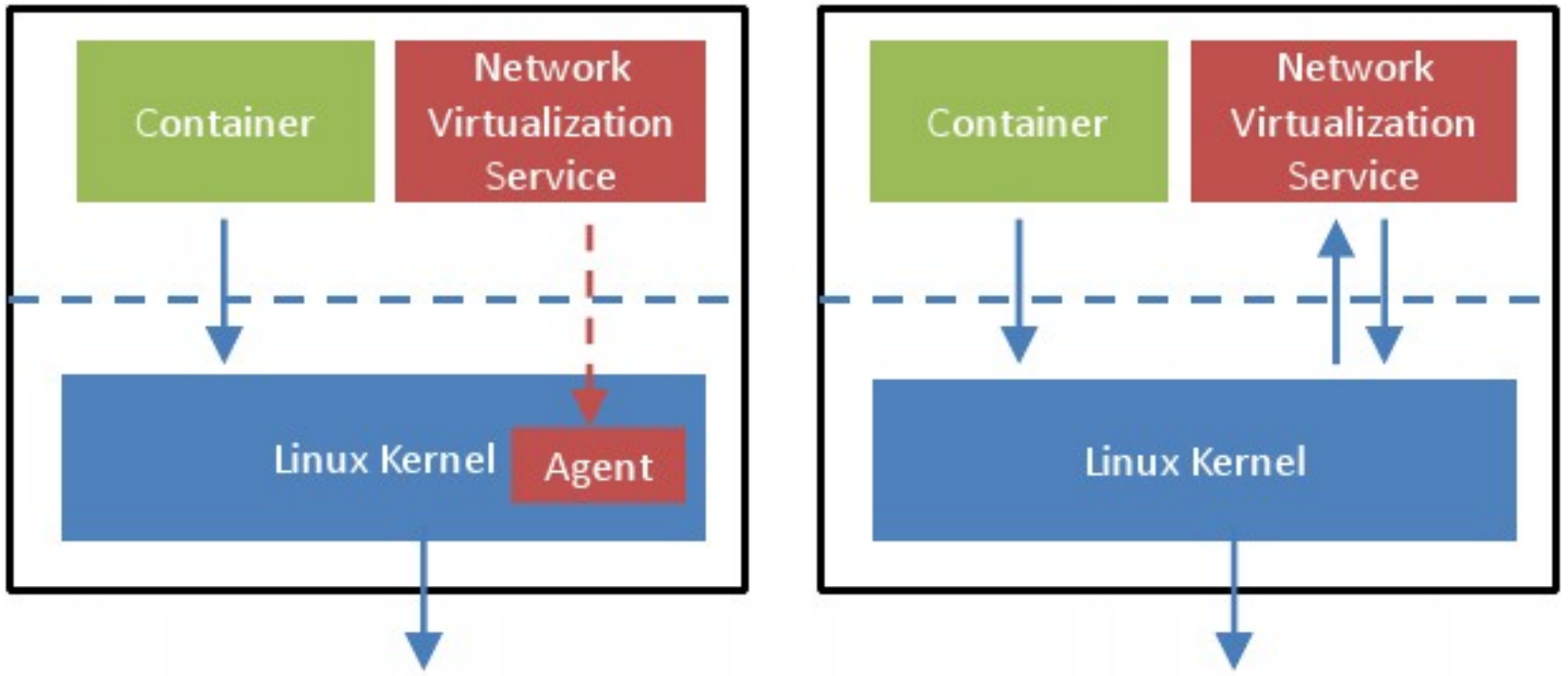
Container Networking Powered by

Agenda

- Background
- DPDK-powered techniques
  - Using SR-IOV + DPDK in Containers
  - Connect containers with user space vswitch
  - User space network stack
- DPDK-powered VNFs

Container networking status quo

- Multi-host networking



But not ready for scenarios like…

- High-throughput networking functions like
  - LB, FW, IDS/IPS, DPI, VPN, pktgen, Proxy, AppFilter
- Latency-sensitive and jitter-avoid applications
  - Game applications
  - E-commerce flash sales
  - Stock exchange trading
  - Video conference

Challenges of high perf. network

NIC	Time budget for 64B	Time budget for 1518B
10Gb	67.2 ns	1,230 ns
40Gb	N/A	307 ns
100Gb	N/A	120 ns

NIC	Time budget
System call	75 ns/42 ns
Atomic ops	8.25 ns
Spinlock lock/unlock	16+ ns
L3 miss	~80 ns

- FWD 1~2 Mpps per core

Data from LWN article, 3GHz CPU

Open Source Software

DPDK

DATA PLANE DEVELOPMENT KIT

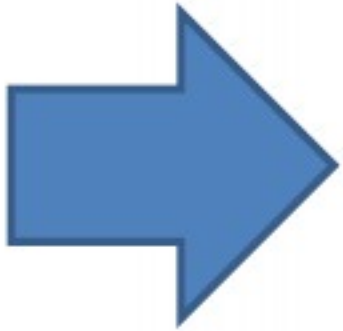
The Data Plane Development Kit (DPDK) is a set of software libraries for accelerating packet processing workloads on COTS hardware platforms.

Customer Adoption

HUAWEI, cisco, ebay, hp, ORACLE, VERISIGN, JUNIPER NETWORKS, ZTE中兴, DELL, Alibaba.com, ERICSSON, Tencent 腾讯, Google, BARCLAYS, Alcatel·Lucent, amazon.com, Nokia Siemens Networks, Baidu 百度, ADVANTECH, J.P.Morgan

How do we solve it in BM - DPDK

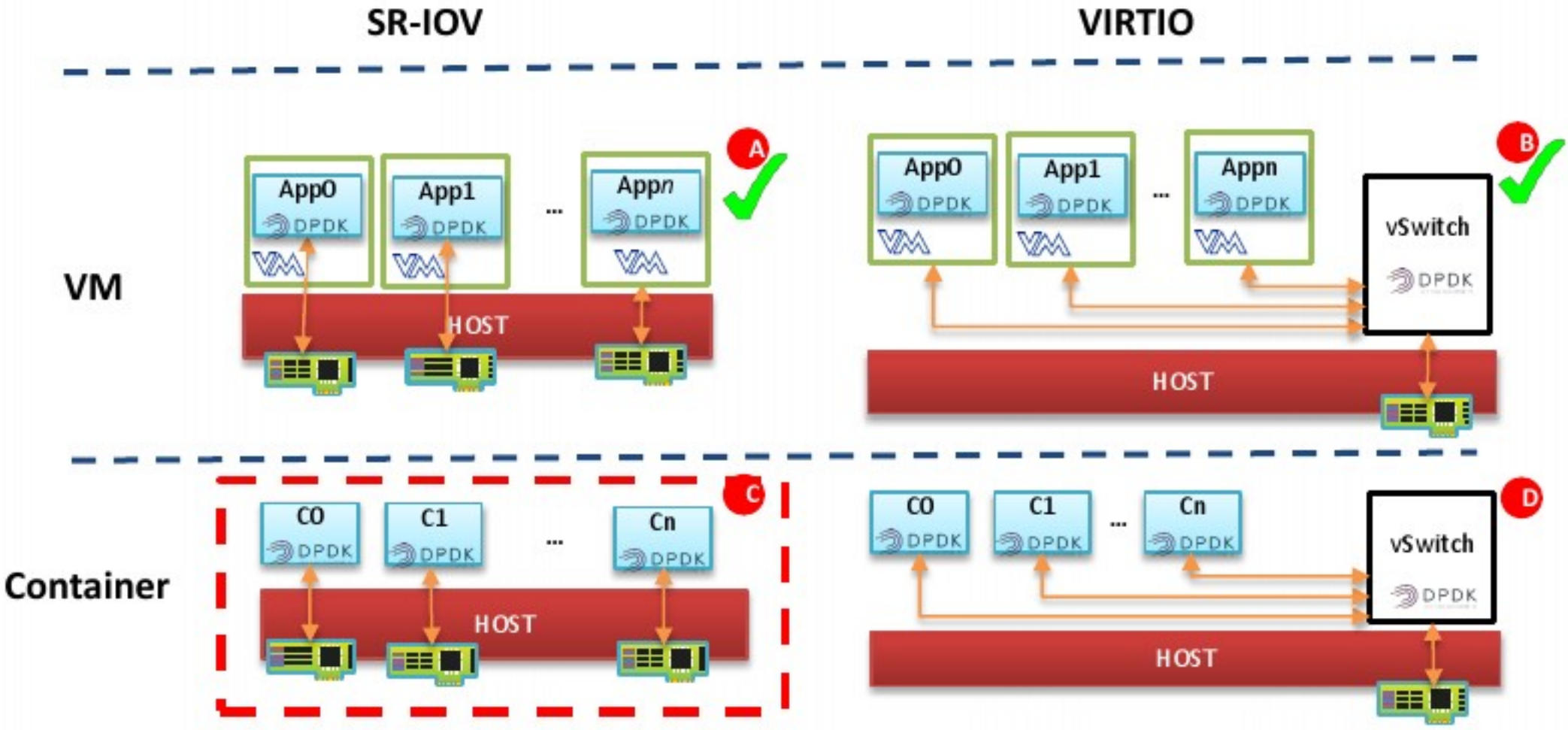
- CPU affinity
- Hugepages
- **UIO**
- **Polling**
- Lockless
- **Batching**
- SSE/AVX



- High-throughput
- Low-latency
- Deterministic

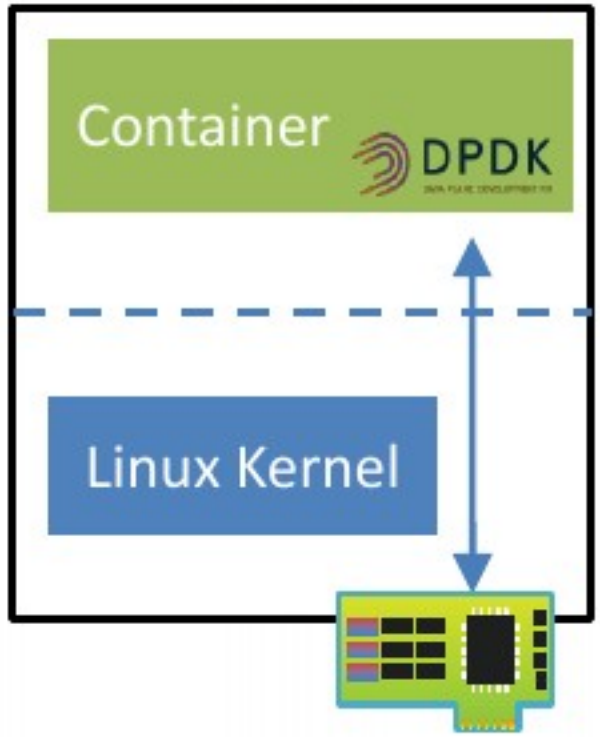
Can we leverage DPDK to accelerate Container Networking?

VM vs Container

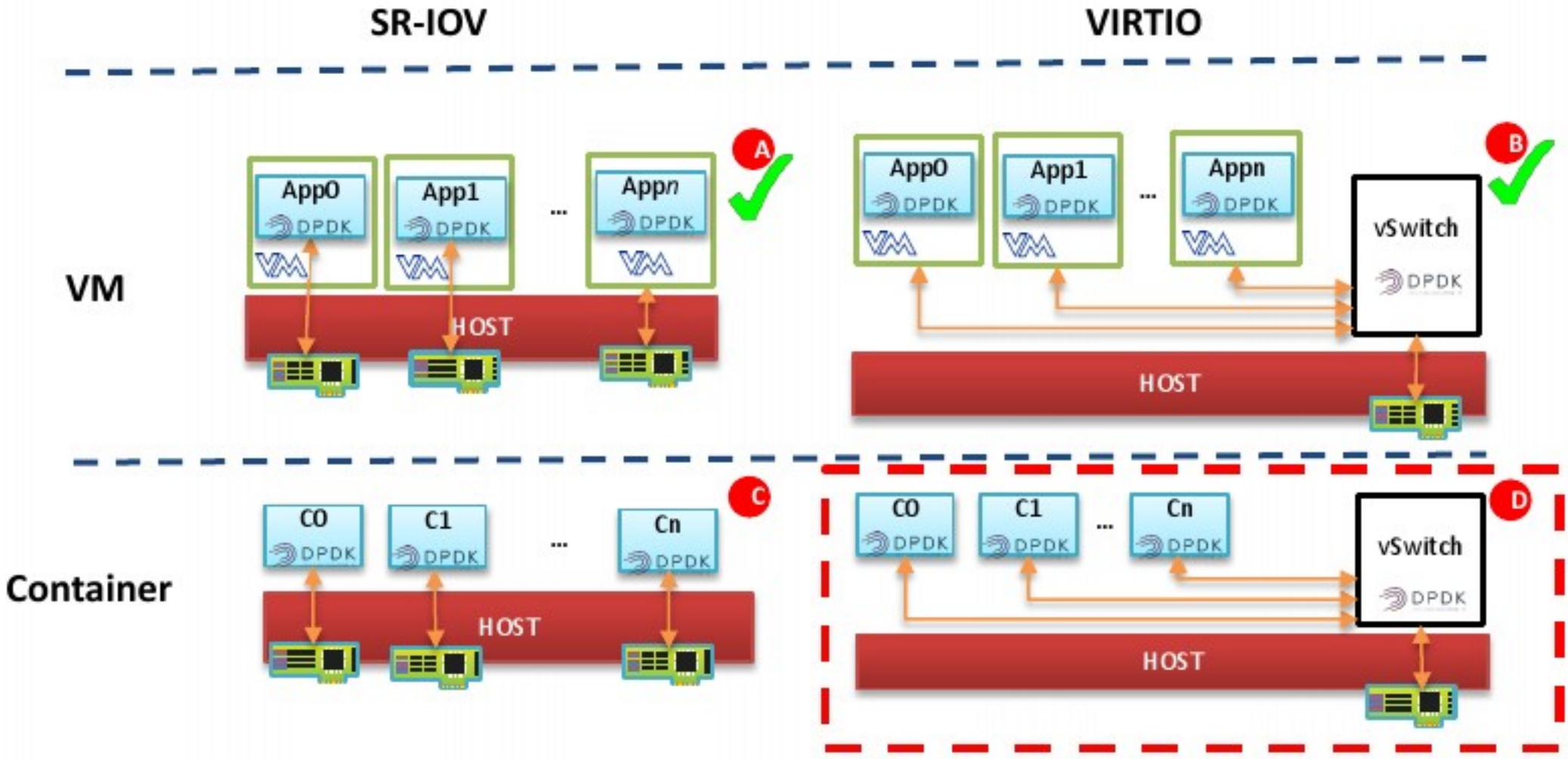


Using SR-IOV + DPDK in Container

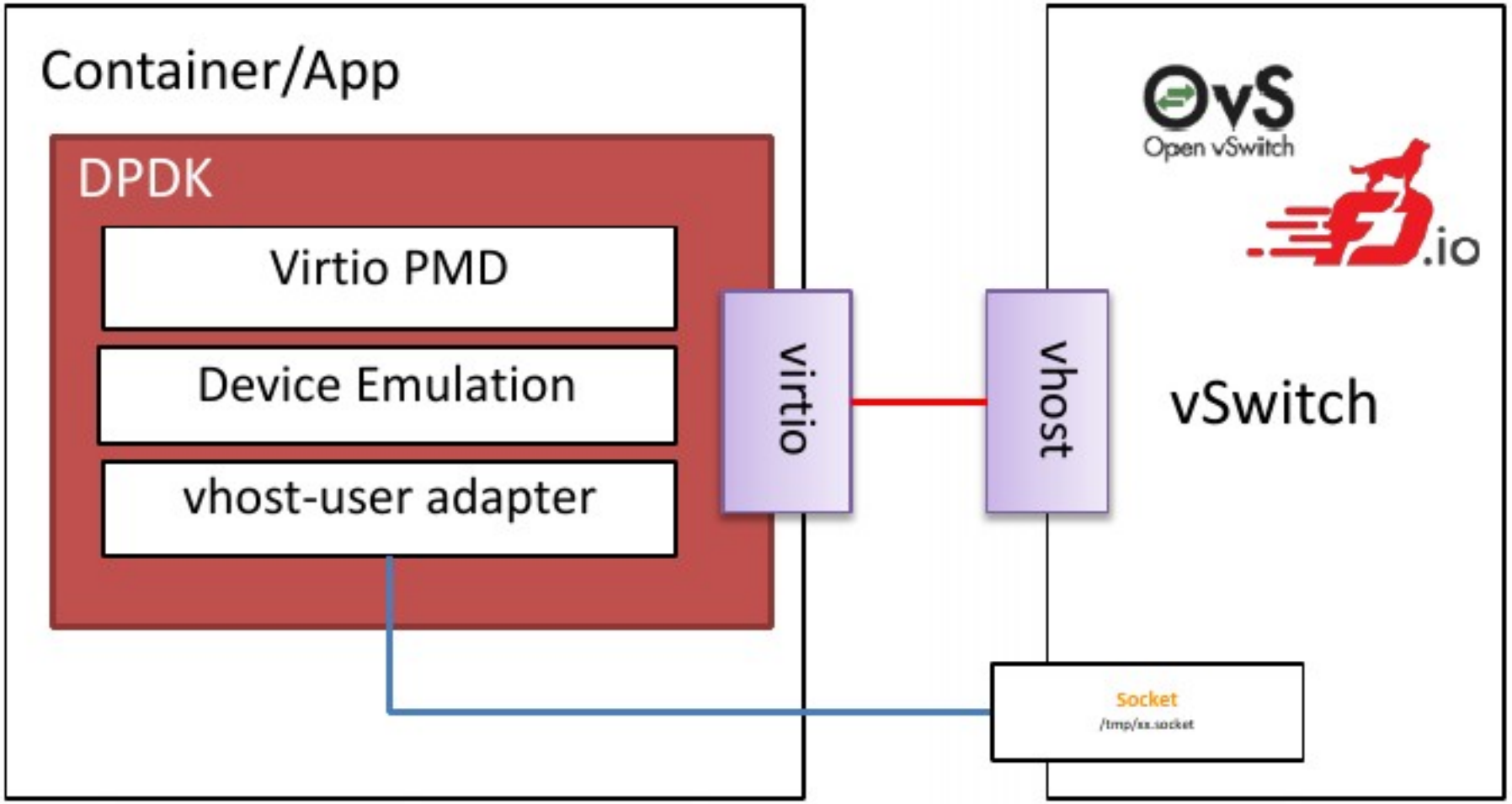
- Requires: device mapping (vfio)
- High-performance: small pkts line rate with 10 GbE
- but
  - # of VFs is limited (64 or 128)
  - Not flexible (by HW)



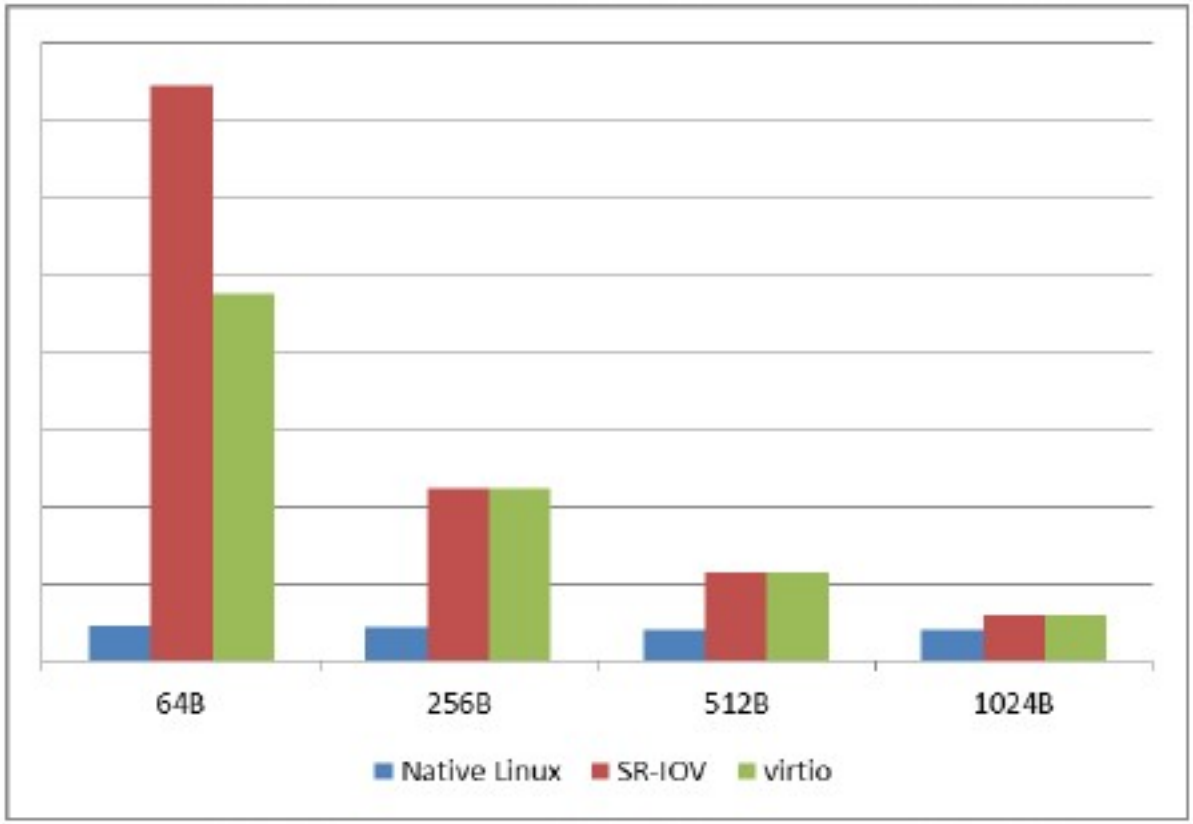
VM vs Container

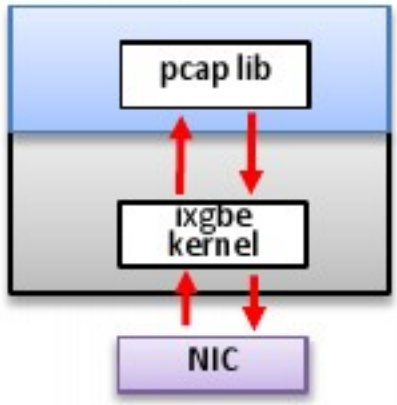


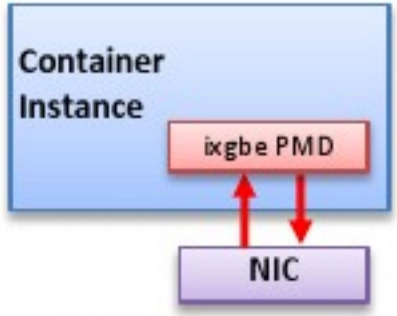
Connect containers with user space vswitch

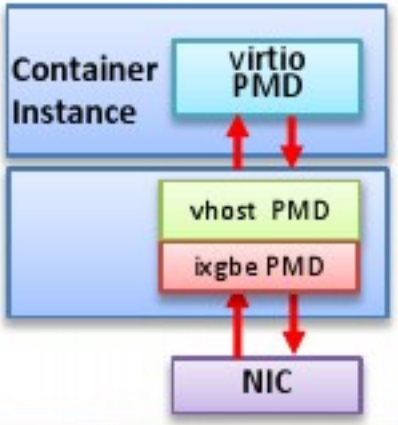


Performance Evaluation - throughput



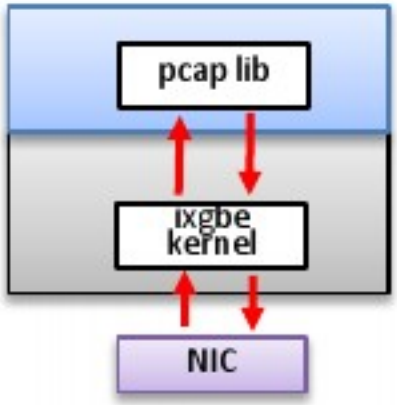


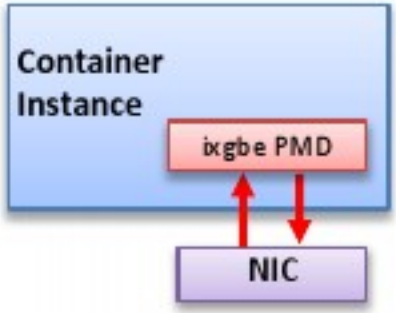


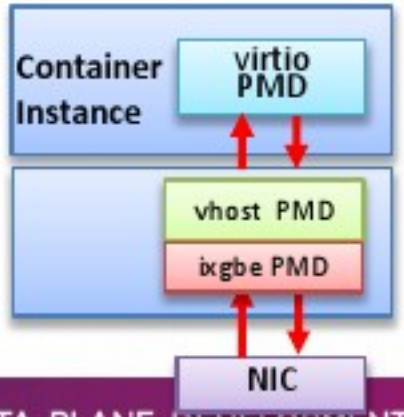


Performance Evaluation - latency

- For native Linux, ms level
- For the other two, us level







More about determinacy

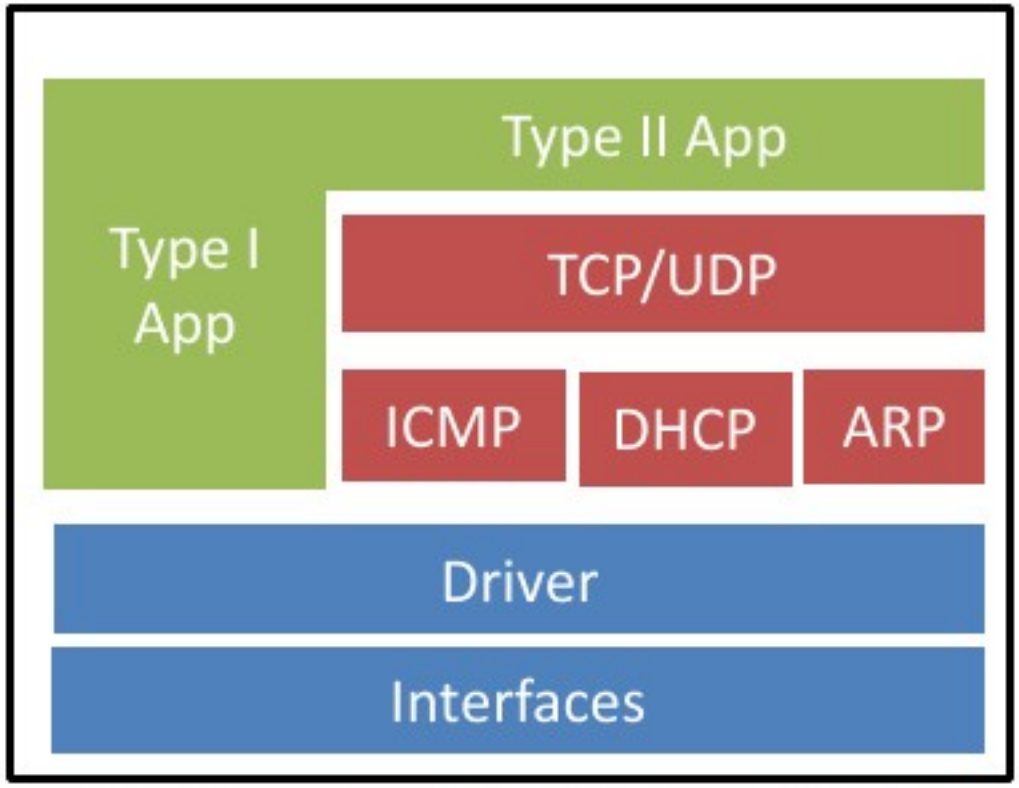
- Deterministic CPU env
  - Disable timer / task scheduler
  - Core-thread affinity
- Deterministic cache/memory env
  - Data Direct I/O (DDIO) technology
  - Cache Allocation Technology (CAT)
  - Software prefetch

Agenda

- *Background*
- *DPDK-powered techniques*
  - *Using SR-IOV + DPDK in Containers*
  - *Connect containers with user space vswitch*
  - **User space network stack**
- DPDK-powered VNFs

User space network stack

- Type I: DPI, FW …
- Type II: Applications in need of TCP/UDP stack



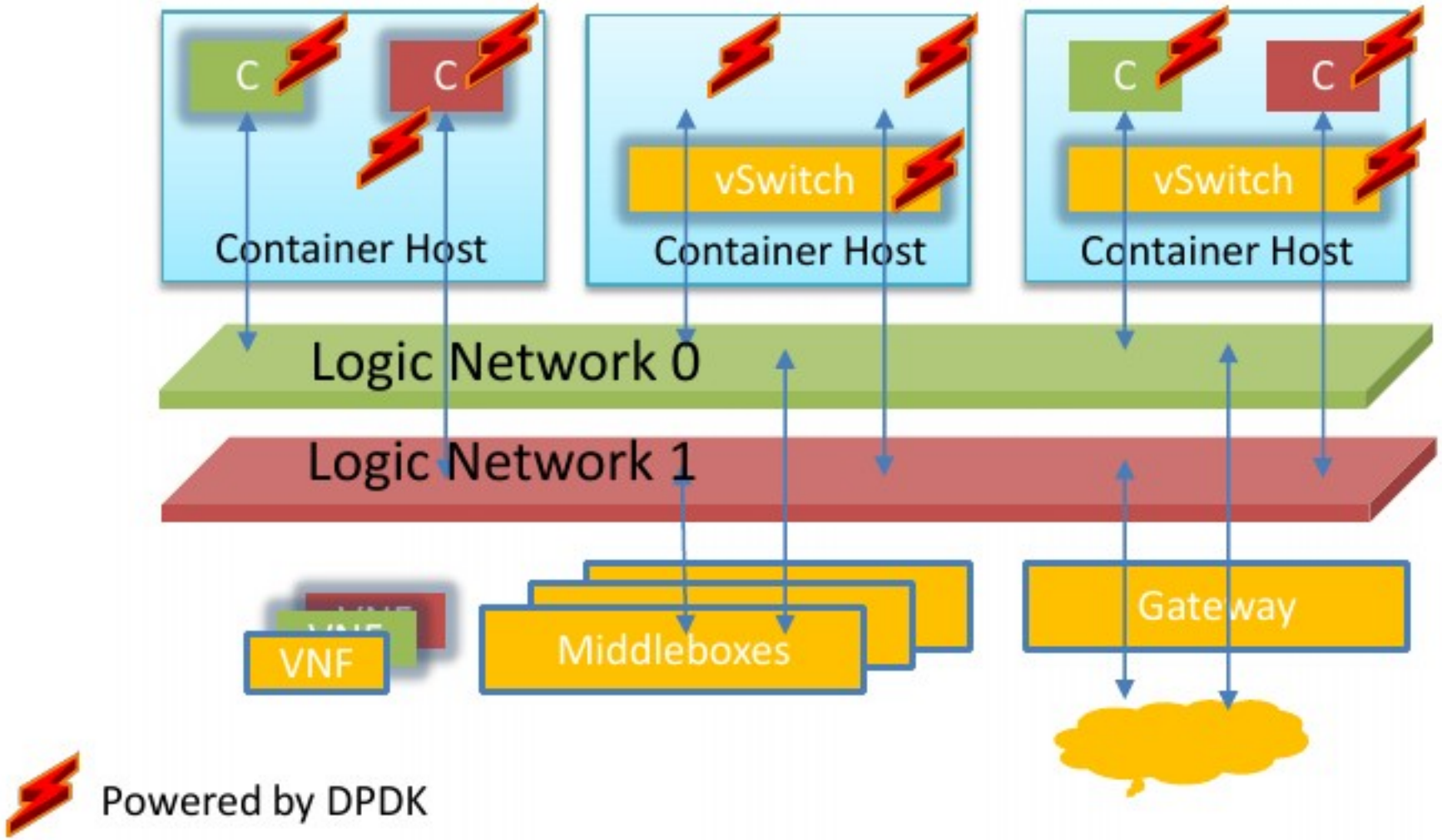
User space network stack

- TCP/UDP stacks
  - From scratch: mTCP, LwIP, Light
  - Ported: libuinet, NUSE (libos), Linux Kernel Library
  - To choose a open source stack, consider
    - Integration effort
    - Performance
    - Compatibility

Agenda

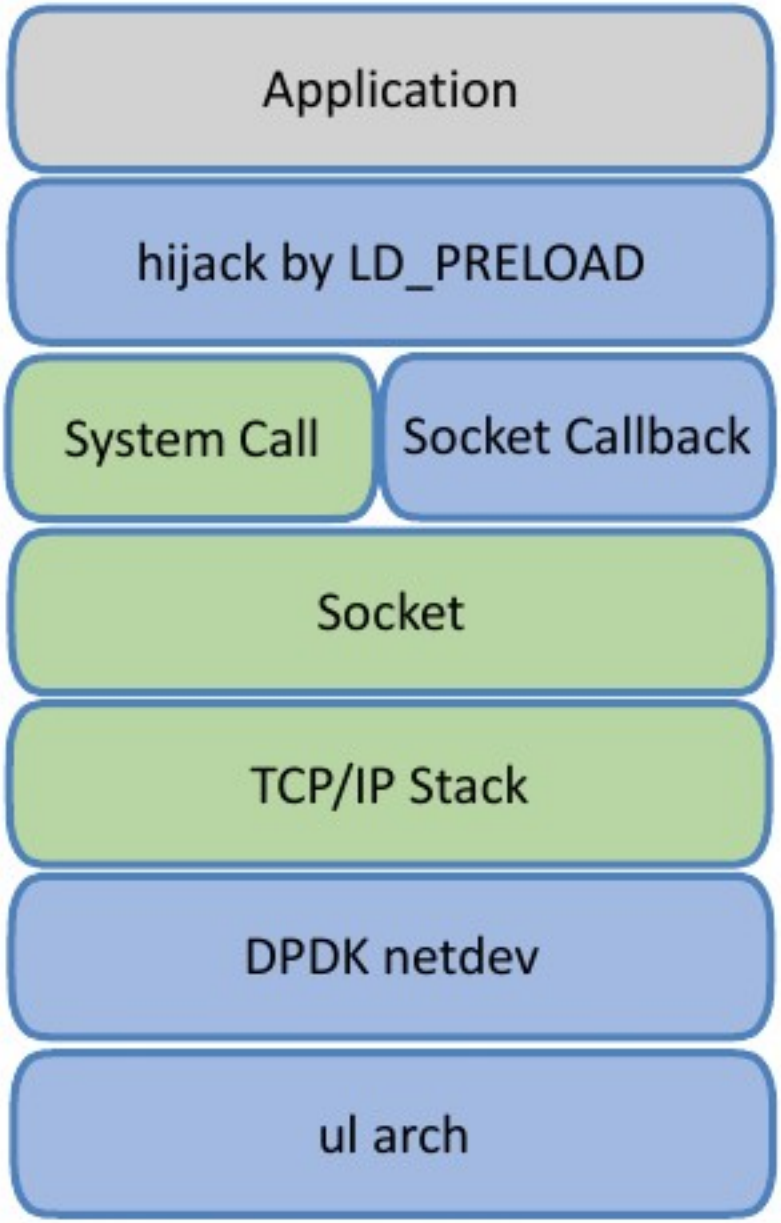
- *Background*
- *DPDK-powered techniques*
  - *Using SR-IOV + DPDK in Containers*
  - *Connect containers with user space vswitch*
    - *DPDK-powered vSwitch*
    - *Virtio for container*
  - *User space networking*
- **DPDK-powered VNFs**

Transform middleboxes with DPDK-powered VNFs



Vortex from Ucloud

- Scale-up L4 LB
  - PPS: 14M (64B line rates)
  - CPS: 200K+
  - CC: 30M+
- Hardware
  - CPU: Xeon E5-2670v2 (10 core 2.5G) * 2
  - NIC: 82599ES 10GbE
- LKL



pkt-gen - TRex

- Features
  - L4-7 traffic
  - Latency/Jitter measurements
  - Flow ordering checks
  - NAT, PAT dynamic translation learning
  - Cross flow support (e.g RTSP/SIP) using plugins
- Performance
  - 200Gb/sec with one Cisco UCS (Intel XL710)

Other promising workloads

- Webserver: nginx
- In-memory DB: redis
- Memory cashing system: memcached
- Distributed FS: Ceph
- ……

Summary

- Use DPDK to power container networking
  - SR-IOV (existing)
  - Virtio (will be available in DPDK 16.07)
- Compared to traditional ways, we provide a way to achieve
  - High throughput
  - Low latency
  - Deterministic networking