DPDK
SUMMIT
NORTH AMERICA

DPDK for Windows Platform goes Mainstream

WINDOWS WORKGROUP

Agenda

DPDK SUMMIT NORTH AMERICA

- Motivation
- Journey so far
- Current Status
- Demos
- Challenges
- Learnings for Cross Compat
- Roadmap
- Call to Action
- Q&A

Motivation

- Developers need a low latency, high throughput Data path for modern workloads.

  "After about 5 years we have likely exhausted what existing Windows sockets can do for us. We've tried completion ports and RIO et all. As each new technology came out, we spent a fair amount of time testing the network throughput. Net result is, as now seems obvious, you don't get DPDK like performance without DPDK or similar"

- Windows N/W stack optimizations Diminishing returns
  - From 2016 -> 2019 Vs 2019 -> 2022
  - Not comparable to N/W stack bypass

Journey so far

DPDK SUMMIT NORTH AMERICA

- 2017: Fork launched
  - Trailing behind by many releases from community-maintained repo.
  - Separate development, build, testing pipeline.
  - Summit 2018, Summit 2019
- 2019: Upstreaming begins
  - DPDK releases v19.05, introduces Windows Support! - Microsoft Tech Community
- 2020: Accelerated contribution upstream
  - Microsoft joins as a contributor and Sponsor
  - Community expands
    - NVIDIA, Intel, Cisco, Marvell
    - Community Subject matter experts
    - MayaData, Datapath and other Independent Software Vendors
    - Storage Performance Development Kit(SPDK) based on DPDK 21.05

Current Status – 21.05

DPDK SUMMIT NORTH AMERICA

Supported devices	DPDK on Windows 21.05
Platform	**X86**
Hardware NICs	**Mellanox ConnectX-4,5,6X Series, Intel® Ethernet 700 and 800 Series**
PMDs	**mlx5, i40e, ice, pcap , vmxnet3**
DPDK Libraries	**~15** core libraries
Kernel Drivers	**NetUIO, Virt2Phys**
Applications	Hello world, l2fwd, Testpmd(subset), cmdline, flow_filtering, ipv4_multicast, link_status_interrupt, qos_meter, rxtx_callbacks, service_cores, skeleton
Tool chain	**Meson-Ninja, Clang and Mingw64**
MISC	**Pmdinfogen**
CI/CD	**Compilation Gate: Clang/Mingw64**

Windows EAL — Challenges

DPDK SUMMIT NORTH AMERICA

Focused:

- Per-lcore variables
- Alarms
- Memory management

There are more:

- Threading (WIP)
- Interrupts (WIP)
- Meson (occasionally)
- Kernel-mode driver development
- Scripts
- Type layout differences
- Unix-isms

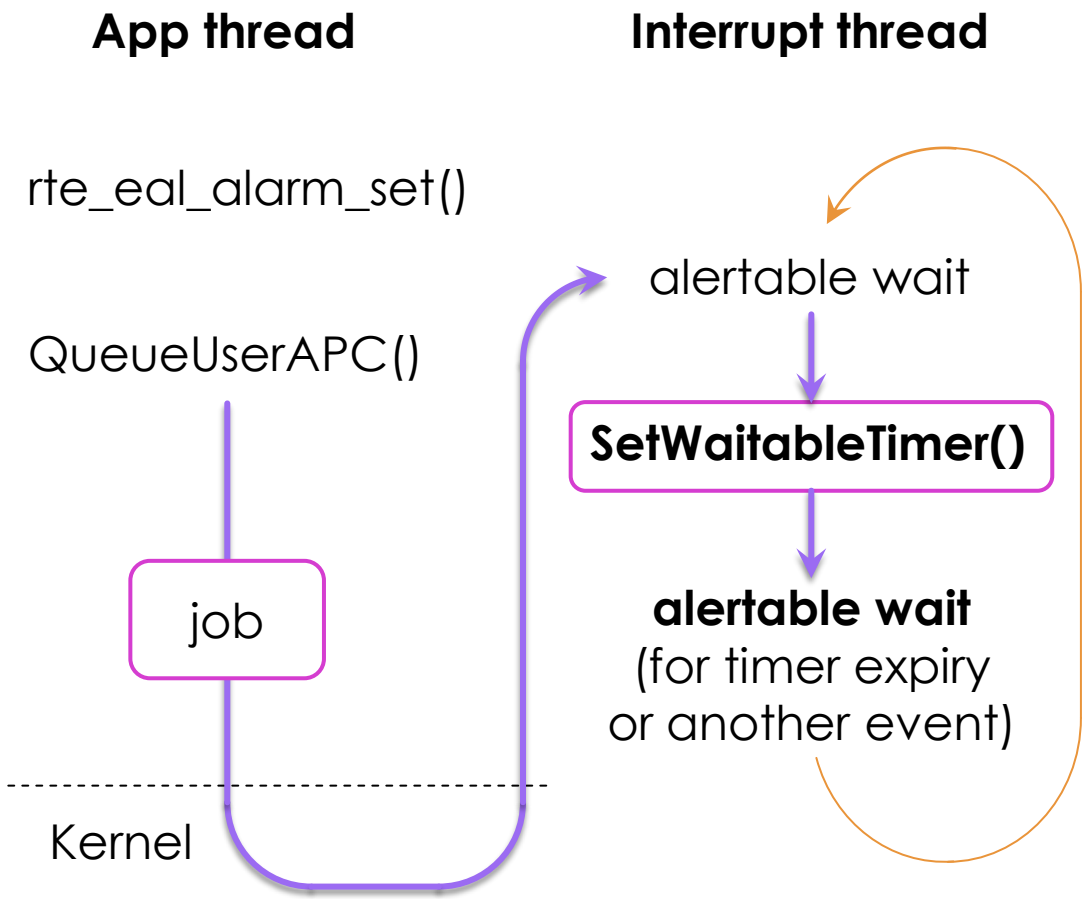
Per-lcore Variables (TLS)

DPDK SUMMIT NORTH AMERICA

- Make RTE_DEFINE_PER_LCORE from DLL work in EXE.
- No ELF SHF_TLS analog in PE format.
- Clang: compiler support
  - Fastest way possible
  - Not for variables defined in DLL
- MinGW: emulated with Win32 TLS API
  - About 15% slower than Linux
  - Requires exporting __emutls.VAR
- Demo code with results and comments.

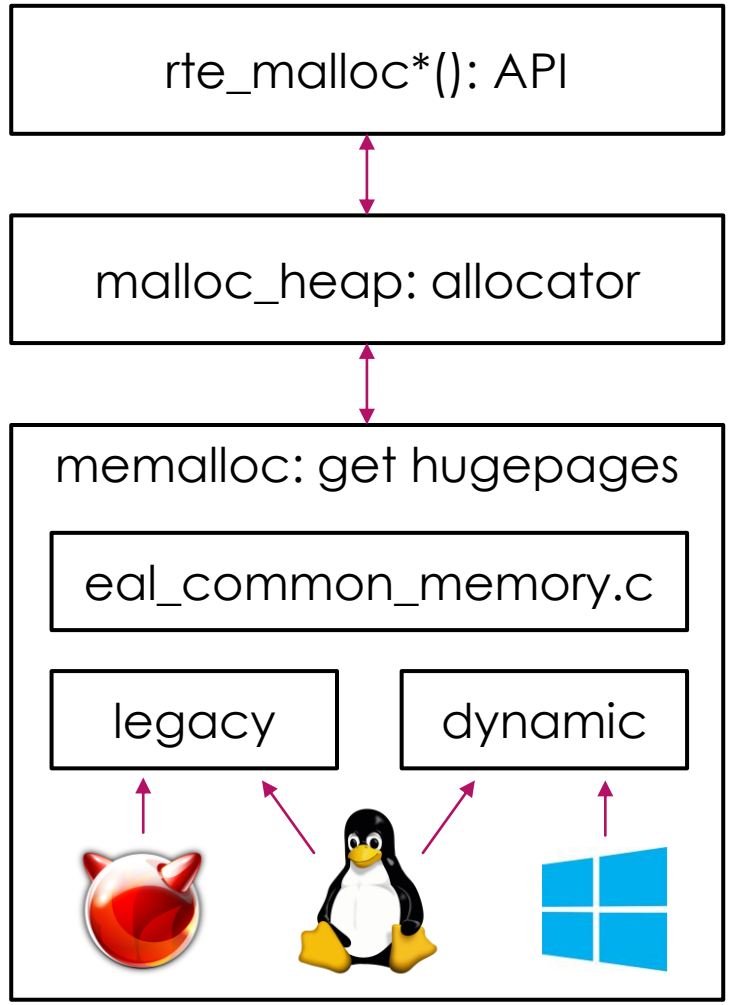
Alarms

- App thread sets alarms,
  DPDK interrupt thread runs callbacks.
  - Alarm is a “software interrupt”.
- Windows facilities:
  - **SetWaitableTimer**(timer,
    deadline, callback, userdata)
  - **“Alertable wait”** functions
  - When timer expires:
    - Thread doing alertable wait wakes up.
    - Callback is executed.
  - Must be called by the same thread.



Fitting into DPDK Memory Manager

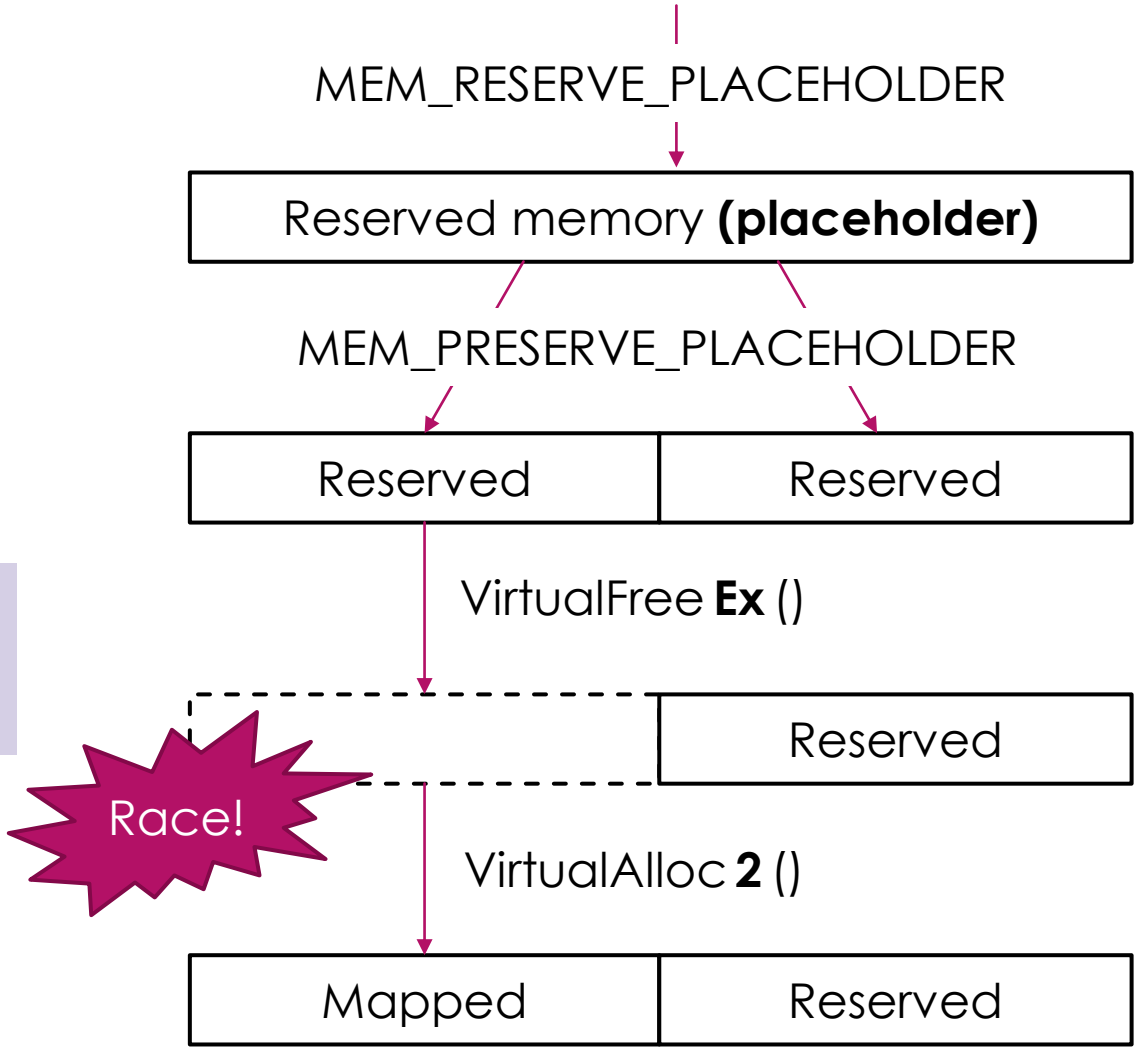
- Windows EAL must provide memalloc layer.
  - No changes required in allocator and API.
- Common sublayer requirements:
  - Reserve large areas of virtual addresses (VA).
    - Identical VA to support multi-process.
    - Similar layout with or without multi-process.
  - Map hugepages at parts of reserved VA areas.
- Windows limitations:
  - No multi-process
  - No legacy (static) allocation
  - 64-bit only



Mapping Hugepages on Windows

API	Huge page	Fixed VA	Both
Address Windowing Extensions (AWE)		Yes	
MapViewOfFile3	Yes	Yes	
VirtualAlloc2	Yes	Yes	Yes

* Windows Server 2019 / Windows 10 only

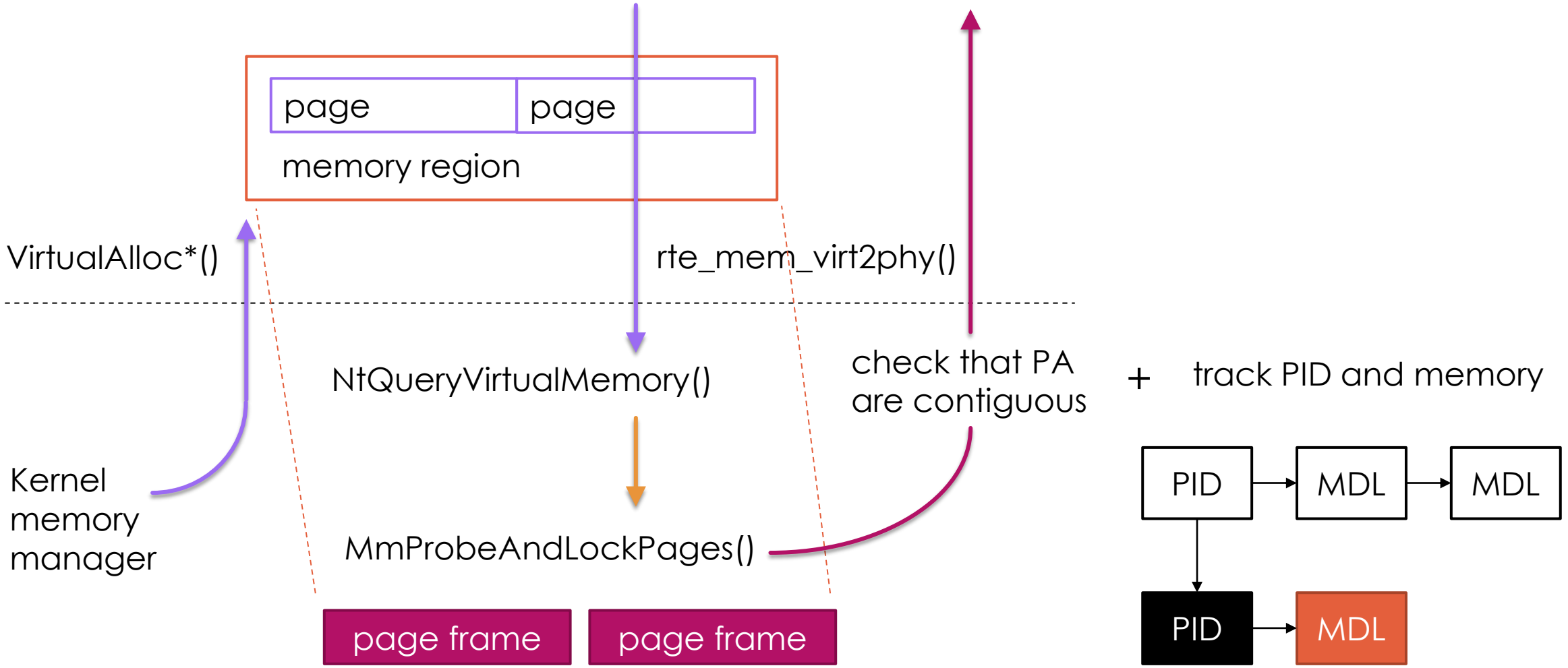


Physical Addresses (PA)

DPDK SUMMIT NORTH AMERICA

- Task: translate virtual address to physical address for DMA.
  - Kernel-mode driver required.
  - Driver code must be assessed, certified, and signed by Microsoft.
- Security: do not reuse PA while the process has it.
  - If VA is paged out and PA reused, malicious process can try access other process data by mapping PA back with AWE or some vulnerability.
  - Need to pin the memory **until the process exits**.
  - Need to track processes using the driver and their memory.
- Security: pinning exhausts RAM, tracking consumes CPU.
  - Need configurable limits.

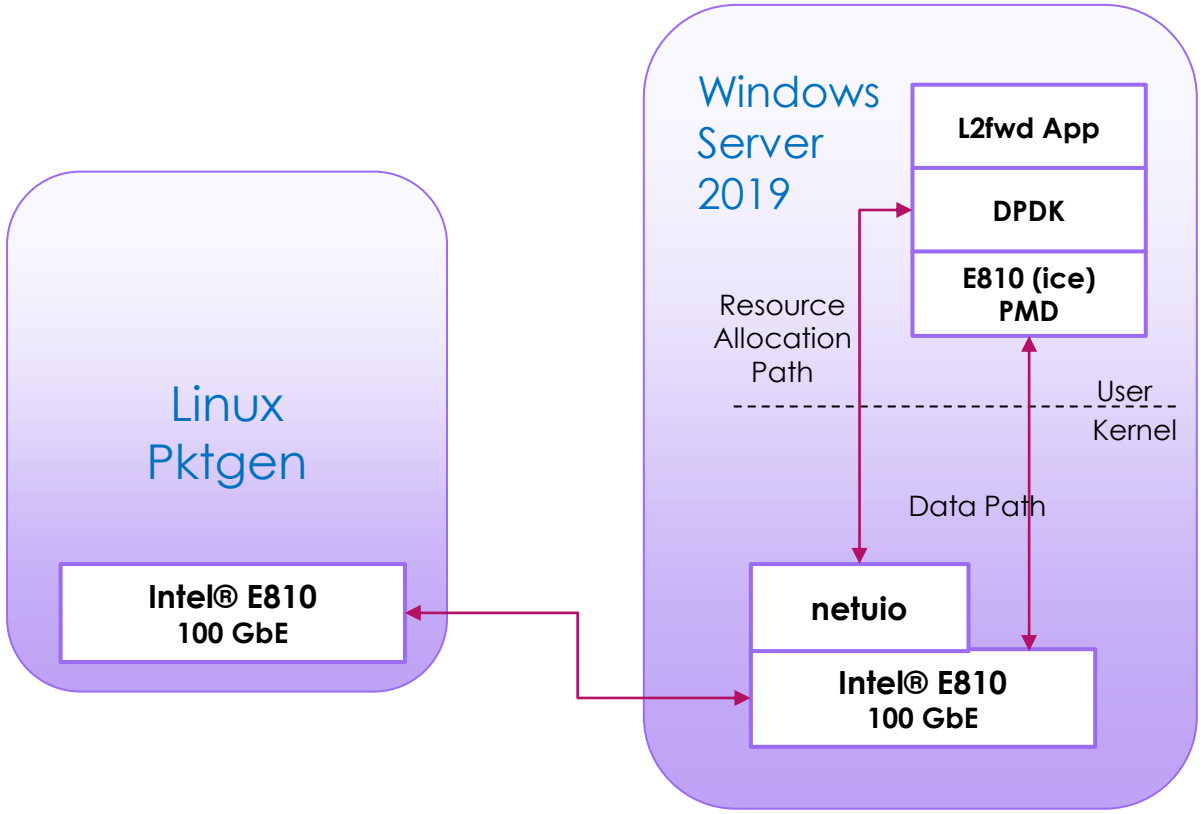
Virt2phys


page
page
memory region
VirtualAlloc*()
rte_mem_virt2phy()
NtQueryVirtualMemory()
check that PA
are contiguous
+
track PID and memory
Kernel
memory
manager
MmProbeAndLockPages()
page frame
page frame
PID
MDL
MDL
PID
MDL

Intel PMD

- Challenges/Learnings
  - Enabling DPDK on Windows with minimum code changes to DPDK code base
  - Clang LLVM compiler – supports gcc extensions and avoids *#ifdef windows* clutter
  - Optimal replace of system include files
  - Easily portable Intel PMDs
    - Avoid warnings – typecasting, integer comparisons
    - Define new macro for Linux specific APIs that are deprecated on Windows
- Future Roadmap
  - Verify the i40e and ice PMD features on Windows
  - Add iAVF PMD support on Windows
  - Automate the validation tasks for Intel PMDs on Windows - Functionality and performance analysis data

Intel PMD – l2fwd Demo



System Configuration:

CPU	Intel® Xeon® CPU E5-2680 v4 @ 2.40GHz
Total number of cores	28
OS	Windows server 2019
NIC	Intel® E810

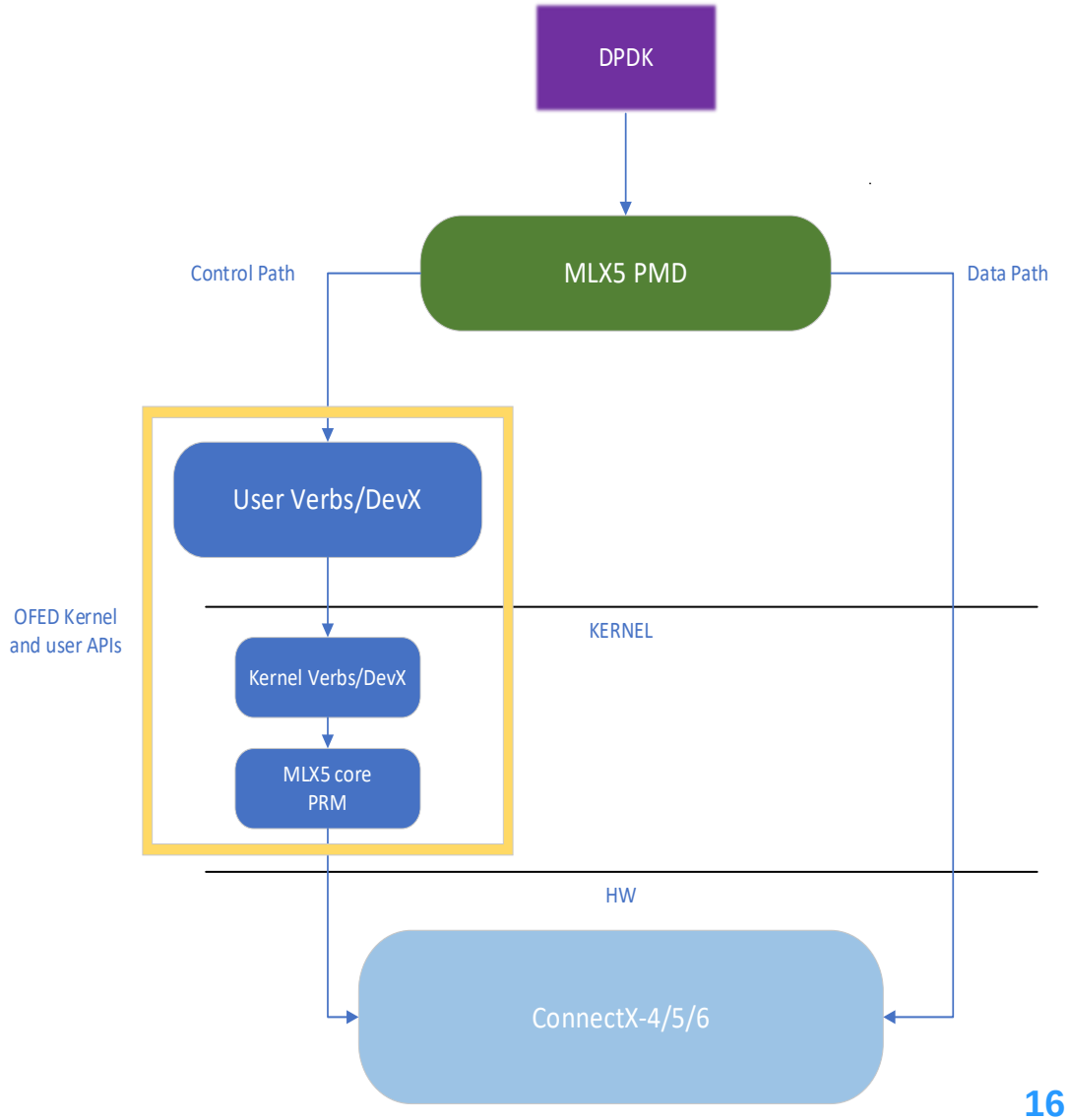
```
\ Ports 0-0 of 1   <Main Page>  Copyright (c) <2010-2019>, Intel Corporation
  Flags:Port        : P------Single      :0
Link State          :       <UP-100000-FD>      ---Total Rate---
Pkts/s Max/Rx       :                  0/0                  0/0
       Max/Tx       :                  0/0                  0/0
MBits/s Rx/Tx       :                  0/0                  0/0
Broadcast           :                    0
Multicast           :                    0
Sizes 64            :                    0
      65-127        :                    0
      128-255       :                    0
      256-511       :                    0
      512-1023      :                    0
      1024-1518     :                    0
Runts/Jumbos        :                  0/0
ARP/ICMP Pkts       :                  0/0
Errors Rx/Tx        :                  0/0
Total Rx Pkts       :                    0
      Tx Pkts       :                    0
      Rx MBs        :                    0
      Tx MBs        :                    0
                    :
Pattern Type        :               abcd...
Tx Count/% Rate     :          Forever /30%
Pkt Size/Tx Burst   :             64 /   64
TTL/Port Src/Dest   :          4/ 1234/ 5678
Pkt Type:VLAN ID    :        IPv4 / TCP:0001
802.1p CoS/DSCP/IPP :             0/  0/  0
VxLAN Flg/Grp/vid   :      0000/    0/    0
IP  Destination     :           192.168.1.1
    Source          :         192.168.0.1/24
MAC Destination     :     00:00:00:00:00:00
    Source          :     40:a6:b7:18:d4:30
PCI Vendor/Addr     :     8086:1592/82:00.0

-- Pktgen 19.12.0 (DPDK 19.11.6)  Powered by DPDK  (pid:962080) ---------------



Pktgen:/>
```

MINGW64:/c/windows_dpdk/dpdk

```
Administrator@WIN-DPDK-PALLAVI MINGW64 /c/windows_dpdk/dpdk (main)
$ ./build/examples/dpdk-l2fwd.exe -l 16-17 -n 4 -v -- -p 1
```

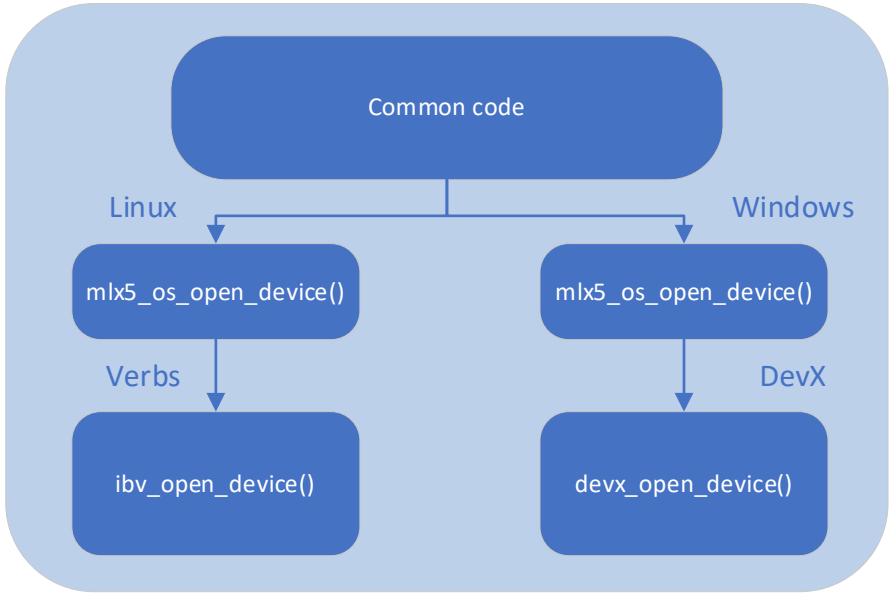
Porting the MLX5 PMD - Challenges

- MLX5 is a bifurcated PMD and requires control path communication with the kernel
  - APIs used for Linux kernel control path communication does not exist on Windows kernel (e.g. Verbs).
  - Windows kernel exposes a DevX API which is partly used in the PMD.
  - The PMD used specific Linux libraries in the PMD code (pthreads, OS system calls).

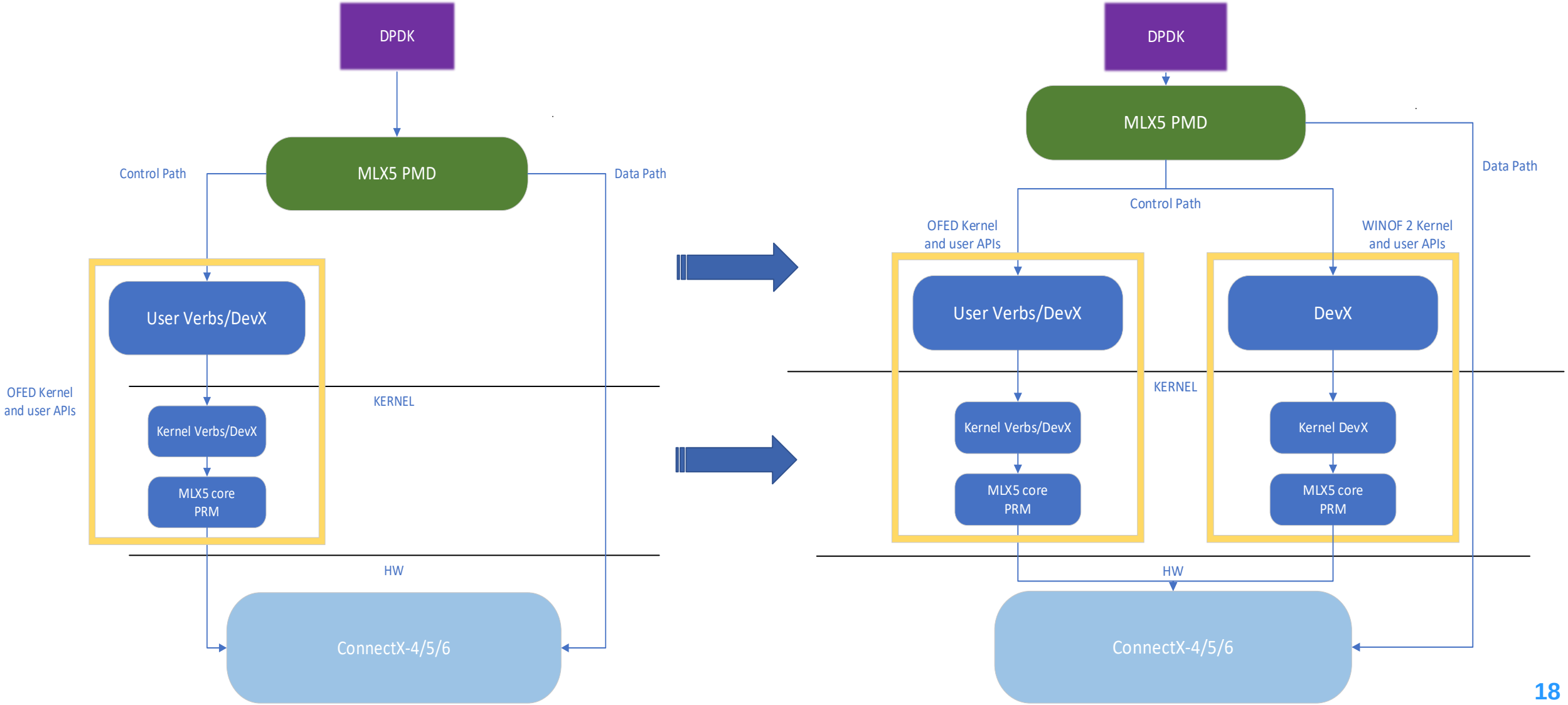


Porting the MLX5 PMD - Effort

- The effort to turn MLX5 to OS agnostic bifurcated PMD
  - Design for maximum code reuse in control path code, no modification to data-path code.
  - Move OS-specific code to isolated functions implemented for both operating systems.
  - Linux specific library calls within the PMD were replaced by EAL equivalent functions.
  - WINOF2 added supports for an official DevX installation package to allow DPDK compilation.



Porting the MLX5 PMD - Result

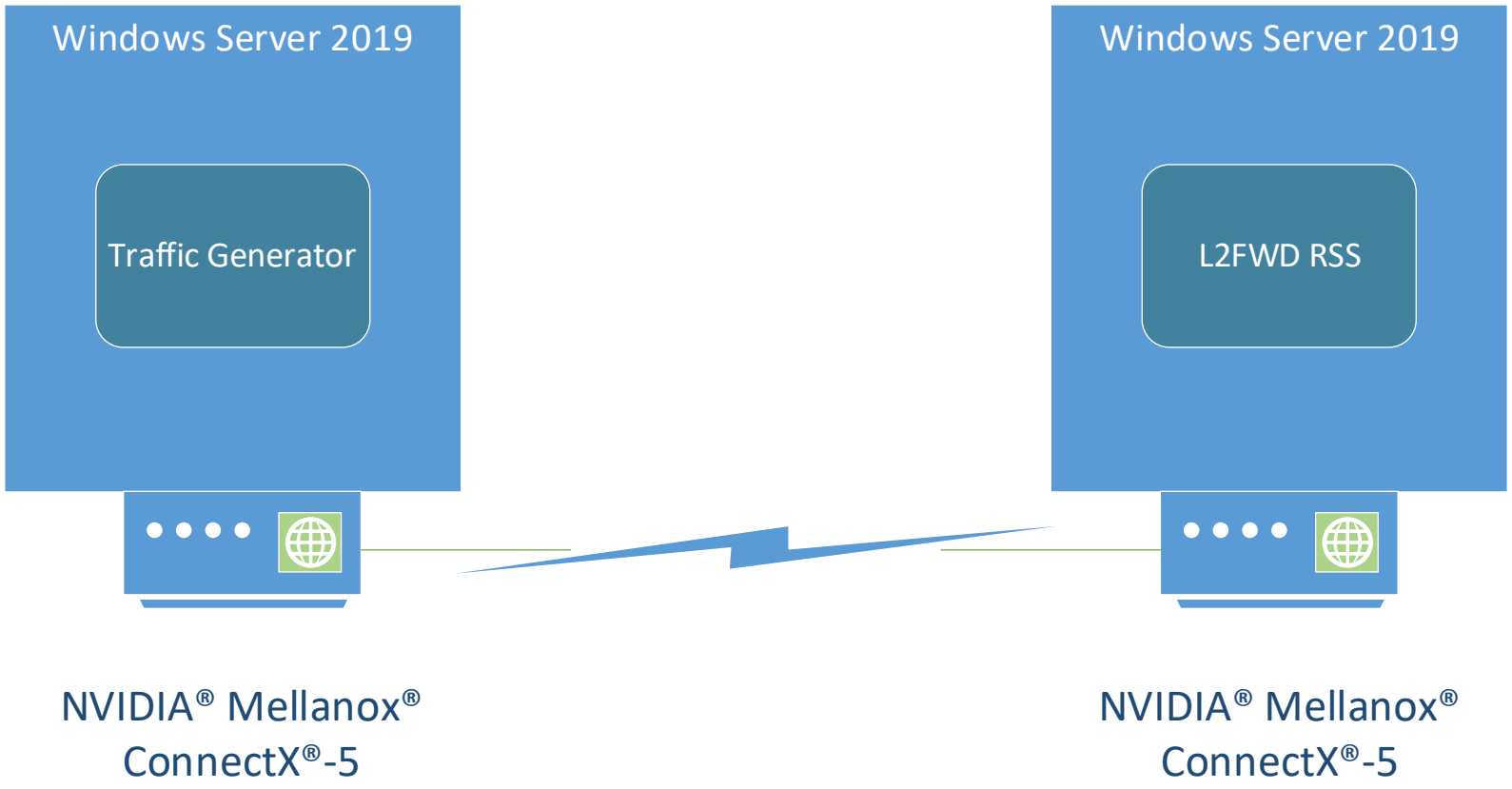


MLX5 PMD - Status and Roadmap

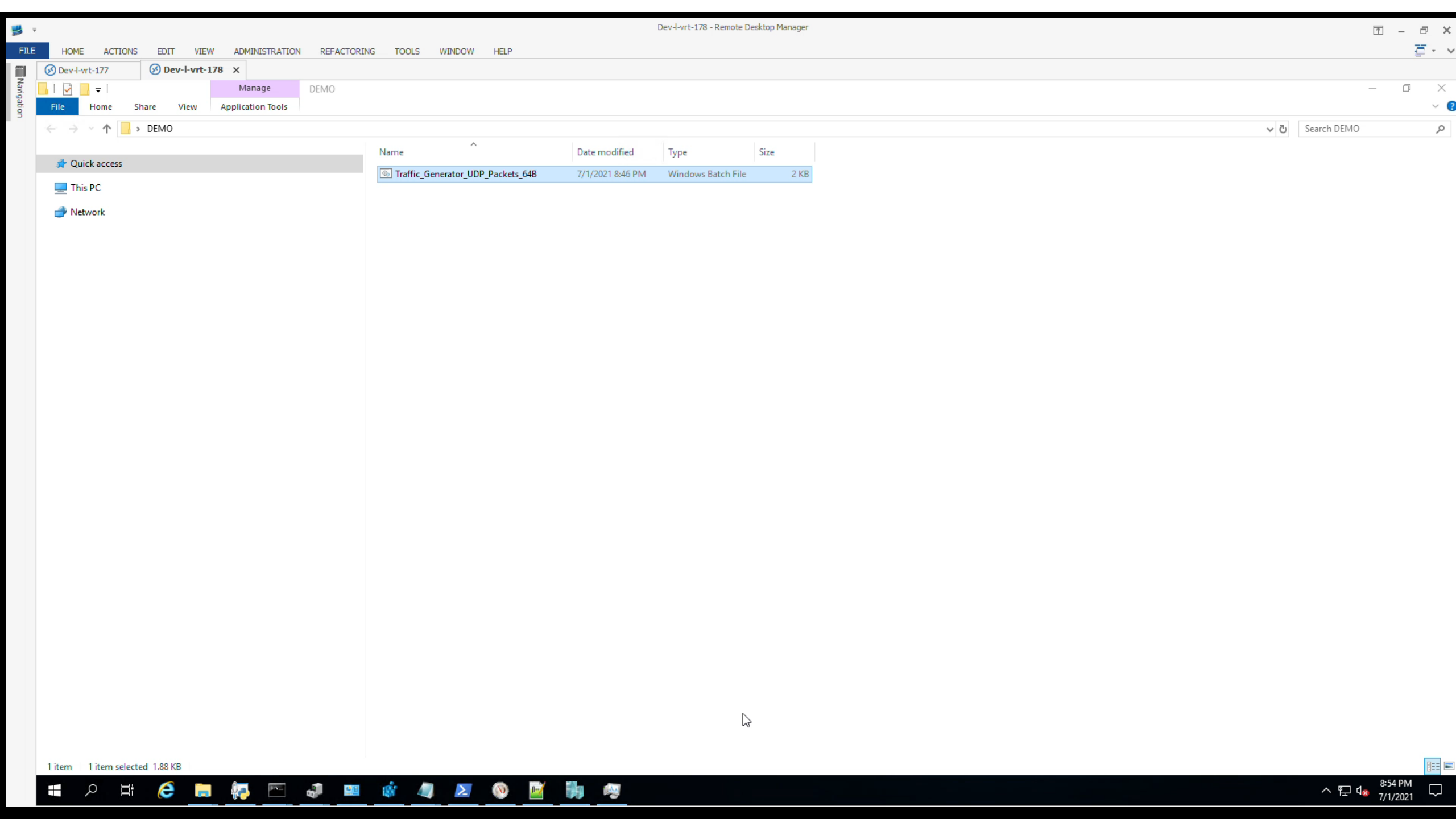
DPDK SUMMIT NORTH AMERICA

- Status
  - MLX5 on Windows was introduced on DPDK 21.02 using kernel driver WINOF2 2.60
    - Supported on all ConnectX-4 and higher Nvidia cards.
    - Supported on native and Windows virtual machines using SR-IOV.
- Roadmap
  - MLX5 offloads and abilities are enhanced in each DPDK and WINOF2 release, prioritized per customer needs.
    - DPDK 21.05
      - Additional checksum offloads.
    - WINOF2 2.70
      - Per port RSS and promiscuous mode.
    - WINOF2 2.80
      - TSO
      - CRC, VLAN and inner checksum offloads.
      - Rx jumbo frames
      - QinQ offloading
      - Packet type parsing

MLX5 PMD - Demo

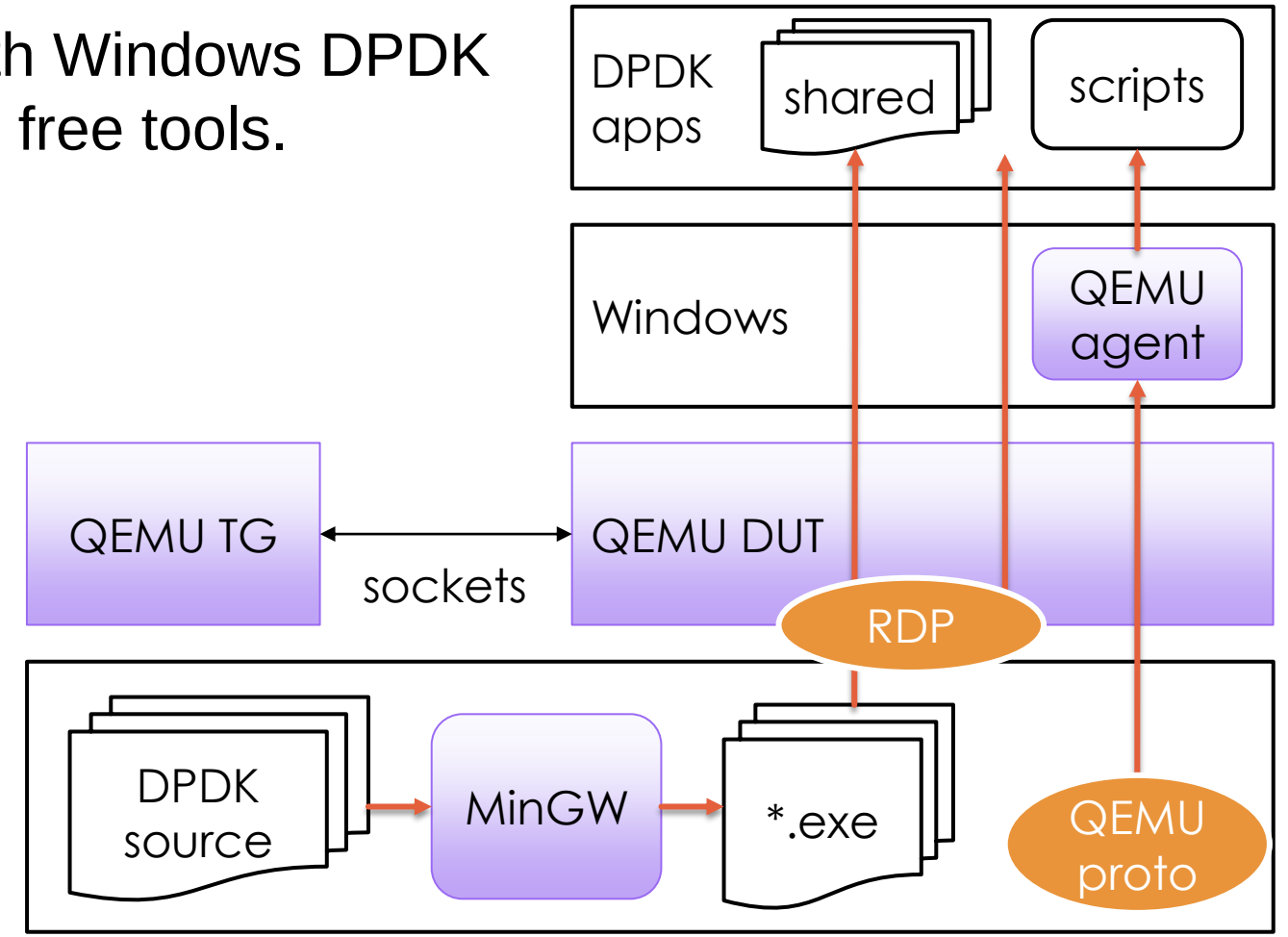


- System Configuration: Intel® Xeon® CPU E5-2697 v3 @ 2.60GHz, 28 Cores.


Dev-l-vrt-178 - Remote Desktop Manager
FILE
HOME
ACTIONS
EDIT
VIEW
ADMINISTRATION
REFACTORING
TOOLS
WINDOW
HELP
Dev-l-vrt-177
Dev-l-vrt-178
Navigation
Manage
DEMO
File
Home
Share
View
Application Tools
DEMO
Search DEMO
Quick access
This PC
Network
Name
Date modified
Type
Size
Traffic_Generator_UDP_Packets_64B
7/1/2021 8:46 PM
Windows Batch File
2 KB
1 item
1 item selected 1.88 KB
8:54 PM
7/1/2021

Feel Like $HOME

- Linux developers can work with Windows DPDK from familiar environment with free tools.
- MinGW-w64
- Free dev. VM from Microsoft.
  - Only required for drivers.
- Run in QEMU.
  - Emulate vmxnet3, e1000.
  - Run commands via agent.
- PowerShell / Python scripts



Roadmap

- Expanded HW support.
- Feature-full PMDs.
  - Enriched support for offloads
- Interface to allow NIC sharing with Windows Kernel N/W stack and DPDK app.
- Unit Tests and DTS standardized testing.
- Support for DPDK in a Windows VM.
- Support for additional compilers.

Thank You!

DPDK SUMMIT NORTH AMERICA

- **Call to action**:
  - Download, Build, and Run apps with DPDK on Windows!
  - Provide feedback and make it better!
  - Join the Windows workgroup
  - Contact dpdkwin@microsoft.com

- **Thank you, Team Behind Windows Platform Effort !**
  - **Techboard and Governing Board members**
  - **NVIDIA**: Thomas Monjalon, Tal Shnaiderman, Dmitry Kozlyuk
  - **Intel**: Ranjit Menon, Pallavi Kadam, Elizabeth Kappler, Bruce Richardson
  - **Microsoft**: Omar Cardona, Harini Ramakrishnan, Tyler Retzlaff, Narcisa Vasile, Dmitry Malloy, Jie Zhou, Khoa To
  - **SMEs**: Nick Connolly, Datapath and **many more**

DPDK

SUMMIT

NORTH AMERICA