Interworking with the Linux Kernel

Ferruh Yigit
Intel Shannon
DPDK Summit Userspace - Dublin- 2016
Who am I

- DPDK KNI maintainer
- Working for Intel
What is Kernel Network Interface (KNI)?

Elements of KNI in DPDK

Latest changes in KNI

NOT implementation / usage details, but more discussion

Discussion

Use cases

Alternative approaches for Interworking with the Linux

Pros and Cons of KNI

Future of KNI
What is KNI?

- Existing implementation in DPDK for interworking with the Linux
- Allows an interface with the kernel network stack.
  - Data
  - Control
- For exception data traffic
  - Example “ping”

http://dpdk.org/doc/guides/prog_guide/kernel_nic_interface.html
Quick introduction

DPDK APP

kthread → driver → STACK

RX
TX

DPDK APP

librte_kni

BSD socket app

NIC

kthread

STACK

rte_kni.ko

kernel

user

hardware
Elements of KNI in DPDK

- Kernel module for Linux [rte_kni.ko]
  - lib/librte_eal/linuxapp/kni/
  - lib/librte_eal/linuxapp/kni/ethtool/
- Library [librte_kni.[a,so]]
  - lib/librte_kni/
- Sample app [kni]
  - examples/kni/
    - make -C examples/kni
    - ./examples/kni/build/kni --p0x3 -P --config "(0,1,2,5),(1,3,4,21)"
Latest changes in KNI (DPDK v16.11)

- Update for latest environment
  - kernel version (4.8), gcc version 6.1, OS RHEL 7.3 and Centos
- Remove single mempool restriction
- Bind kthread to specific core for single _thread
- Support chained mbufs
- Syntax cleanup
- KNI PMD [?]
Current implementation requires more maintenance effort!

Only supports igb and ixgbe drivers.

Is it really used by community or can we remove it?

Use cases

Kernel Control Path

- Remove drivers from KNI

- Virtual network driver, netlink to DPDK driver

- Still out of kernel module ...
Faster comparing other methods to get packet from userspace to kernel
  - Eliminates system calls, userspace – kernelspace copy

Performance concerns?

Use cases

Kernel Data Path

KNI VHOST (KNI as Kernel vHost backend)
  - Any user? Can we remove?

BSD implementation?
Use cases

Pros and Cons of KNI

Alternative approaches

Future of KNI

- Depends on what community wants, use cases

- ....
Things to improve in KNI

- Out of tree kernel module
  - Is it really problem?
  - Operating System Vendors?
- Performance
  - Is performance real reason why KNI is used?
Alternative Solutions

- Tun/Tap
  - Recently tap PMD patch sent
- af_packet
- virtio-user + vhost-net
- Bifurcated driver
Future of the KNI

- Remove ethtool support?
- Remove KNI VHOST?
- What to do with out of tree kernel module?
- Switch completely to an alternative approach?
- Any improvement on library or sample app?
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this
document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of
merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from
course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information
provided here is subject to change without notice. Contact your Intel representative to obtain the latest
forecast, schedule, specifications and roadmaps.

Intel technologies’ features and benefits depend on system configuration and may require enabled
hardware, software or service activation. Performance varies depending on system configuration. No
computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more
at intel.com.

What’s Inside logo are trademarks of Intel. Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.
Questions?

Ferruh Yigit
ferruh.yigit@intel.com