DPDK
SUMMIT
NORTH AMERICA

Transitioning Flow Based ethdev Ops to rte_flow

AJIT KHAPARDE - BROADCOM

Agenda

DPDK SUMMIT NORTH AMERICA

- ethdev ops
- rte_flow
- Convergence
- Challenges
- What next?

eth_dev_ops

- Provides device-specific functions for an Ethernet driver
- Device config, start, stop
- Promiscuous, multicast enable/disable
- Queue setup, start, stop, release
- VLAN, MAC filters
- Link get, set, update
- Stats/xstats get, reset
- RSS, MTU, Queue rate limit, LED, Timesync, EEPROM ….

- mirror_rule_set
- mirror_rule_reset
- rte_eth_dev_filter_ctrl

rte_flow_ops

- Generic mechanism to configure hardware to match ingress or egress traffic
- API to create, destroy, query and more..
- Flexible
  - Supports different headers and items for pattern match
  - Allows various actions
- But..
  - Provides possibilities which overlap with well known ethdev API
  - In some cases its even better..

The Convergence

- RTE_FLOW_ACTION_TYPE_SAMPLE action deprecated the rte_eth_mirror_rule_set
- rte_eth_dev_filter_ctrl became redundant and were deprecated
- And we can go further
  - mac_addr_remove, mac_addr_add, mac_addr_set
  - vlan_filter_set
  - udp_tunnel_port_add, udp_tunnel_port_del
  - Multicast, Promiscuous enable/disable
  - RSS

Challenges

DPDK SUMMIT NORTH AMERICA

- rte_flow API provides more config options
  - RSS
    - Hash level
    - Hash function selection
    - Pattern
- Deprecate ethdev ops?
  - But the guidelines are clear.
- State after life cycle of flow?

What Next?

- Deprecate ethdev ops?
  - But the guidelines are clear.
- Improve ethdev ops?
- Leave both and let application or user decide?
- Midground?
  - Port-level vs Per-flow
  - Port level configuration could remain out of rte_flow API?

DPDK

SUMMIT

NORTH AMERICA