DPDK ARCHITECTURE AND ROADMAP DISCUSSION

KANNAN BABU RAMIA, INTEL
DEEPAK KUMAR JAIN, INTEL
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.
• The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
• Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm
• Intel and the Intel 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.
• Copyright © 2017, Intel Corporation. All rights reserved.
• Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804
• Mileage may vary Disclaimer: Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Test and System Configurations: Estimates are based on internal Intel analysis using at least Data Plane Development Kit ipSec sample application on Intel(R) Xeon(R) CPU E5-2658 v4@ 2.30GHz with atleast using Intel(R) Communications Chipset(s) 8955 with Intel(R) QuickAssist Technology.
Agenda

- Key trends in network transformation
- DPDK role
- DPDK Architecture
- Multi Architecture/ Multi vendor support
- Open source projects using DPDK
- DPDK Roadmap
- Open Questions
End-to-end network infrastructure

- Macro Base Station
- Small Cell
- Router
- Core
- Backbone
- Data Center Network
- Switch

WORKLOAD CONVERGENCE
NETWORK VIRTUALIZATION
END-TO-END TRANSFORMATION

Copyright © 2017 Intel Corporation. All rights reserved.
DPDK Generational Performance Gains

IPV4 L3 Forwarding Performance of 64Byte Packets

<table>
<thead>
<tr>
<th>Year</th>
<th>L3Fwd Performance (MPPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>55 Gbps</td>
</tr>
<tr>
<td>2011</td>
<td>80.1 Gbps</td>
</tr>
<tr>
<td>2012</td>
<td>164.9 Gbps</td>
</tr>
<tr>
<td>2014</td>
<td>255 Gbps</td>
</tr>
<tr>
<td>2015</td>
<td>279.9 Gbps</td>
</tr>
<tr>
<td>2016</td>
<td>*346.7 Gbps</td>
</tr>
</tbody>
</table>

*System Configuration

Hardware
- Platform: SuperMicro® - X10DRX
- CPU: Intel® Xeon® E5-2658 v4 Processor
- Chipset: Intel® C612 chipset
- Sockets: 2
- Cores per Socket: 14 (28 threads)
- LL CACHE: 30 MB
- QPI/DMI: 9.6GT/s
- PCIe: Gen3x8
- MEMORY: DDR4 2400 MHz, 1Rx4 8GB (total 64GB), 4 Channel per Socket
- NIC: 10 x Intel® Ethernet CNA XL710-QDA2PCI-Express Gen3 x8 Dual Port 40 GbE Ethernet NIC (1x40G/card)
- NIC Mbps: 40,000
- BIOS: BIOS version: 1.0c (02/12/2015)

Software
- OS: Debian® 8.0
- Kernel version: 3.18.2
- Other: DPDK 2.2.0

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.
VM to VM communication between Virtual Network Functions, needs multi-100s of Gbps throughput

Intel® QuickAssist Technology Crypto Accelerator

VNF – Virtual Network Functions

VEB – Virtual Embedded Bridge
TEP – Tunnel End Point
ACL – Access Control List
GFT – Generic Flow Table
VNF – Virtual Network Functions
DPDK Architecture

DPDK Fundamentals

- Implements run-to-completion and pipeline models
- No scheduler - all devices accessed by polling
- Supports 32-bit and 64-bit OSs with and without NUMA
- Scales from Intel® Atom™ to Intel® Xeon® processors
- Number of cores and processors is not limited
- Optimal packet allocation across DRAM channels
- Use of 2M & 1G hugepages and cache aligned structures
- Uses bulk concepts - processing ‘n’ packets simultaneously
Multi-Architecture/Multi-Vendor Support

2014
First non-IA contributions.

2015
Non-Intel NIC support.

2016
Significant ARM vendor engagement.

2017
SoC enhancements. Non-Intel crypto.

CPU Architectures

POWER 8
IBM

TILE-Gx
EZchip Technologies
Rehitech
CAVIUM

ARM v7/v8

Enhanced ARM Support
CAVIUM
NXP
Rehitech

SoC Enhancements
NXP

2014
BNX2X
MLX4/MLX5

2015
ThunderX PMD

2016
ThunderX PMD

2017
ARMv8 Crypto
OcteonTX
LiquidIO

Poll Mode Drivers

ENIC
Cisco

BNX2X
Broadcom

MLX4/MLX5
Mellanox Technologies

ThunderX PMD
CAVIUM
BNXT

BNXT

SFC

ARv8 Crypto
OcteonTX
LiquidIO

AVP

DPAA2
NXP

ENIC
Cisco

BNX2X
Broadcom

MLX4/MLX5
Mellanox Technologies

ThunderX PMD
CAVIUM
BNXT

BNXT

SFC

ARv8 Crypto
OcteonTX
LiquidIO

AVP
Q1’17 (v 17.02)

• Added Elastic Flow Distributor library (rte_efd).
• Added generic flow API (rte_flow).
• Added support for representing buses in EAL.
• Added APIs for MACsec offload support to the ixgbe PMD.
• Added VF Daemon (VFD) for i40e. – EXPERIMENTAL.
• virtio-user with vhost-kernel as another exceptional path.
• Added ARMv8 crypto PMD and updates to QAT, AESNI-MB PMDs.

Released
DPDK Roadmap

Q2’17 (v 17.05)

- Added Eventdev PMD.
- Added event driven programming model library (rte_eventdev).
- Added bit-rate calculation, latency stats and information metric library.
- Kept consistent PMD batching behaviour.
- Added VFIO hotplug and vmxnet3 version 3 support.
- Added MTU feature support to Virtio and Vhost.
- Added interrupt mode support for virtio-user.
DPDK Roadmap

Q2’17 (v 17.08)

• Generic QoS API
• Cryptodev Multi-Core SW Scheduler
• Generic Receive Offload
• Generic Flow Enhancements
• VF Port Reset for IXGBE
• API to Configure Queue Regions for RSS
• Support for IPFIX
OPEN QUESTIONS?

- What is missing from DPDK?
- What are the major pain-points in using DPDK?
- What can be improved in DPDK? Build process? Logging?
- What are the big performance bottlenecks?
- Working with Kernel?
THANK YOU
DPDK Acceleration Enhancements

**DPDK Framework**
- VNF Apps
- Future features
- Event based program models
- Video Apps
- IPSec
- DPI Hyperscan
- Proxy Apps, ...

**DPDK API**
- Crypto
- Programmable Classifier/Parser
- Compression
- Future accelerators

**Legacy DPDK**
- Core Libraries
- Platform
- Packet Access (PMD)

**Future accelerators**
- External mempool manager
- SoC PMD
- SoC HW
- SOC model

**Future features**
- Event based program
- Threading Models
- lthreads, ...

**DPDK example apps**
- vSwitch
- OVS, Lagopus, ...
- Traffic Gens
- Pktgen, T-Rex, Moongen, ...

**DPDK**
- AEAL
- MALLOC
- MBUF
- MEMPOOL
- RING
- TIMER
- CMNULNE
- KVARGS

**Core Libraries**
- KNI
- POWER
- IVSHMEM
- JOBSTAT

**Platform**
- METER
- SCHED
- QoS

**Packet Access (PMD)**
- ETHDEV
- e1000
- af, pkt
- igbe
- bonding
- fm10k
- vmxnet3
- virtio
- xenvirt
- ring
- emic
- mlx4
- memnic
- others

**Classify**
- LPM
- Classification
- ACL
- TABLE

**Utilities**
- CMDLINE
- JOBSTAT
- KVARGS
- REORDER
- TABLE

**Legacy DPDK**
- Network Stacks
  - libUNS, mTCP, SeaStar, libuinet, TLDK, ...

**Future DPDK**
- Crypto
- Programmable Classifier/Parser
- Compression

**EAL**
- MALLOC
- MBUF
- MEMPOOL
- RING
- TIMER
- CMNULNE
- KVARGS

**Core Libraries**
- KNI
- POWER
- IVSHMEM
- JOBSTAT

**Platform**
- METER
- SCHED
- QoS

**Packet Access (PMD)**
- ETHDEV
- e1000
- af, pkt
- igbe
- bonding
- fm10k
- vmxnet3
- virtio
- xenvirt
- ring
- emic
- mlx4
- memnic
- others

**Classify**
- LPM
- Classification
- ACL
- TABLE

**Utilities**
- CMDLINE
- JOBSTAT
- KVARGS
- REORDER
- TABLE

**Future DPDK**
- Crypto
- Programmable Classifier/Parser
- Compression

**Legacy DPDK**
- Network Stacks
  - libUNS, mTCP, SeaStar, libuinet, TLDK, ...

**Future accelerators**
- External mempool manager
- SoC PMD
- SoC HW
- SOC model

**Future features**
- Event based program
- Threading Models
- lthreads, ...

**DPDK example apps**
- vSwitch
- OVS, Lagopus, ...
- Traffic Gens
- Pktgen, T-Rex, Moongen, ...

**DPDK**
- AEAL
- MALLOC
- MBUF
- MEMPOOL
- RING
- TIMER
- CMNULNE
- KVARGS

**Core Libraries**
- KNI
- POWER
- IVSHMEM
- JOBSTAT

**Platform**
- METER
- SCHED
- QoS

**Packet Access (PMD)**
- ETHDEV
- e1000
- af, pkt
- igbe
- bonding
- fm10k
- vmxnet3
- virtio
- xenvirt
- ring
- emic
- mlx4
- memnic
- others

**Classify**
- LPM
- Classification
- ACL
- TABLE

**Utilities**
- CMDLINE
- JOBSTAT
- KVARGS
- REORDER
- TABLE
Packet Processing Kernel vs. User Space

Kernel Space Driver

User Space
- Applications
- System Calls
- Driver

Kernel Space
- NIC
- CSR Rings
- CSRs

Memory (RAM)
- Packet Data
- Socket Buffers (skb's)
- Descriptors
- Configuration

Interrupts
- DMA

Benefit #1
- Removed Data copy from Kernel to User Space

Benefit #2
- No Interrupts

Benefit #3
- Network stack can be streamlined and optimized

User Space Driver with Zero Copy

User Space
- DPDK Enabled App
- Socket Buffers (mbuf's)
- Descriptors
- Configuration

Kernel Space
- System Calls
- Stack
- UIO Driver
- Descriptors
- Configuration

Memory (RAM)
- DMA

NIC
- Descriptor Rings
- CSR Rings
DPDK IN-DEPTH
PCIe* Connectivity and Core Usage

Using run-to-completion or pipeline software models

**Run to Completion Model**
- I/O and Application workload can be handled on a single core
- I/O can be scaled over multiple cores

**Pipeline Model**
- I/O application disperses packets to other cores
- Application work performed on other cores

**Can handle more I/O on fewer cores with vectorization**
Core Components Architecture

X uses Y

- **rte_timer**: Timer facilities. Based on HPET interface that is provided by EAL.
- **rte_mempool**: Supports a pool of objects held in a ring with lockless lists available per logical core to support fast bulk enqueue/dequeue of the objects (typically mbufs).
- **rte_mbuf**: Support of packet buffers for IP traffic and message data.
- **rte_ring**: Fixed-size lockless FIFO for storing objects or to allow communications between cores.
- **rte_eal + libc**: Environment abstraction layer: application loading, memory allocation, time reference, PCI access and logging.
- **rte_malloc**: Allocation of named memory zones constructed from hugepages.
Intel® DPDK allocates packet memory equally across 2, 3, 4 channels. Aligned to have equal load over channels.

Rings for cached buffers
Per core lists, unique per lcore. Allows packet movement without locks

Intel® DPDK allocates packet memory equally across 2, 3, 4 channels. Aligned to have equal load over channels.

Stacks available from Eco Systems
Run to completion model on each core used

“Open-source Stack” (NetBSD)

DPDK model
High Performance Components of DPDK

- Environment Abstraction Layer
  - Abstracts huge-page file system, provides multi-thread and multi-process support, etc.

- Memory Manager
  - Responsible for allocating pools of objects in memory. A pool is created in huge page memory space and uses a ring to store free objects. It also provides an alignment helper to ensure that objects are padded to spread them equally on all DRAM channels.

- Buffer Manager
  - Reduces by a significant amount the time the operating system spends allocating and de-allocating buffers. The Intel® DPDK pre-allocates fixed size buffers which are stored in memory pools.

- Queue Manager
  - Implements safe lockless queues, instead of using spinlocks, that allow different software components to process packets, while avoiding unnecessary wait times.

- Flow Classification
  - Provides an efficient mechanism which incorporates Intel® Streaming SIMD Extensions (Intel® SSE) to produce a hash based on tuple information so that packets may be placed into flows quickly for processing, thus greatly improving throughput.
EAL Initialization in a Linux Environment
Ethernet Device Framework

Application (calls rte_ethdev API)

--- (Port ID, Queue ID) ---

rte_eth_rx_burst(...)   rte_eth_tx_burst(...)  (PMD specific context)

rrc_recv_pkts(...)     rrc_xmit_pkts(...)     (Descriptors)

--- (Descriptors) ---

Network H/W
1. Initialization
   - Init Memory Zones and Pools
   - Init Devices and Device Queues
   - Start Packet Forwarding Application

2. Packet Reception (RX)
   - Poll Devices’ RX queues and receive packets in bursts
   - Allocate new RX buffers from per queue memory pools to stuff into descriptors

3. Packet Transmission (TX)
   - Transmit the received packets from RX
   - Free the buffers that we used to store the packets

30,000 ft overview of packet flow