA)

```
Samples: 6K of event 'cycles:ppp', Event count (approx.): 4965165348
Overhead  Command      Shared Object      Symbol
  16.21%  swapper      [kernel.kallsyms]  [k] bpf_prog_f3c85e56f45806c0
   9.57%  swapper      [kernel.kallsyms]  [k] intel_idle
   8.23%  swapper      [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
   5.36%  ksoftirqd/4  [kernel.kallsyms]  [k] bpf_prog_f3c85e56f45806c0
   4.39%  swapper      [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
   4.10%  ksoftirqd/4  [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
   3.73%  swapper      [kernel.kallsyms]  [k] poll_idle
   2.24%  swapper      [kernel.kallsyms]  [k] mlx4_eq_int
   2.13%  swapper      [kernel.kallsyms]  [k] lookup_elem_raw
   2.13%  swapper      [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
   2.03%  swapper      [kernel.kallsyms]  [k] htab_lru_map_delete_node
   1.72%  swapper      [kernel.kallsyms]  [k] cpuidle_enter_state
   1.55%  swapper      [kernel.kallsyms]  [k] htab_lru_map_update_elem
   1.55%  swapper      [kernel.kallsyms]  [k] mlx4_en_recycle_tx_desc
   1.35%  ksoftirqd/4  [kernel.kallsyms]  [k] htab_lru_map_delete_node
   1.27%  swapper      [kernel.kallsyms]  [k] __htab_map_lookup_elem
   1.18%  swapper      [kernel.kallsyms]  [k] __alloc_pages_nodemask
   1.17%  ksoftirqd/4  [kernel.kallsyms]  [k] lookup_elem_raw
   1.14%  swapper      [kernel.kallsyms]  [k] tasklet_action
   1.06%  swapper      [kernel.kallsyms]  [k] memcmp
   1.02%  swapper      [kernel.kallsyms]  [k] mlx4_en_xmit_frame
   0.97%  swapper      [kernel.kallsyms]  [k] native_irq_return_iret
   0.92%  swapper      [kernel.kallsyms]  [k] mlx4_en_prepare_rx_desc
   0.79%  swapper      [kernel.kallsyms]  [k] get_page_from_freelist
   0.74%  ksoftirqd/4  [kernel.kallsyms]  [k] htab_lru_map_update_elem
   0.72%  swapper      [kernel.kallsyms]  [k] menu_select
   0.69%  swapper      [kernel.kallsyms]  [k] cpu_startup_entry
```

XDP under flood (remote NUMA)

```
Samples: 11K of event 'cycles:ppp', Event count (approx.): 8856982421
Overhead  Command          Shared Object      Symbol
  16.77%  ksoftirqd/19     [kernel.kallsyms]  [k] bpf_prog_f3c85e56f45806c0
  12.63%  ksoftirqd/19     [kernel.kallsyms]  [k] _raw_spin_lock_irqsave
  10.84%  swapper          [kernel.kallsyms]  [k] bpf_prog_f3c85e56f45806c0
   5.96%  swapper          [kernel.kallsyms]  [k] intel_idle
   4.88%  ksoftirqd/19     [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
   4.64%  ksoftirqd/19     [kernel.kallsyms]  [k] htab_lru_map_delete_node
   4.06%  ksoftirqd/19     [kernel.kallsyms]  [k] lookup_elem_raw
   3.47%  swapper          [kernel.kallsyms]  [k] mlx4_en_process_rx_cq
   2.24%  swapper          [kernel.kallsyms]  [k] mlx4_eq_int
   1.91%  ksoftirqd/19     [kernel.kallsyms]  [k] htab_lru_map_update_elem
   1.55%  swapper          [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
   1.33%  ksoftirqd/19     [kernel.kallsyms]  [k] __bpf_lru_list_shrink
   1.21%  swapper          [kernel.kallsyms]  [k] __alloc_pages_nodemask
   1.21%  ksoftirqd/19     [kernel.kallsyms]  [k] __htab_map_lookup_elem
   1.11%  ksoftirqd/19     [kernel.kallsyms]  [k] mlx4_en_poll_tx_cq
   1.08%  swapper          [kernel.kallsyms]  [k] cpuidle_enter_state
   0.96%  swapper          [kernel.kallsyms]  [k] mlx4_en_recycle_tx_desc
   0.90%  ksoftirqd/19     [kernel.kallsyms]  [k] mlx4_en_xmit_frame
   0.84%  swapper          [kernel.kallsyms]  [k] tasklet_action
   0.79%  ksoftirqd/19     [kernel.kallsyms]  [k] memcmp
```

Single core performance



+14% (+10%) perf

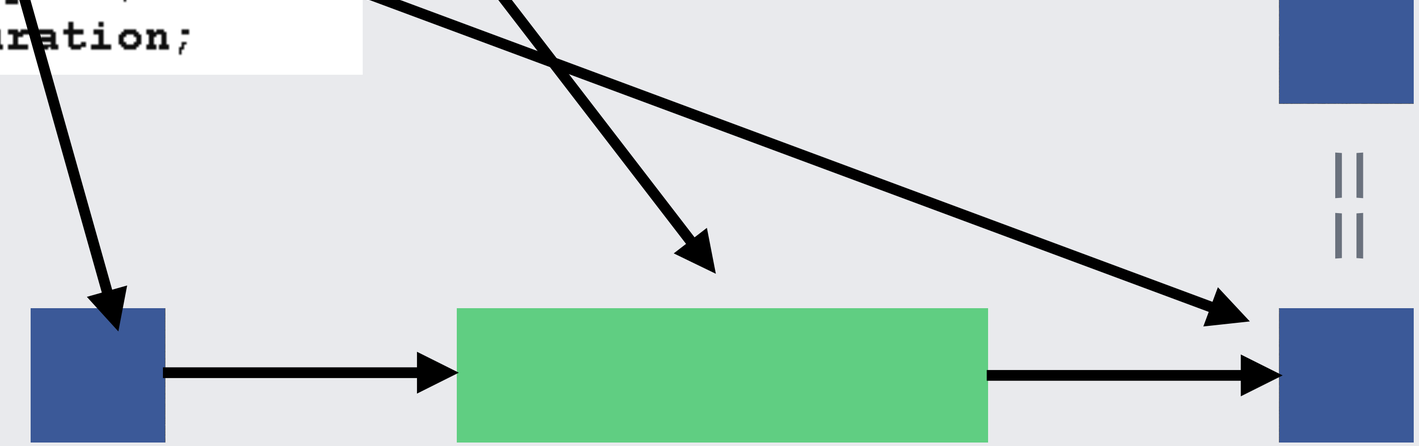
I wish i could unit test XDP program

BPF_PROG_TEST_RUN

```
struct { /* anonymous struct used by BPF_PROG_TEST_RUN command */
        __u32           prog_fd;
        __u32           retval;
        __u32           data_size_in;
        __u32           data_size_out;
        __aligned_u64   data_in;
        __aligned_u64   data_out;
        __u32           repeat;
        __u32           duration;
} test;
```

BPF_PROG_TEST_RUN

```
__u32           prog_fd;
__u32           retval;
__u32           data_size_in;
__u32           data_size_out;
__aligned_u64   data_in;
__aligned_u64   data_out;
__u32           repeat;
__u32           duration;
```



BPF_PROG_TEST_RUN

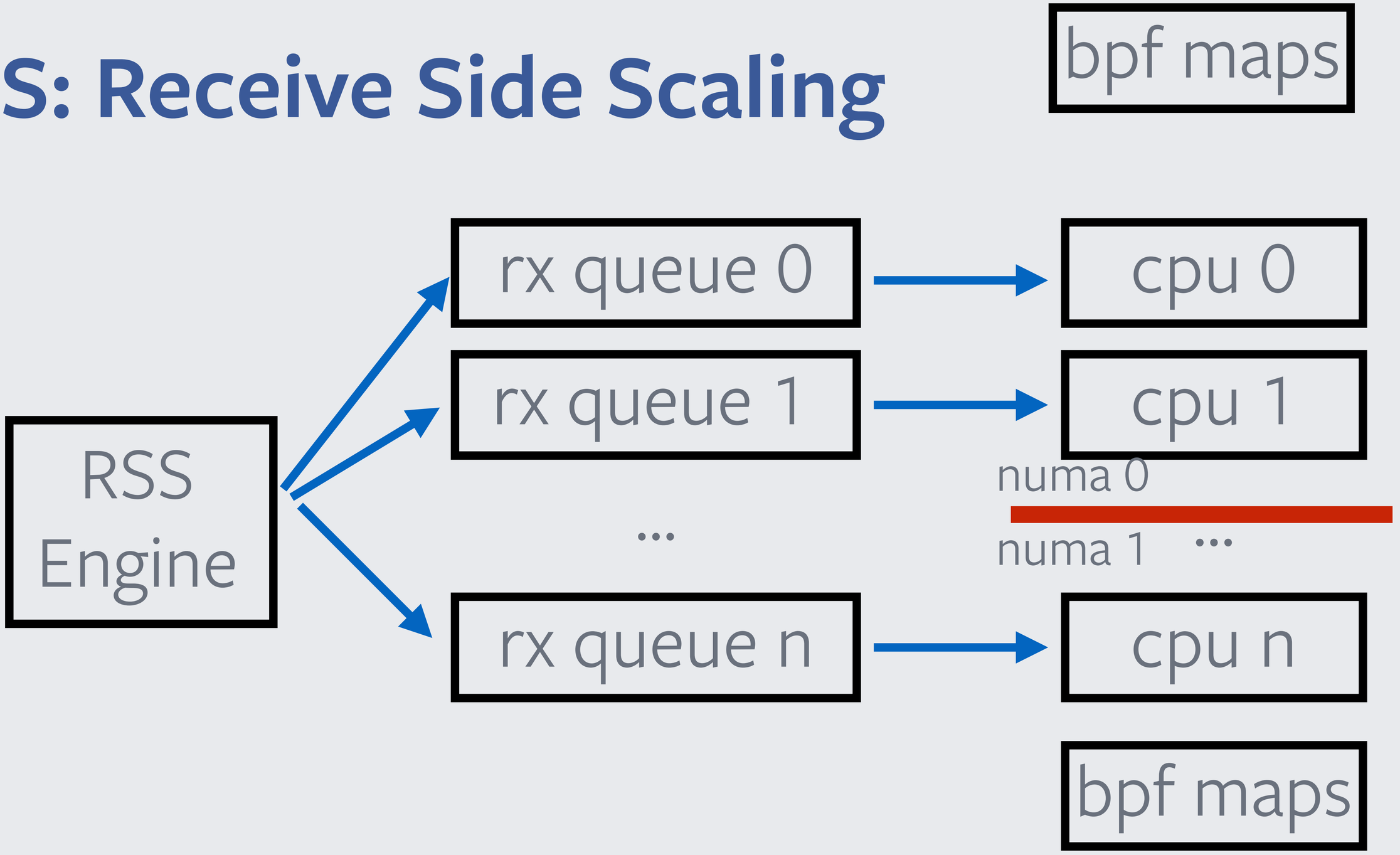
```
{
  // Ether(src="0x1", dst="0x2")/IP(src="192.168.1.1", dst="10.200.1.1")/UDP(sport=31337, dport=80)/"katran test pkt"
  "AgAAAAAAAQAAAAAACABFAAArAAEAAEARrU/AqAEBCsgBAXppAFAAF5fea2F0cmFuIHRlc3QgcGt0",
  "packet to UDP based v4 VIP (and v4 real)"
},
```

```
 {
   "AADerb6vAgAAAAAACABFAAA/AAAAAEAEXC2sEGh7CgAAA0UAACsAAQAAQBGtT8CoAQEKyAEBemkAUAAXl95rYXRyYW4gdGVzdCBwa3Q=",
   "XDP_TX"
 },
```

BPF_PROG_TEST_RUN

```
I0904 19:25:18.770014 20857 XdpTester.cpp:167] Test: packet to UDP based v4 VIP (and v4 real)                    result: Passed
I0904 19:25:18.770071 20857 XdpTester.cpp:167] Test: packet to TCP based v4 VIP (and v4 real)                    result: Passed
I0904 19:25:18.770124 20857 XdpTester.cpp:167] Test: packet to TCP based v4 VIP (and v4 real; any dst ports).    result: Passed
I0904 19:25:18.770166 20857 XdpTester.cpp:167] Test: packet to TCP based v4 VIP (and v6 real)                    result: Passed
I0904 19:25:18.770216 20857 XdpTester.cpp:167] Test: packet to TCP based v6 VIP (and v6 real)                    result: Passed
I0904 19:25:18.770243 20857 XdpTester.cpp:167] Test: v4 ICMP echo-request                                        result: Passed
I0904 19:25:18.770279 20857 XdpTester.cpp:167] Test: v6 ICMP echo-request                                        result: Passed
I0904 19:25:18.770328 20857 XdpTester.cpp:167] Test: v4 ICMP dest-unreachabe fragmentation-needed               result: Passed
I0904 19:25:18.770391 20857 XdpTester.cpp:167] Test: v6 ICMP packet-too-big                                      result: Passed
I0904 19:25:18.770426 20857 XdpTester.cpp:167] Test: drop of IPv4 packet w/ options                             result: Passed
I0904 19:25:18.770458 20857 XdpTester.cpp:167] Test: drop of IPv4 fragmented packet                             result: Passed
I0904 19:25:18.770498 20857 XdpTester.cpp:167] Test: drop of IPv6 fragmented packet                             result: Passed
I0904 19:25:18.770531 20857 XdpTester.cpp:167] Test: pass of v4 packet with dst not equal to any configured VIP result: Passed
I0904 19:25:18.770571 20857 XdpTester.cpp:167] Test: pass of v6 packet with dst not equal to any configured VIP result: Passed
I0904 19:25:18.770597 20857 XdpTester.cpp:167] Test: pass of arp packet                                         result: Passed
I0904 19:25:18.770635 20857 XdpTester.cpp:167] Test: LRU hit                                                    result: Passed
I0904 19:25:18.770673 20857 XdpTester.cpp:167] Test: packet #1 dst port hashing only                            result: Passed
I0904 19:25:18.770711 20857 XdpTester.cpp:167] Test: packet #2 dst port hashing only                            result: Passed
I0904 19:25:18.770748 20857 XdpTester.cpp:167] Test: ipinip packet                                              result: Passed
I0904 19:25:18.770799 20857 XdpTester.cpp:167] Test: ipv6inipv6 packet                                          result: Passed
I0904 19:25:18.770843 20857 XdpTester.cpp:167] Test: ipv4inipv6 packet                                          result: Passed
I0904 19:25:18.770880 20857 XdpTester.cpp:167] Test: QUIC: long header. Client Initial type. LRU miss           result: Passed
I0904 19:25:18.770918 20857 XdpTester.cpp:167] Test: QUIC: long header. 0-RTT Protected. CH. LRU hit.           result: Passed
I0904 19:25:18.770956 20857 XdpTester.cpp:167] Test: QUIC: long header. Handshake. v4 vip v6 real. Conn Id based. result: Passed
I0904 19:25:18.771003 20857 XdpTester.cpp:167] Test: QUIC: long header. client initial. v6 vip v6 real. LRU miss result: Passed
I0904 19:25:18.771033 20857 XdpTester.cpp:167] Test: QUIC: short header. No connection id. CH. LRU hit        result: Passed
I0904 19:25:18.771068 20857 XdpTester.cpp:167] Test: QUIC: short header w/ connection id                        result: Passed
I0904 19:25:18.771100 20857 XdpTester.cpp:167] Test: QUIC: short header w/ connection id but non-existing mapping result: Passed
I0904 19:25:18.771131 20857 XdpTester.cpp:167] Test: QUIC: short header w/ conn id. host id = 0. CH. LRU hit    result: Passed
```

RSS: Receive Side Scaling



bpf map's NUMA hint

```
commit 96eabe7a40aa17e613cf3db2c742ee8b1fc764d0
Author: Martin KaFai Lau <kafai@fb.com>
Date:   Fri Aug 18 11:28:00 2017 -0700

    bpf: Allow selecting numa node during map creation

    The current map creation API does not allow to provide the numa-node
    preference.  The memory usually comes from where the map-creation-process
    is running.  The performance is not ideal if the bpf_prog is known to
    always run in a numa node different from the map-creation-process.

    One of the use case is sharding on CPU to different LRU maps (i.e.
    an array of LRU maps).  Here is the test result of map_perf_test on
    the INNER_LRU_HASH_PREALLOC test if we force the lru map used by
    CPU0 to be allocated from a remote numa node:
```

w/o bpf map's NUMA hint

```
[root@localhost katran2]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11
node 0 size: 128685 MB
node 0 free: 116921 MB
node 1 cpus: 14 15 16 17 18 19 20 21 2
node 1 size: 129019 MB
node 1 free: 128247 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@localhost katran2]#
```

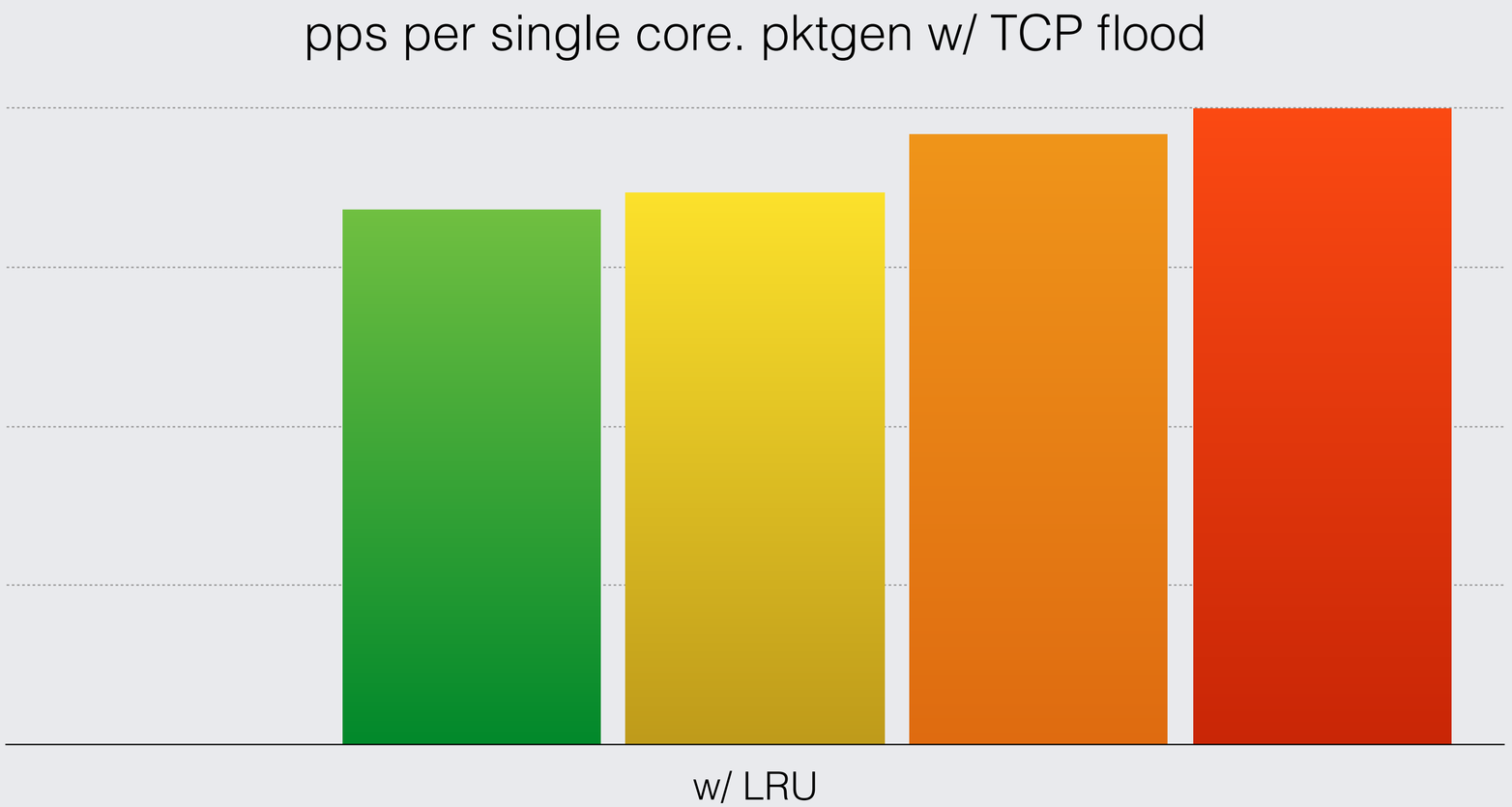
w/ bpf map's NUMA hint

```
[root@localhost katran2]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 1
node 0 size: 128685 MB
node 0 free: 121653 MB
node 1 cpus: 14 15 16 17 18 19 20 21
node 1 size: 129019 MB
node 1 free: 123515 MB
node distances:
node   0   1
  0:  10  20
  1:  20  10
[root@localhost katran2]#
```

XDP under flood

```
%Cpu1  :  0.0 us,  0.7 sy,  0.0 ni, 92.8 id,  0.0 wa,  0.0 hi,  6.6 si,  0.0 st
%Cpu2  :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.0 sy,  0.0 ni, 95.3 id,  0.0 wa,  0.0 hi,  4.7 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu16 :  0.0 us,  0.0 sy,  0.0 ni, 94.9 id,  0.0 wa,  0.0 hi,  5.1 si,  0.0 st
%Cpu17 :  0.0 us,  0.0 sy,  0.0 ni, 96.2 id,  0.0 wa,  0.0 hi,  3.8 si,  0.0 st
%Cpu18 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu19 :  0.0 us,  0.0 sy,  0.0 ni, 96.3 id,  0.0 wa,  0.0 hi,  3.7 si,  0.0 st
%Cpu20 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu21 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu22 :  0.0 us,  0.0 sy,  0.0 ni, 95.6 id,  0.0 wa,  0.0 hi,  4.4 si,  0.0 st
%Cpu23 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu24 :  0.0 us,  0.0 sy,  0.0 ni, 96.2 id,  0.0 wa,  0.0 hi,  3.8 si,  0.0 st
%Cpu25 :  0.0 us,  0.0 sy,  0.0 ni, 98.5 id,  0.0 wa,  0.0 hi,  1.5 si,  0.0 st
%Cpu26 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu27 :  0.0 us,  0.0 sy,  0.0 ni, 94.9 id,  0.0 wa,  0.0 hi,  5.1 si,  0.0 st
%Cpu28 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu29 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu30 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu31 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

Single core performance



+19% (+4%) perf

XDP + fragmentation

XDP can not fragment packets

2.1 Points of Presence

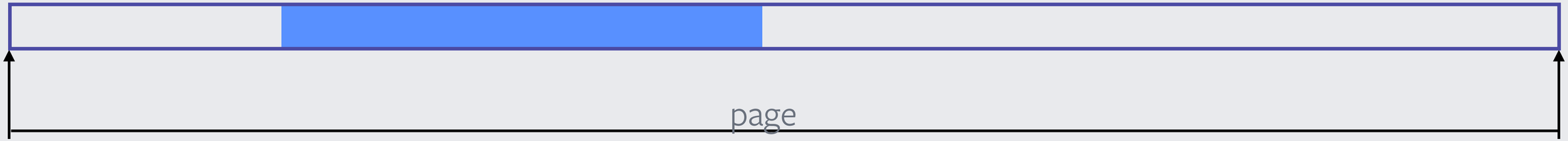
To help reduce user latencies, FACEBOOK has deployed Points of Presence (PoPs) in dozens of locations globally. A PoP serves users from racks of servers, which connect via intermediate aggregation switches (ASWs) to multiple peering routers (PRs), as seen in Fig-

Facebook's infrastructure consists of many geo-distributed data centers that host millions of servers.

XDP adjust tail

```
commit a579ccc445e1f53d3c6054aa69459010a16316b5
Author: Nikita V. Shirokov <tehnerd@tehnerd.com>
Date:   Tue Apr 17 21:42:13 2018 -0700

    bpf: adding bpf_xdp_adjust_tail helper

    Adding new bpf helper which would allow us to manipulate
    xdp's data_end pointer, and allow us to reduce packet's size
    indended use case: to generate ICMP messages from XDP context,
    where such message would contain truncated original packet.
```

XDP adjust tail



XDP adjust tail

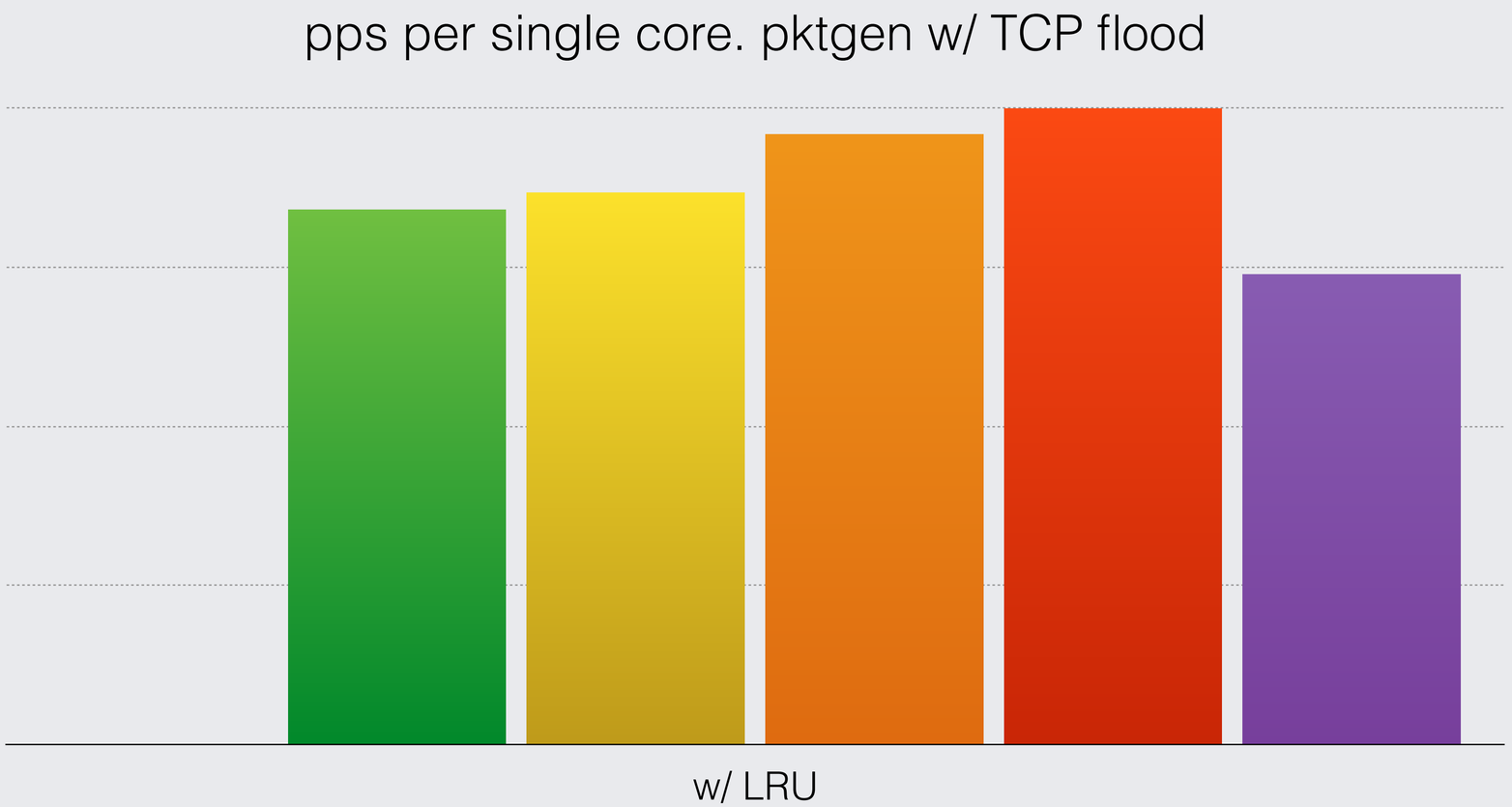
XDP adjust tail

XDP adjust tail

```
__attribute__((__always_inline__))
static inline int send_icmp_too_big(struct xdp_md *xdp,
                                    bool is_ipv6, int pckt_size) {

  int offset = pckt_size;
  if (is_ipv6) {
    offset -= ICMP6_TOOBIG_SIZE;
  } else {
    offset -= ICMP_TOOBIG_SIZE;
  }
  if(bpf_xdp_adjust_tail(xdp, 0 - offset)) {
    return XDP_DROP;
  }
  if (is_ipv6) {
    return send_icmp6_too_big(xdp);
  } else {
    return send_icmp4_too_big(xdp);
  }
}
```

Spectre v2

Single core performance

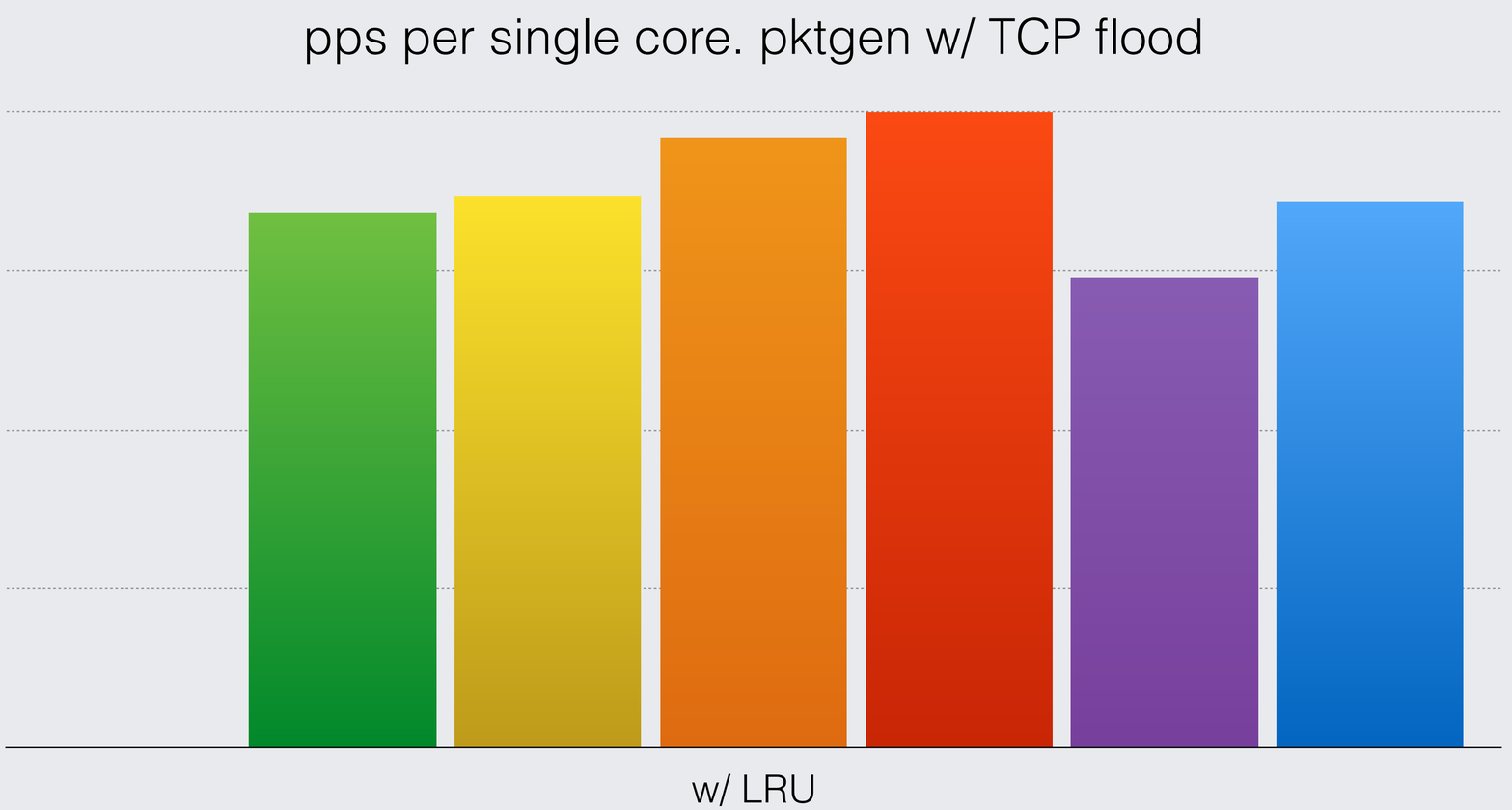


-12% perf

inline update

```
commit 09772d92cd5ad998b0d5f6f46cd1658f8cb698cf
Author: Daniel Borkmann <daniel@iogearbox.net>
Date:   Sat Jun 2 23:06:35 2018 +0200

    bpf: avoid retpoline for lookup/update/delete calls on maps

    While some of the BPF map lookup helpers provide a ->map_gen_lookup()
    callback for inlining the map lookup altogether it is not available
    for every map, so the remaining ones have to call bpf_map_lookup_elem()
    helper which does a dispatch to map->ops->map_lookup_elem(). In
    times of retpolines, this will control and trap speculative execution
    rather than letting it do its work for the indirect call and will
    therefore cause a slowdown. Likewise, bpf_map_update_elem() and
    bpf_map_delete_elem() do not have an inlined version and need to call
    into their map->ops->map_update_elem() resp. map->ops->map_delete_elem()
    handlers.
```

Single core performance



+2% perf

nice to have features

- RX/TX checksum offloading
- Crypto helpers
- loops (bounded)

i wish i had more time...

- bpf function calls
- bpf_prog_test_run for microbenchmarking
- verifier/clang improvements (-g + llvm-objdump -source is your friend!)
- XDP for syncookie generation

Questions?

facebook