DPDK
DATA PLANE DEVELOPMENT KIT

THE LINUX FOUNDATION

DPDK SUMMIT CHINA 2017

主办方： intel

参与方： 腾讯云 ZTE 美团云 Panabit® 太一星晨 Balance Your Networks UnitedStack有云 云杉网络 Yunshan Networks

协办方： SDNLAB 专注网络创新技术

视频支持方： IT大咖说

DPDK
DATA PLANE DEVELOPMENT KIT

THE LINUX FOUNDATION

Optimized Packet Distribution Library

Liang Ma, INTEL

主办方： intel

参与方： 腾讯云 ZTE 美团云 Panabit 太一星晨 云杉网络

协办方： SDNLAB 专注网络创新技术　　视频支持方： IT大咖说

LEGAL DISCLAIMER

- 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.
- © 2017 Intel Corporation. Intel, the Intel logo, Intel. Experience What's Inside, and the Intel. Experience 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.
-

Agenda

- **Problem** – How to distribute workload across cores
- **Solution** – OPDL high level design
- **Characteristics of OPDL** – low latency, in order, asynchronous
- **Simple Example** - IPSEC in-bound processing
- **Next Steps** – Future Work

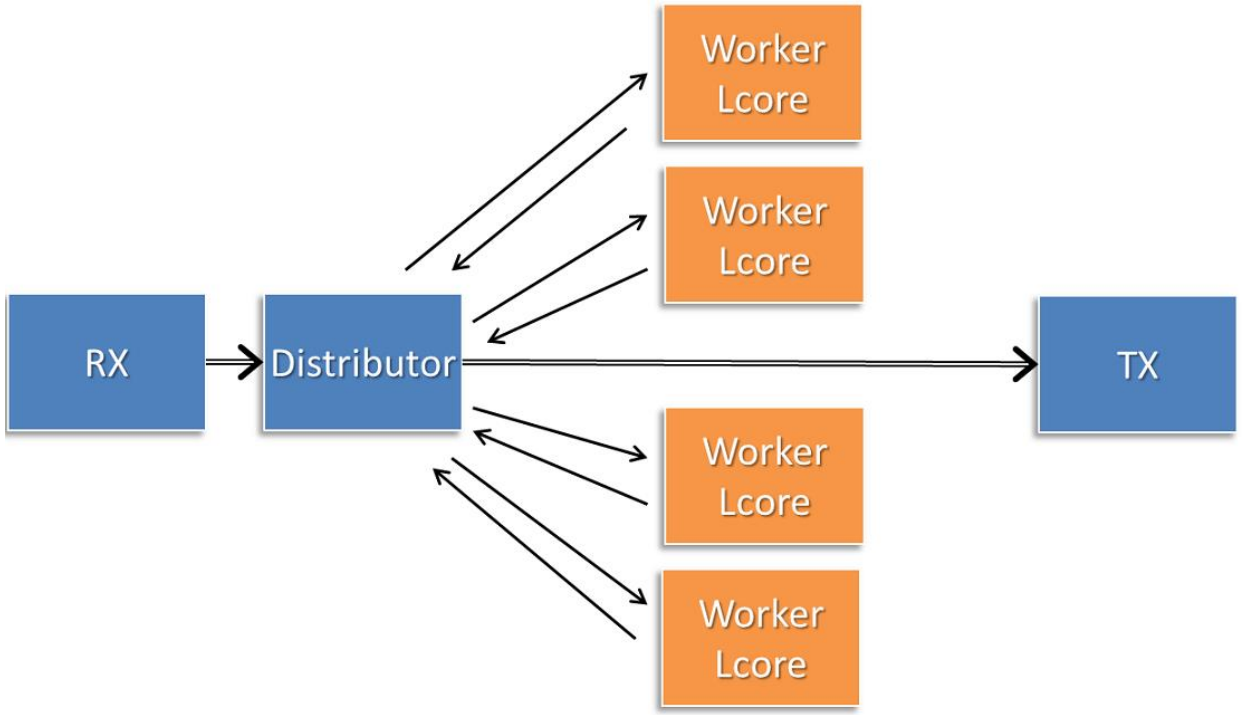
Problem and Challenge

- **Stringent latency and high throughput**
- **Minimizing cross core costs**
- **Re-Order, Asynchronous**
- **Centralized distributor**
- **Scalability, Flexibility**

Flow Distribution Type

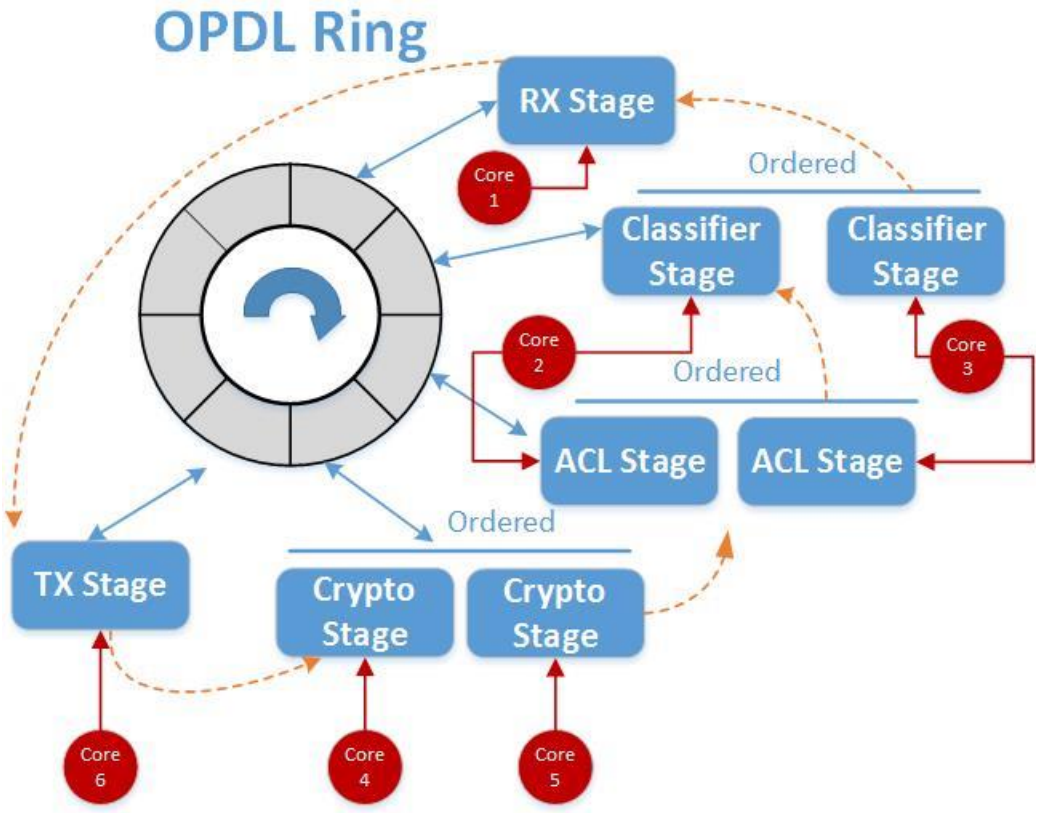
- Parallel
  - Packets from same flow can be distributed to multiple cores
  - Without Ordering
- Ordered
  - Packets from same flow can be distributed to multiple cores
  - With Ordering
- Atomic
  - Only one packet from same flow is processed at a time

DPDK Packet Distributor



- Very efficient for high volume small packets with synchronous work load
- Centralized distributor, Dedicated Core
- Round Robin to worker cores
- Atomic, and Parallel(un-ordered)
- Buffer Pointer will be send back to Distributor
- Asynchronous operation can be very complex

Optimized Packet Distributor

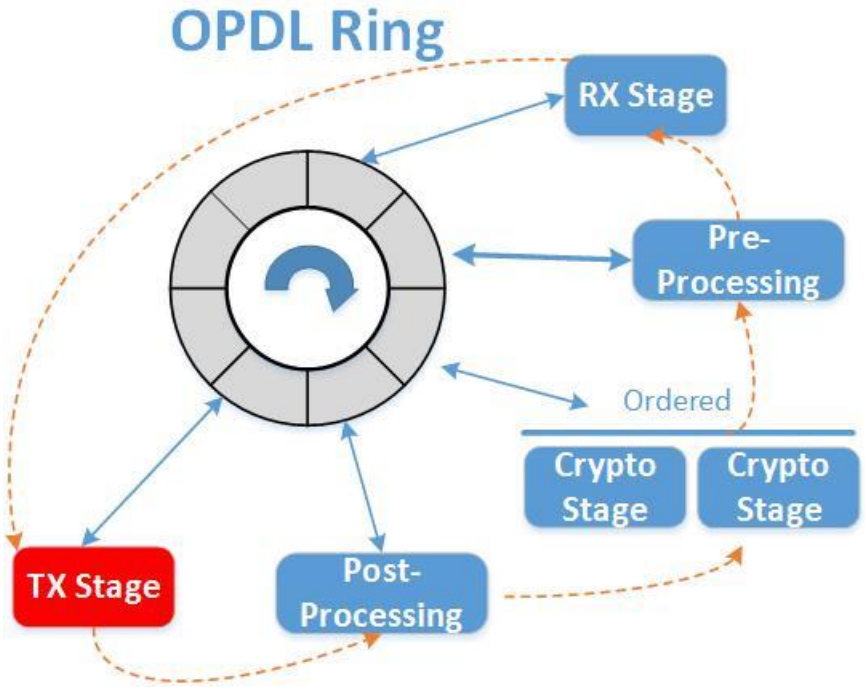


- No multiple queue cost, decentralized distributer
- Stage topology is configurable and extremely flexible.
- All packets are maintained in order
- Using meta-data to synchronize stages within an application
- Support ATOMIC/ORDERED method
- Stage instance and Core mapping is flexible.
- Support asynchronous device seamless.(E.g. Crypto Dev)

OPDL APIs

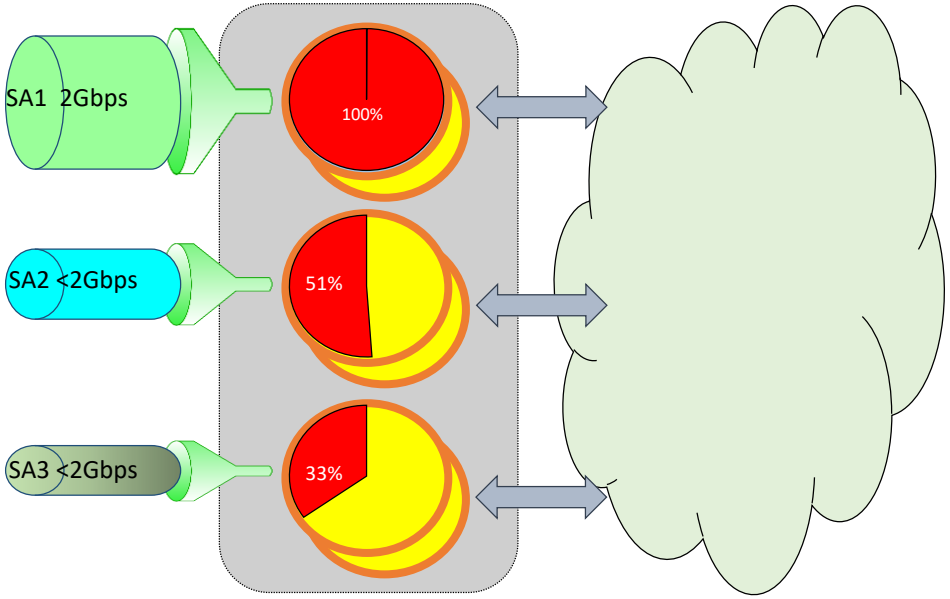
- OPDL_init()
  - Invocate stage initialization handler call back function
- OPDL_claim()
  - Claim available slot from OPDL Ring
- OPDL_Processing()
  - Invocate stage packet processing handler call back function
- OPDL_disclaim_n()
  - Can do partial disclaim to handler asynchronies device

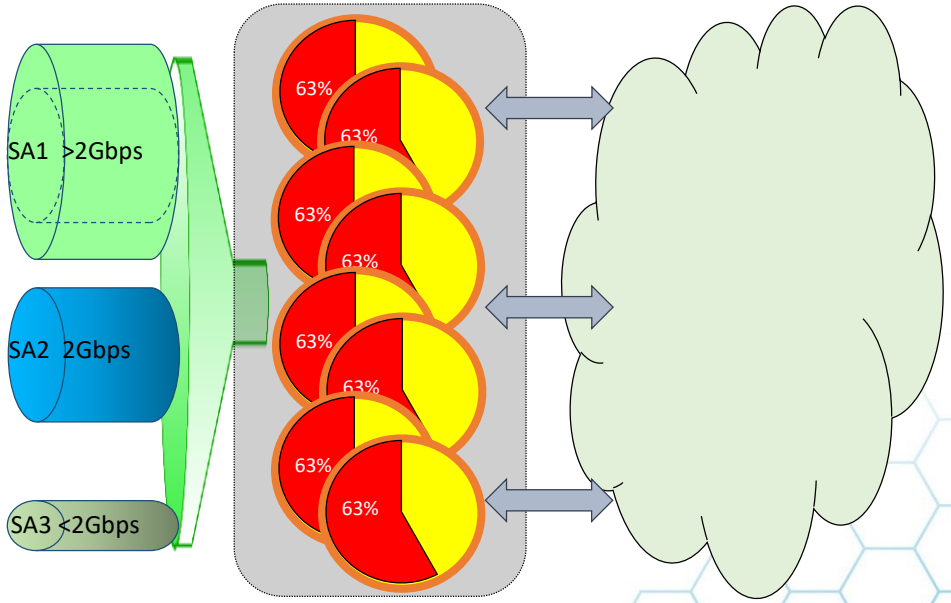
Example : Simple IPSEC In-bound Processing



- RX is atomic stage. RX will poll packet from port/queue and put packet into the OPDL Ring slot.
- Pre Processing stage is atomic. SA look-up, esn processing, put sequence number into meta data
- Decryption Stage is Ordered, there are 2 instances, processing the packet based on modulo.
- Post Processing will do de capsule first then apply Acl rules against the decrypted packet

Example : Single large SA & Multiple unevenly distributed SA's





RTC

- Single SA - Load is distributed according to SA throughput(not evenly distributed across cores)
- External Load Balancer required

OPDL

- Multiple SA - Can scale to max capacity (function of #cores)
- Load is distributed evenly across all available cores

Future Work

- **Looking for opportunity of up-streaming OPDL to DPDK mainstream repository**
- **Optimization research for multiple workload**

Acknowledgement

- I wish to express my sincere gratitude to all team member
  - John, Browne
  - Chris, Macnamara
  - Tomasz, Kantecki
  - Peter, Mccarthy
  - Mairtin, O'loingsigh
  - Sean, Harte

Q&A

Thanks!!

欢迎关注**DPDK开源社区**