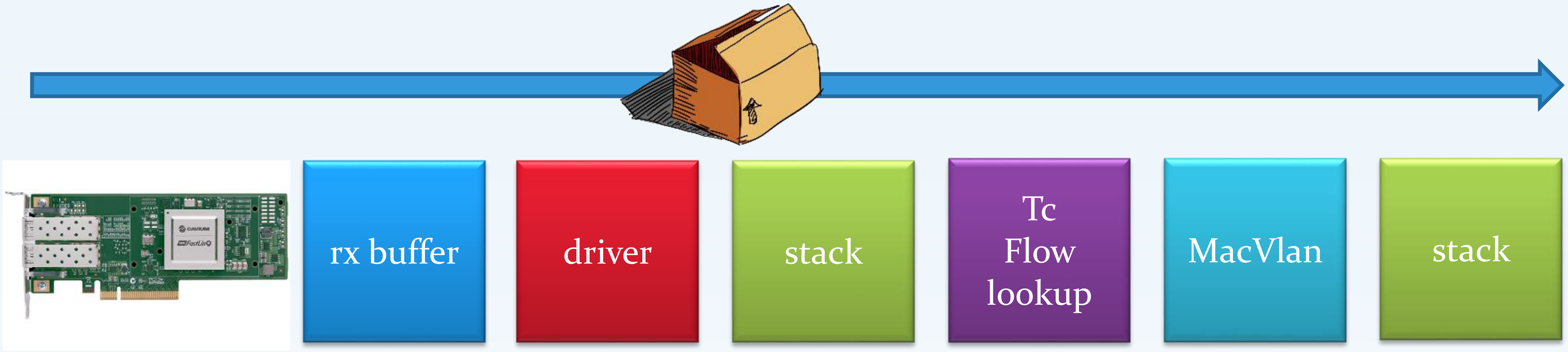
MacVlan and TC Offload in a container environment

Ariel Elior

Problem Statement

- Container environment
- Route traffic to specific containers based on Flow
- HW: Cavium 41000 ArrowHead with four 25G ports
  - 1x100G/2x25G tested as well
- Driver: `qed` and `qede` device drivers
- Requirements
  - 15Mpps
  - 64byte packets across all flows (no caching)
  - 15k/sec flow arrival rate
  - 16M flows
  - vlan push/pop/swap

MacVlan and TC receive flow – non offload



rx buffer

driver

stack

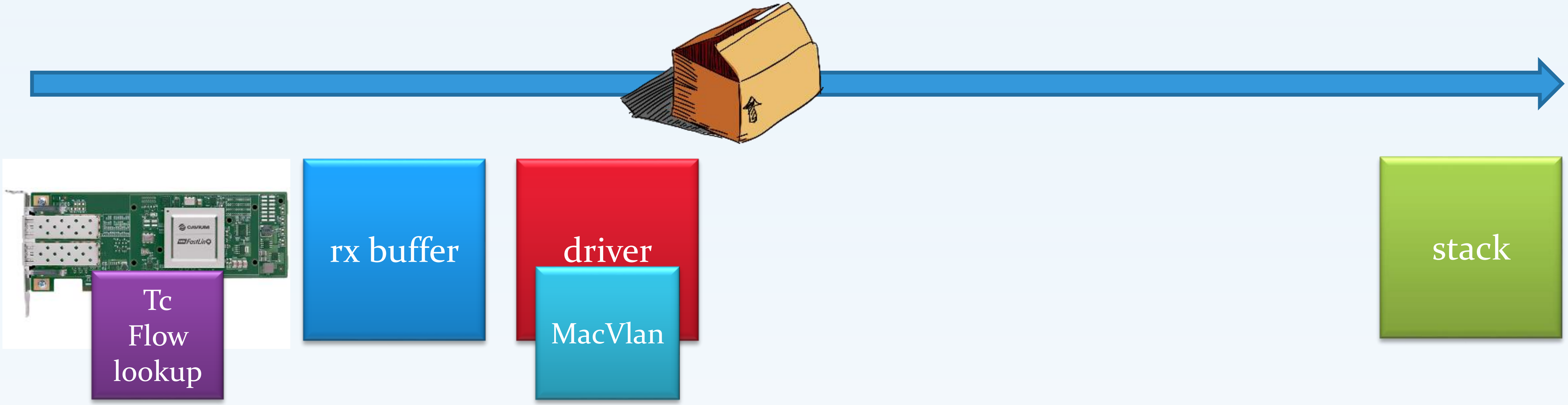
Tc
Flow
lookup

MacVlan

stack

1.6 Mpps
4k/sec Flow arrival rate

MacVlan and TC receive flow – offload



17.2 Mpps
4k/sec flow arrival rate
512k/sec "batch" flow arrival rate

CLI Syntax

Capabilities

```
ethtool -K p5p1 l2-fwd-offload on
ethtool -K p5p1 hw-tc-offload on
```

Create a MacVlan device

```
ip link add link p5p1 name mvlan_1 type macvlan
```

qdisc

```
tc qdisc add dev p5p1 ingress
```

Tc offload

```
tc filter add dev p5p1 protocol ip parent ffff: pref 0x2 flower skip_sw src_ip 0x0
dst_ip 192.168.50.100 action mirred egress redirect dev mvlan_1
```

Implementing MacVlan offload – lessons learned

- **Slowpath:**
- Refactor device (un)load flow (base vs macvlan)
  - Leveraged VF (un)load flow- qed(e) serves both VFs and PFs – unified (un)load flow
- Set rx mode classification
  - Kernel is missing an indication of the upper netdevice
- Statistics
  - Have to call macvlan_count_rx. Could cleanup fastpath by collecting these on demand

- **Fastpath:**
- Very minor changes
- Need to be able to find the net device from the queue structure – be careful about adding extra dereferences

Implementing TC offload – lessons learned

- Syntax parsing
  - useful built-ins
- Find destination device in upper devices list and translate to hw vport
  - Or child VFs network devices on HV
- Map action to device capabilities. Actions we used in this project:
  - mirred
  - drop
  - vlan push/pop/swap
- Statistics
  - HW limited amount of per-flow counter buckets in HW (thousands)
  - HW per action counters
  - HW assist Per flow stats – packets, bytes, last used

    can be implemented on host (hw can indicate per packet which flow it belongs to)

What’s ahead?

- Arrowhead/Bigbear - Actions we plan to add to current HW
  - Skb mark
  - Tunnel encap/decap
  - Pedit
    - Mac addr (NAT, load-balancing) , ip addr, tos, dscp, l4 ports, l2 priority, etc.

- Elbrus (2x100G, Pci Gen 4, 75Mpps) - features to add to next year’s HW
  - Wildcards
  - Reclassification
  - Full HW based stats

Thanks

- **Driver developers**
- Manish Chopra
- Shahed Shaik
- Yuval Mintz…

- **FW developers**
- Vitaly Mireyno
- Michael Shtramvaser

- **Architects**
- Zvika Perry

CAVIUM

Q&A