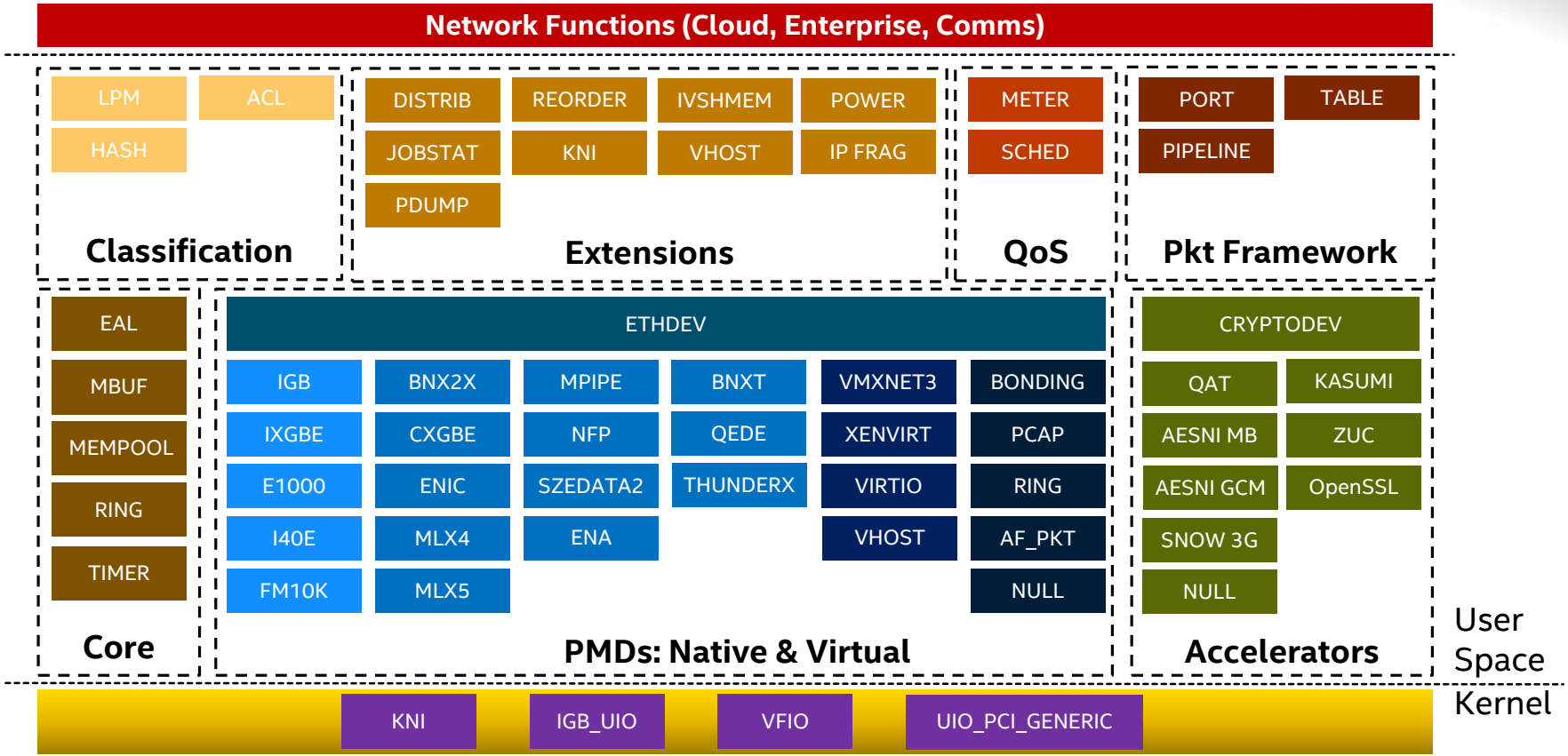
워크로드 통합 핵심 도구 - 인텔 DPDK (Data Plane Development Kit)

박준식
인텔 SSG (Software and Services Group)

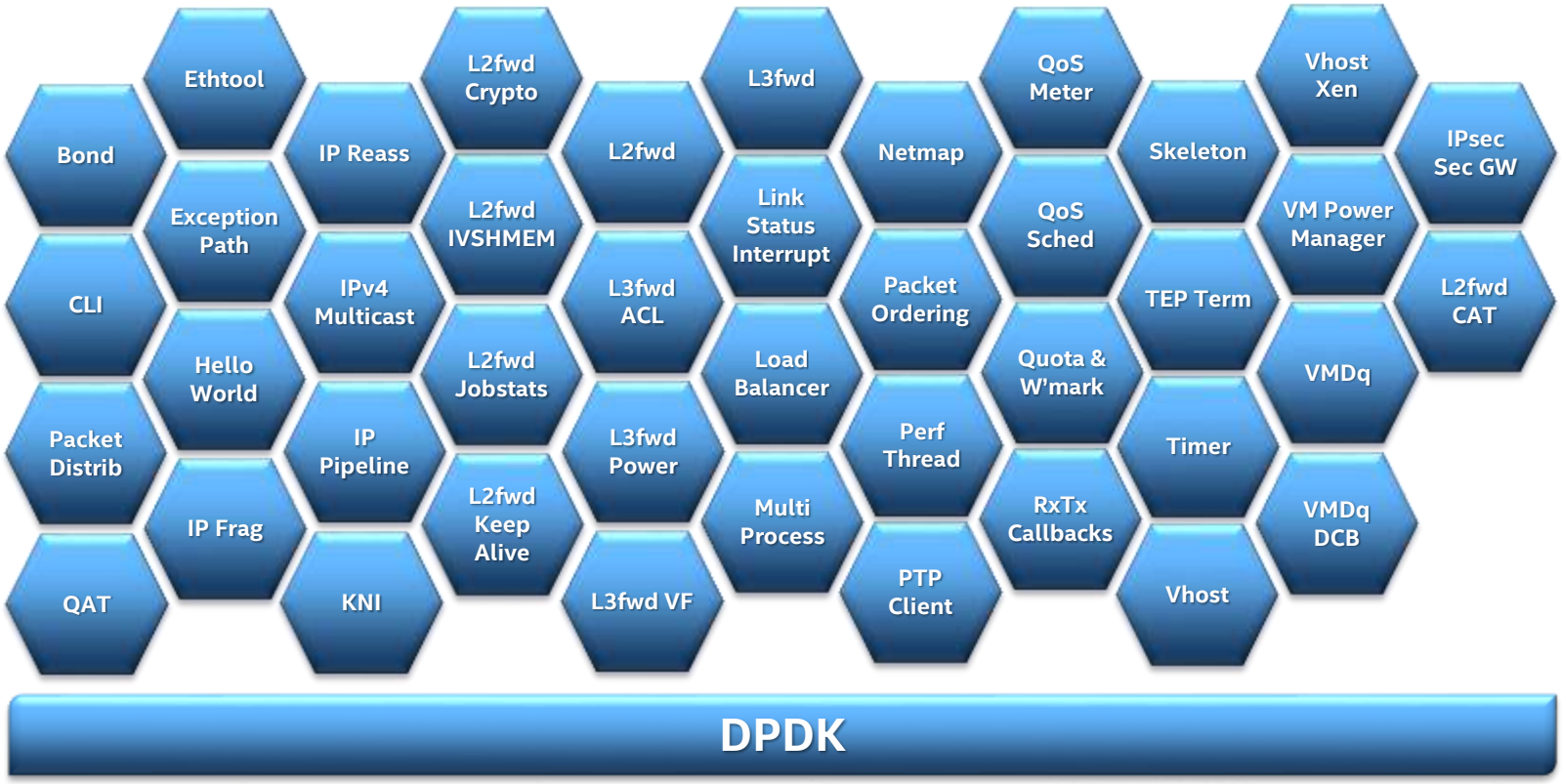
2017년 4월

DPDK OVERVIEW

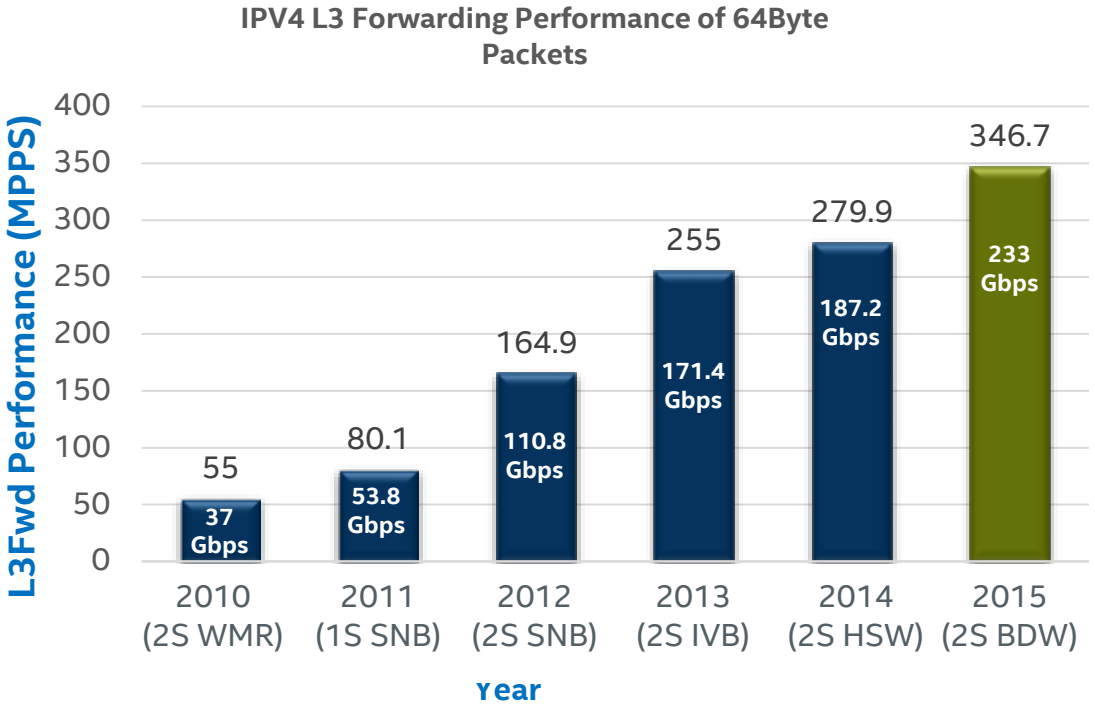
DPDK Framework



DPDK Sample Apps



DPDK Generational Performance Gains



Broadwell EP System Configuration	
Hardware	
Platform	SuperMicro® - X10DRX
CPU	Intel® Xeon® Processor E5-2658 v4
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	DPDK2.2.0

DPDK vs OpenDataPlane

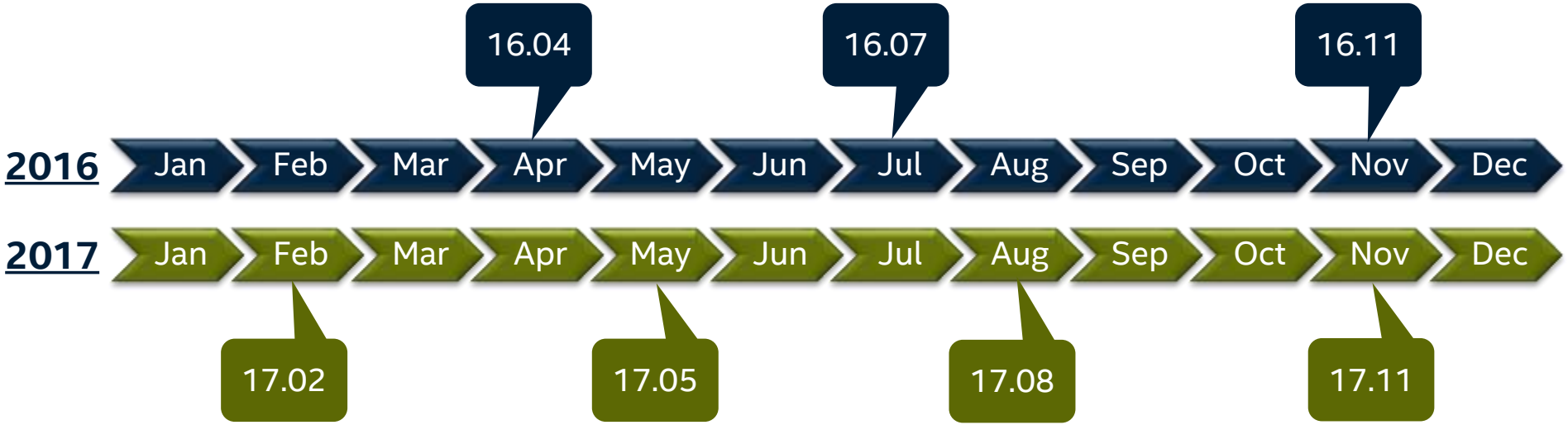
	DPDK	OpenDataPlane.org
Project Activity	Very High Activity	High Activity
Contributors (All Time)	**327**	67
Commits (All Time)	**6084**	3,297
Initial Commit	**over 5 years ago**	over 3 years ago
Contributors (Past 12 Months)	**213**	38
Commits (Past 12 Months)	**2,667**	1,509

Source: https://www.openhub.net/p/_compare?project_0=DPDK&project_1=OpenDataPlane, 25th November, 2016

ROADMAP

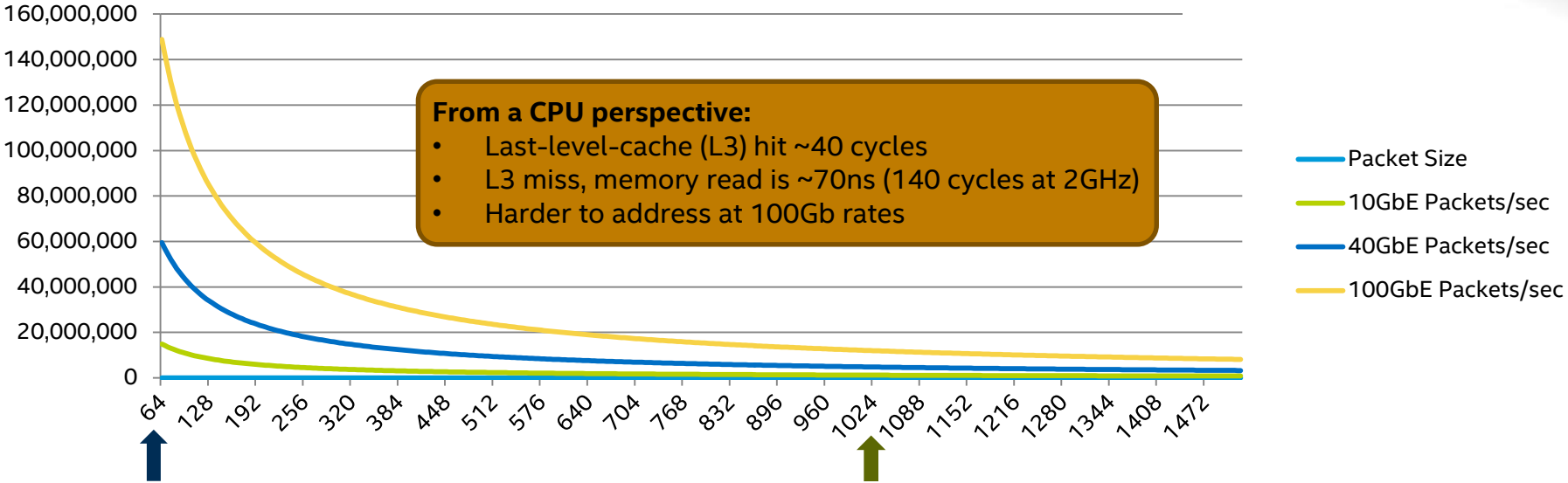
Future Releases

- DPDK releases use the Ubuntu numbering scheme of YY.MM.
- We'll transition gradually from 3 major releases per year to 4.
- Frequency and dates of releases will be fixed from 2017 onwards.



DPDK IN-DEPTH

The Problem Statement



Packet Size	64 Bytes
40G packets/second	59.5 million each way
Packet arrival interval	16.8 ns
2 GHz clock cycles/packet	33 cycles

Typical Network Infrastructure Packet Size

Packet Size	1024 Bytes
40G packets/second	4.8 million each way
Packet arrival interval	208.8 ns
2 GHz clock cycles/packet	417 cycles

Typical Server Packet Size

High Performance Challenges

The system can't keep up with the number of interrupts for packet Rx:

- Switch from an interrupt-driven network device driver to a polled-mode driver.

The Linux scheduler causes too much overhead for task switches:

- Bind a single software thread to a logical core.

Memory and PCIe access is really slow compared to CPU operations:

- Process a bunch of packets during each software iteration and amortize the access cost over multiple packets.

Data doesn't seem to be near the CPU when it needs to be:

- For memory access, use HW or SW controlled prefetching. For PCIe access, use Data Direct IO to write data directly into cache.

Access to shared data structures is a bottleneck:

- Use access schemes that reduce the amount of sharing (e.g. lockless queues for message passing).

Page tables are constantly evicted (DTLB Thrashing):

- Allow Linux to use Huge Pages (2MB, 1GB)

Achieving Performance – Silicon Features

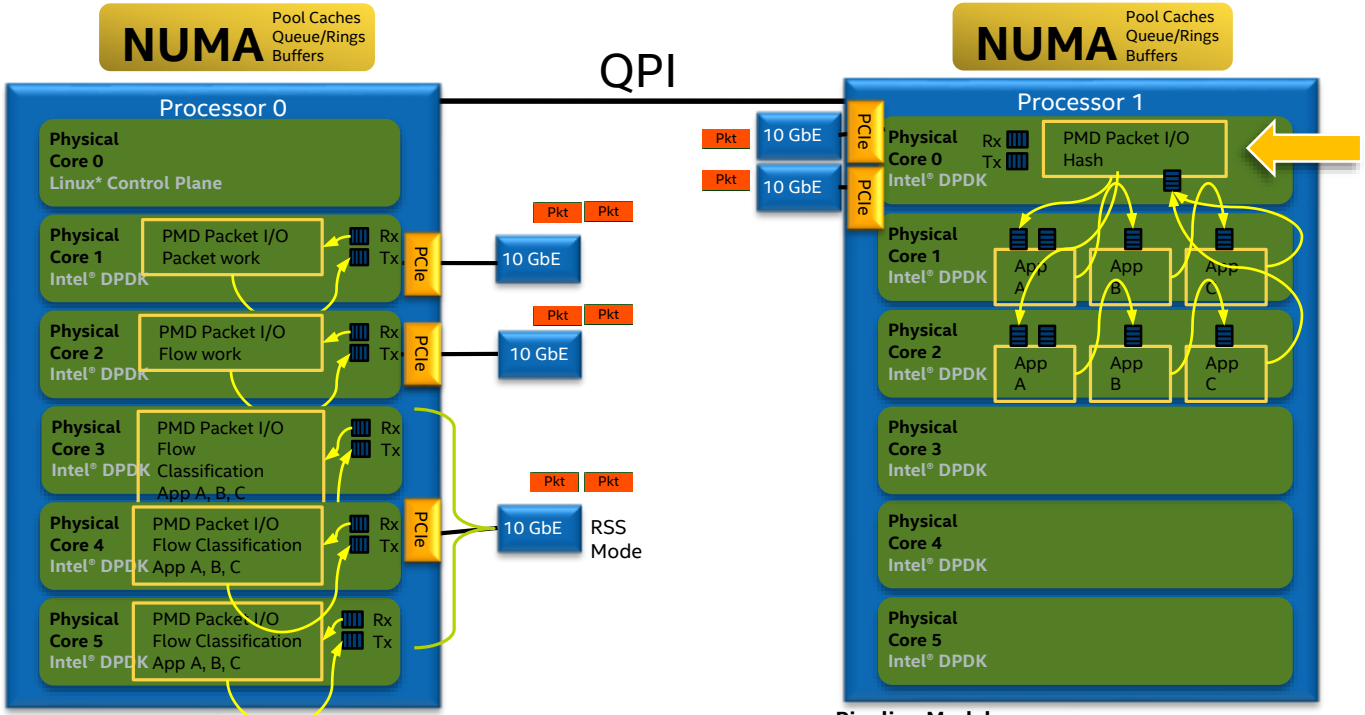
Attribute	Comments
Vector instruction set	CPU core supports vector instruction set for integer and floating point (SSE: 128-bit integer, AVX1:128-bit integer, AVX2: 256-bit integer)
Huge-pages	Intel CPUs support 4K, 2MB and 1GB page sizes. Picking the right page size for the data structure minimizes TLB thrashing. DPDK uses hugetlbfs to manage physically mapped huge page area
Hardware prefetch	Intel CPUs support prefetching data into all levels of the cache hierarchy (L1, L2, LLC).
Cache and memory alignment	DPDK aligns all its data structures to 64B multiples. This avoids elements straddling cache lines and DDR memory lines fulfilling requests with single read cycles
Intel Data Direct I/O (DDIO)	Is a methodology on Xeon E5 and E7 Platforms where packet I/O data is placed directly in LLC on ingress and sourced from LLC on egress
Cache QoS	Allows way-allocation control of LLC between multiple applications, controlled by software
CPU Frequency scaling/Turbo	Allows the core to temporarily boost CPU frequency higher for single threaded performance
NUMA	Non-Uniform Memory Architecture – as much as possible, DPDK tries to allocate memory as close to the core where the code is executing.

Achieving Performance – Software Concepts

Attribute	Comments
Complete user space implementation	Allows quick prototyping and development. Compiler can aggressively optimize to use complete instruction set
Software prefetch	DPDK also uses SW prefetch instructions to limit the effect of memory latency for software pipelining
Core-thread affinity	Threads are affinitized to a particular core, and quite often have cores dedicated to certain functions. Prevents reloading L1, L2 with instructions/data when threads hop from core to core
Use of vector instructions	The code implements algorithms using as much of the instruction set as possible – we use vector (SSE, AVX) instructions to implement some components providing significant speed up
Function in-lining	DPDK implements a number of performance critical functions in header files for easier compiler in-lining.
Algorithmic optimizations	To implement functions common in network processing e.g. n-tuple lookups, wildcards, ACLs etc.
Hardware offload libraries	Hardware offloads can complement the software implementation when the required hardware capability is available. E.g. 5-tuple lookups can be done on most modern NICs, and act in conjunction with a software classifier implementation
Bulk functions	Most functions support a “bulk” mode – processing ‘n’ packets simultaneously. Allows for software pipelining to overcome memory latency

PCIe* Connectivity and Core Usage

Using run-to-completion or pipeline software models



Can handle more I/O on fewer cores with vectorization

Run to Completion Model

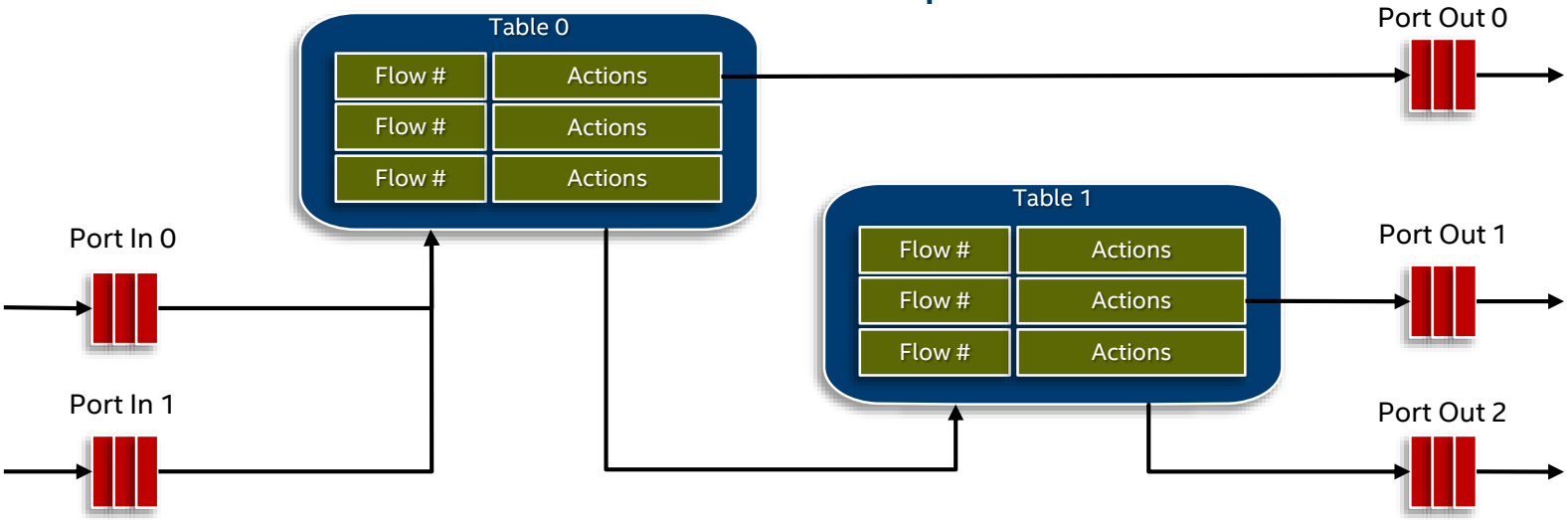
- I/O and Application workload can be handled on a single core
- I/O can be scaled over multiple cores

Pipeline Model

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

Packet Framework

You don't need to write code to move packets.



Standard methodology for ***pipeline*** development.
Ports and ***tables*** are connected together in tree-like topologies , with tables providing the ***actions*** to be executed on input packets.

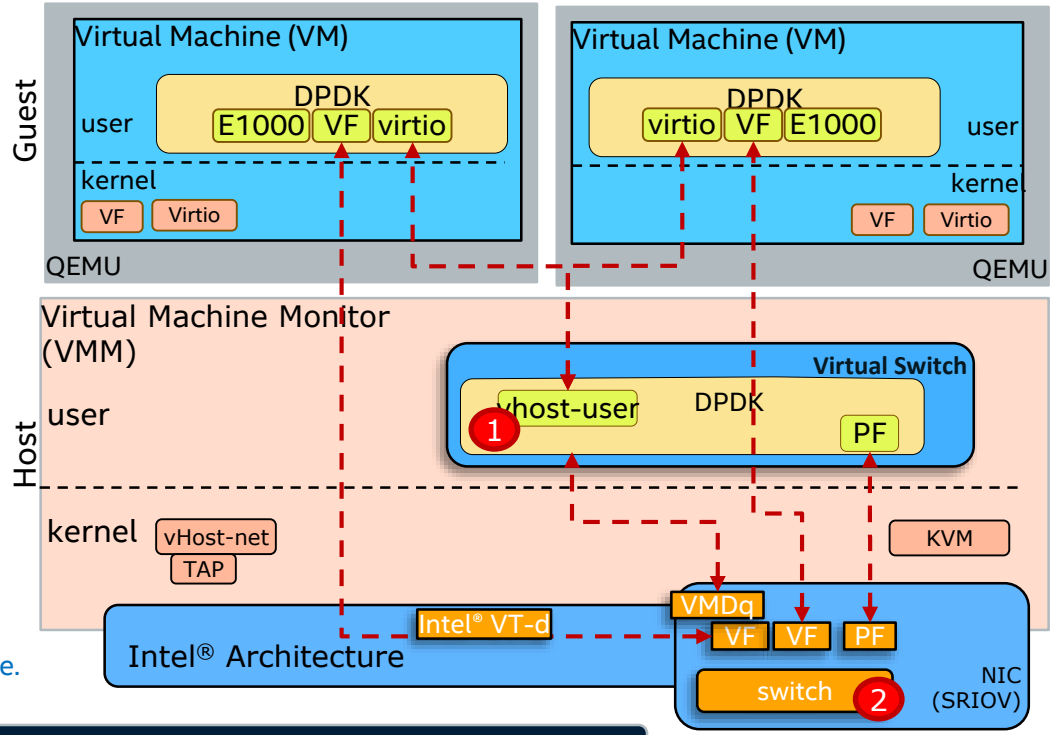
VIRTUALIZATION

DPDK Virtualization Architecture

Packet movement models

1. **DPDK Virtio PMD with QEMU "vhost-user" backend**
   - API compatible with existing Virtio/Vhost.
   - Control plane based on communication between QEMU and another userspace process using unix domain socket. HugeTLBFS is required. New netdev backend used to initialize vhost-net with vhost-user backend.
2. **Packet Switching between VMs**
   - through NIC features such as Mirroring, Virtual Ethernet Bridging, and SRIOV (using Flow Director redirection).

Note: Use of VM's Linux virtio will result in poor performance.



In depth solution brief: Enabling NFV to deliver on its Promise

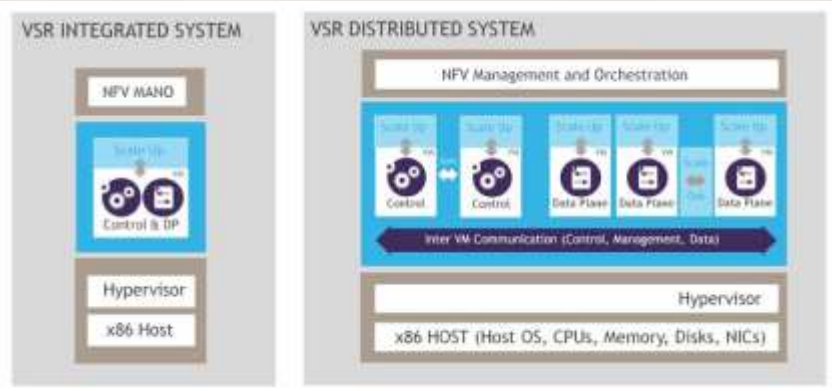
SUCCESSFUL DPDK DEPLOYMENTS

Alcatel-Lucent Virtualized Service Router

Alcatel·Lucent

To help service providers realize the vision of delivering more dynamic network services in a more flexible, efficient manner, Alcatel-Lucent is introducing the Virtualized Service Router (VSR): the industry's first full-service, carrier-grade, virtualized IP/MPLS edge router. The VSR builds on over a decade of investment and edge routing expertise gained by working with over 650 service providers worldwide.

In addition, Alcatel-Lucent has partnered with Intel® to optimize how the VSR interacts with the underlying server and its input/output (I/O) ports. Tools such as the Intel **Data Plane Development Kit (DPDK)** and Single Root I/O Virtualization (SR-IOV) are used to drive the highest possible data plane performance for the VSR operating in x86 environments.



https://www.sdxcentral.com/wp-content/uploads/2015/04/Alcatel-Lucent-MKT2014108119EN_VSR_eBrochure.pdf

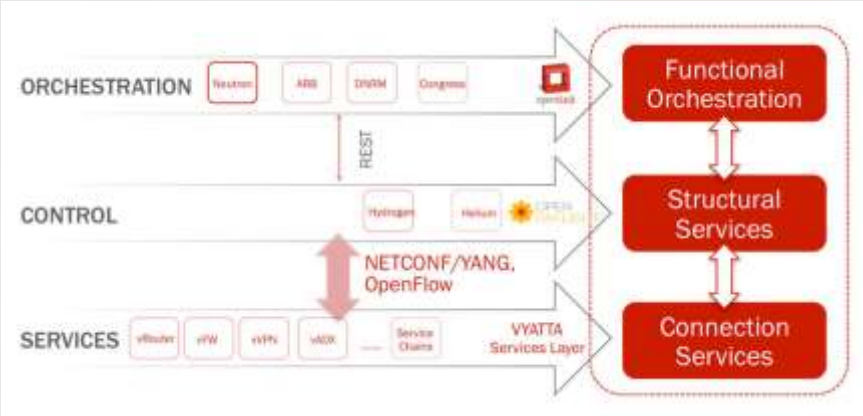
Brocade Vyatta 5600 vRouter

BROCADE

The Brocade® Vyatta® 5600 vRouter is purpose-built for Network Functions Virtualization (NFV), bringing an impressive performance boost.

Brocade vPlane™ technology enables hardware-like routing performance in a software-based network appliance. It is the industry's first highly scalable data forwarding plane for next-generation telco, enterprise, and cloud networks. Leveraging innovations from Brocade and the Intel **Data Plane Development Kit (DPDK)**, vPlane technology delivers breakthrough levels of performance and enables more efficient network designs for various data center and telco use cases.

https://www.brocade.com/content/dam/common/documents/content-types/datasheet/brocade-5600-vrouter-ds.pdf

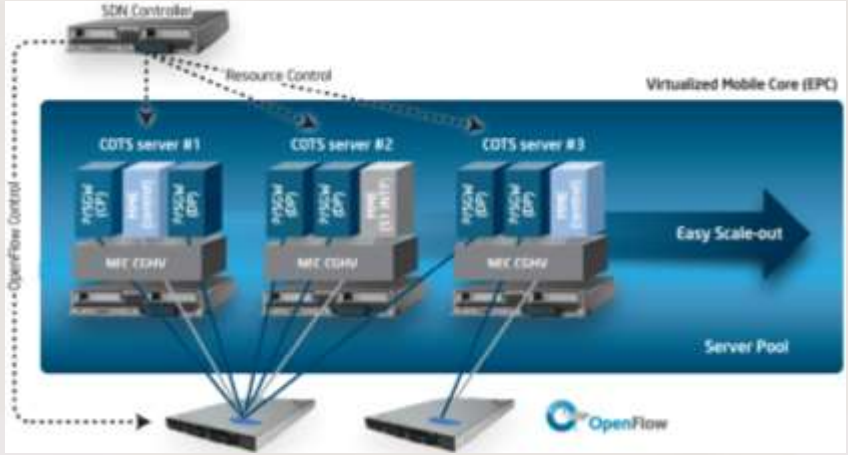


NEC Virtualized EPC

NEC

EPC has been traditionally deployed in various platforms using dedicated hardware for specific workloads within the core network. Advances in Intel® microarchitecture, software and networking solutions have made the consolidation of these specific workloads onto a common Intel architecture server platform possible.

The NEC vEPC has been realized on commercial off-the-shelf (COTS) servers as virtualized networking functions. Most of NEC vEPC software reuses that of existing ATCA-based, non-virtualized EPC products, which have a rich experience and proven quality in commercial networks. Also, in order to maintain carrier-grade qualities on a virtualization platform and maximize virtualization benefits, NEC CGHV (Carrier-Grade HyperVisor) has been introduced to vEPC and the Intel **DPDK technology**.



http://networkbuilders.intel.com/docs/communications_nec_virtualized_epc_paper.pdf

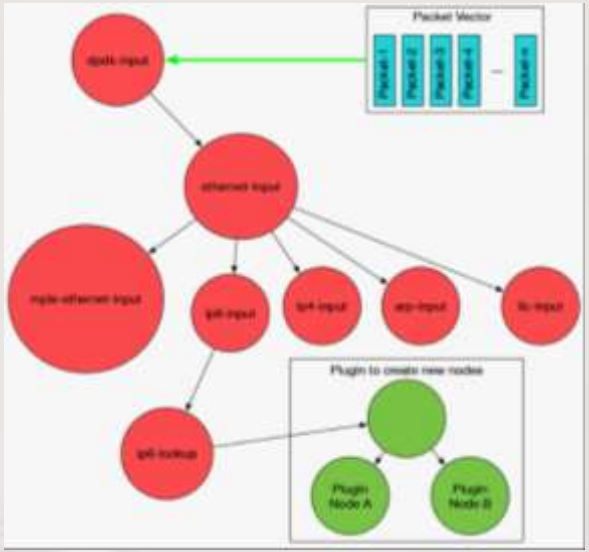
OPEN SOURCE PROJECTS BASED ON DPDK

Fast Data (FD.io) Vector Pkt Processing (VPP)

FD.io is an open source project to provide an IO services framework for the next wave of network and storage software.

The initial release of FD.io is fully functional and available for download, providing an out-of-the-box vSwitch/vRouter utilizing the **Data Plane Development Kit (DPDK)** for high-performance, hardware-independent I/O. The initial release will also include a full build, tooling, debug, and development environment and an OpenDaylight management agent. FD.io will also include a Honeycomb agent to expose netconf/yang models of data plane functionality to simplify integration with OpenDaylight and other SDN technologies.

https://fd.io/

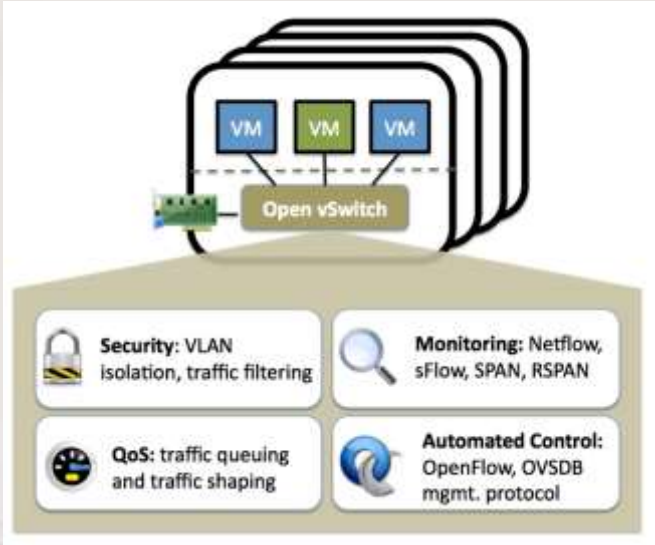


Open vSwitch

Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag).

Open vSwitch can use **DPDK** to operate entirely in userspace. Using DPDK with OVS gives us tremendous performance benefits. Similar to other DPDK-based applications, we see a huge increase in network packet throughput and much lower latencies.

http://openvswitch.org/
https://software.intel.com/en-us/articles/using-open-vswitch-with-dpdk-for-inter-vm-nfv-applications



Packet Generators

Pktgen

Pktgen, (*Packet Gen*-erator) is a software based traffic generator powered by the DPDK fast packet processing framework.
https://github.com/Pktgen/Pktgen-DPDK

Trex

TRex is an open source, low cost, stateful traffic generator fuelled by DPDK. It generates L4-7 traffic based on pre-processing and smart replay of real traffic templates.
http://trex-tgn.cisco.com/

MoonGen

MoonGen is a fully scriptable high-speed packet generator built on DPDK and LuaJIT.
https://github.com/emmericp/MoonGen

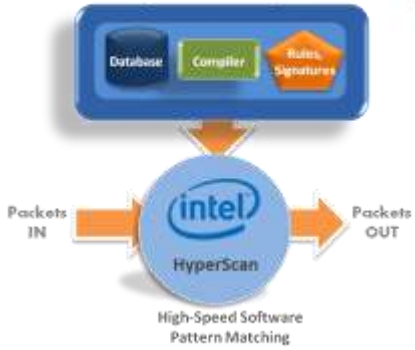
Ostinato

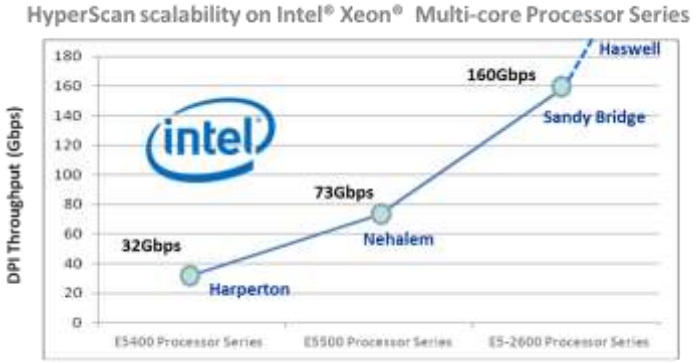
Ostinato is an open-source, cross-platform network packet crafter/traffic generator and analyzer with a friendly GUI and powerful Python API.
http://ostinato.org/

RELATED PACKET PROCESSING TECHNOLOGIES

Hyperscan

- Software Regex (Pattern Matching) engine
  - Regex and Fixed-string matching
  - Massively parallel matching
- Scan content for malware
  - Using customers' patterns sets
- High performance
  - Market moving to software DPI
- Fully scales IA
- Low latency and overhead
- Portable, easy to integrate
- Wide application
  - Network Security and infrastructure equipment suppliers





Using the same test criteria and database for every platform benchmark

Intel® Architecture + DPDK + HyperScan -> Best in class DPI

experience
what's inside™