Integrating AF_XDP into DPDK

XIAO LONG YE
INTEL
Agenda

- AF_XDP introduction
- AF_XDP PMD in DPDK
- Performance
- Future work
AF_XDP

- **Overview**
  - XDP’s user space interface
  - Use XDP program to trigger Rx path for selected queue
  - Zero Copy from DMA buffers to user space with driver support
  - Copy mode for non-modified drivers

- **Benefits**
  - Performance boost
  - Support all Linux network devices
AF_XDP

sfd = socket(PF_XDP, SOCK_RAW, 0);
buffs = calloc(num_buffs, FRAME_SIZE);
setsockopt(sfd, SOL_XDP, XDP_MEM_REG, buffs);
setsockopt(sfd, SOL_XDP, XDP_{RX|TX|FILL|COMPLETION}_RING, ring_size);
mmap(..., sfd, .......); /* map kernel rings */
bind(sfd, "/dev/eth0", queue_id, ....);
for (;;) {
    read_process_send_messages(sfd);
};
AF_XDP integrated into DPDK

• Motivation
  • Support DPDK in container usecase
  • Support non DPDK NICs
  • Reuse DPDK libraries for applications
  • Use of hugepages for performance
  • Linux kernel driver handles hardware
  • Better security and isolation
  • Utilize existing Linux network tools

• Goal
  • DPDK apps can run unmodified using AF_XDP interface
  • Performance is on par with kernel xdpsock sample
AF_XDP integrated into DPDK
AF_XDP PMD

- Implemented by vdev
- Create af_xdp socket, umem, fill/completion/Rx/Tx rings
- Features:
  - Multi-queues
  - Zero copy between mbuf and umem

```
./build/app/testpmd -I 5,6,7 -n 4 --log-level=pmd.net.af_xdp:info --no-pci --vdev net_af_xdp0,iface=ens786f1,start_queue=0,queue_count=3 --i --rxq=3 --txq=3
```
ZERO-COPY (through external mbuf)

- Use external buf to hold the payload

- MP HDROBJ size (64 bytes)
- *buf_addr
- struct rte_mbuf (128 bytes)
- mbuf headroom (128 bytes)

- 2K aligned address

- rtce_mem_pool
- Mbuf_0
- Mbuf_1
- Mbuf_2
- Mbuf_3
- Mbuf_4
- ...

- umem data

- xsk_umem
- 2K Chunk
- 2K Chunk
- 2K Chunk
- 2K Chunk
- 2K Chunk
- 2K Chunk
Performance

- DPDK 19.05 + ZC patch
- Broadwell E5 2660 @ 2.0 GHz
- IXIA load generator blasting at full 40 Gbit/s
- Intel XL710 card (40G, i40e driver)

![Graph showing throughput for different data sizes and model types.]

- l2fwd

Throughput

- xdpsock
- i40e pmd
- AF_XDP pmd
- AF_XDP pmd zero copy
Future work

- Flexible umem chunk size and alignment
- Optimize AF_XDP pmd in container scenario
Thanks