## DPDK and Poll Mode Drivers (PMD)

When DPDK is used, Network interfaces are no more managed in Kernel space. Regular Linux NIC driver which is usually used to manage the NIC has to be replaced by a new driver which is able to run into user space. This new drive, called **Poll Mode Driver (PMD)** will be used to manage the network interface into user space with the DPDK library.

**Linux NIC drivers**

With usual Linux NIC Kernel, both NIC configuration and Packet processing is done in Kernel Space. User applications which have to establish a TCP connection or send a UDP packet is using the sockets API, exposed by libc library.

|  |  |
| --- | --- |
| *NIC configuration* | *NIC packet processing* |

Linux Packet Processing with sockets API is requiring following operations which are costly:

* Kernel Linux System calls
* Multitask context switching on blocking I/O
* Data copying from kernel (ring buffers) to user space
* Interrupt handling in kernel

With usual Linux Drivers most of operations are occurring in Kernel modes and are requiring lots of user space to kernel space context switching and interruption mechanisms. The heavy context switching usage is costing lots of CPU cycles and is a limiting the numbers of packets that a CPU is able to process. Such drivers are not able to perform packet processing at expected high speed, expecially when 10/40/100G Ethernet generation cards are used on a Linux System.

**Poll Mode Drivers**

A Poll Mode Driver consists of APIs, provided through the BSD driver running in user space, to configure the devices and their respective queues. In addition, a PMD accesses the RX and TX descriptors directly without any interrupts (with the exception of Link Status Change interrupts) to quickly receive, process and deliver packets in the user’s application.

|  |  |
| --- | --- |
| *NIC configuration* | *NIC packet processing* |

Poll Mode drivers are involved in NIC configuration They are exposing NIC configuration registers into host memory area which is directly reachable from user space.

Hence user applications can configure directly from Linux user space where they are running the NIC cards they are using. A first configuration phase is using Poll Mode Drivers and DPDK library to configure DPDK rings buffers into Linux user space.

Incoming packets will be automatically transferred with DMA mechanism from NIC physical RX queues in NIC memory to DPDK RX rings buffer in host memory. DMA is also used to transfer outgoing packets from DPDK TX rings buffer in host memory to NIC physical TX queues in NIC memory.

DMA (Direct Memory Access) offloads expensive memory operations, such as large copies or scatter-gather operations, from the CPU. DMA is a mechanism that is using a specific hardware controller to manage read and write operation into the main system memory (RAM: Random Access Memory). This mechanism which is totally independent of the central processing unit (CPU) and does not consume any CPU resource. A DMA transfer is used to manage data transfer. DMA transfer triggered by the CPU and is working in background using the specific hardware resource (DMA controller).

DPDK rings and NIC buffers are synchronized with DMA. DPDK application can access transparently to NIC packets in user space reading or writing data in DPDK rings.

## Linux drivers for PMD

Several Poll Mode drivers have been developed in Linux systems. They are using different mechanisms to perform NIC register memory mapping. They could also be supported on a limited set of NIC cards. We also have to make a difference between real physical NIC and virtual ones. Hence; DPDK applications (like Contrail vRouter) are not only used to process packets originated from a real physical NIC, but also from virtual NIC of virtual machines exposed by an hypervisor.

* PMD used to manage physical interfaces :
  + I40e PMD for Intel X710/XL710/X722 10/40 Gbps family of adapters <http://dpdk.org/doc/guides/nics/i40e.html>
  + IXGBE PMD <http://dpdk.org/doc/guides/nics/ixgbe.html>
  + Linux bonding PMD <http://dpdk.org/doc/guides/prog_guide/link_bonding_poll_mode_drv_lib.html>
* PMD used to manage virtual interfaces :
  + Virtio PMD <http://dpdk.org/doc/guides/nics/virtio.html>

In Linux user space environment, the DPDK application runs as a user-space application using the pthread library.

PCI information about devices and address space is discovered through the /sys kernel interface and through kernel modules such as:

* uio\_pci\_generic,
* igb\_uio or
* vfio-pci

Different PMDs may require different kernel drivers in order to work properly. Depending on the PMD being used, a corresponding kernel driver should be loaded and bound to the network ports. Before loading, make sure that each NIC has been flashed with the latest version of NVM/firmware.

### UIO

Supported NICs

* Intel [igb](http://dpdk.org/browse/dpdk/tree/drivers/net/e1000) (82575, 82576, 82580, I210, I211, I350, I354, DH89xx)
* Intel [ixgbe](http://dpdk.org/doc/guides/nics/ixgbe.html) (82598, 82599, X520, X540, X550)
* Intel [i40e](http://dpdk.org/doc/guides/nics/i40e.html) (X710, XL710, X722)

RHEL does not support "**uio\_pci\_generic**" driver

To enable igb\_uio driver change physical\_uio\_driver in /etc/contrail/contrail-vrouter-agent.conf file and restart supervisor-vrouter.

/etc/contrail/contrail-vrouter-agent.conf  
[DEFAULT]  
physical\_uio\_driver=igb\_uio

### VFIO

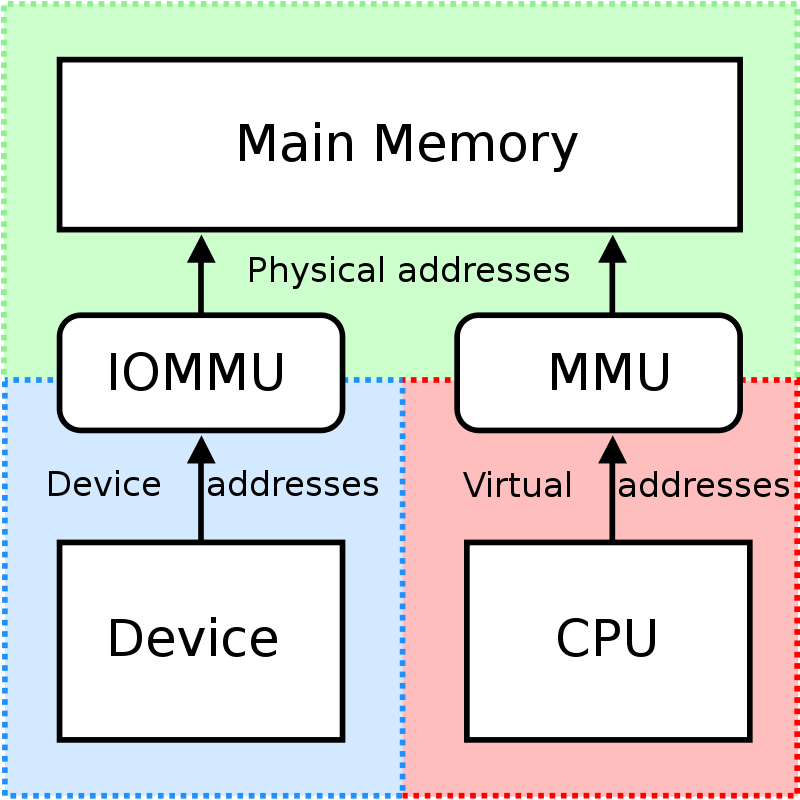
Supported NICs

* Intel [i40e](http://dpdk.org/doc/guides/nics/i40e.html) (X710, XL710, X722)

### VFIO and IOMMU

Input–Output Memory Management Unit (IOMMU) is a memory management unit (MMU) that connects a Direct Memory Access (DMA) capable I/O bus to the main memory.

In Virtualization, an IOMMU is re-mapping the addresses accessed by the hardware into a similar translation table that is used to map guest-physical address to host-physical addresses.



IOMMU provides a short path for the guest to get access to the physical device memory. Intel has published a specification for IOMMU technology as **Virtualization Technology for Directed I/O**, abbreviated as **VT-d**.

VFIO need to get IOMMU enabled :

* both kernel and BIOS must support and be configured to use IO virtualization (such as Intel® VT-d).
* IOMMU must be enabled into Linux Kernel parameters in /etc/default/grub and run update-grub command.

GRUB configuration example :

|  |
| --- |
| GRUB\_CMDLINE\_LINUX\_DEFAULT="**iommu=pt intel\_iommu=on**" |

**VFIO can be also be used without IOMMU.**

While this is just as unsafe as using UIO, it does make it possible for the user to keep the degree of device access and programming that VFIO has, in situations where IOMMU is not available.

To enable vfio-pci driver change physical\_uio\_driver in /etc/contrail/contrail-vrouter-agent.conf file and restart supervisor-vrouter.

[DEFAULT]  
physical\_uio\_driver=\*\*vfio-pci\*\*

**Drivers features compatibility list.**

[cols=",,,,",options="header",] |==== | |\*RHEL DPDK\* |\*Ubuntu DPDK\*|\*RHEL SRIOV (VF)\*\*|\*Ubuntu SRIOV (VF)\*\* |\*igb\_uio\* |No (no dkms support) |Yes (dkms) |No |Yes |\*uio\_pci\_generic\*|No (not supported by RHEL)|Yes |No |No |\*vfio\_pci\* |Yes |Yes |Yes |Yes |====

\*vRouter in parallel with SRIOV (VF support on VM)

## NICs supporing DPDK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NICs** | **Ubuntu - KVM** | **Ubuntu - DPDK** | **Redhat - KVM** | **Redhat - DPDK** | **vCenter - ESX** |
| Intel 82599/X520 "Niantic" - 10G | Yes | Yes | Yes | Yes | Yes |
| Intel X710 "Fortville" - 10G/25G/40G | Yes | Yes | Yes | Yes | Not tested |
| Broadcom bnxt 2x25G | Yes | Yes | Not tested | Not tested | Not tested |
| Mellanox 2x25G | Yes | Yes | Not tested | Not tested | Not tested |
| Netronome | 3.1.x only | No | No | No | No |

These are the guidelines from Intel with regard to Intel NIC X710 (in order not to bump into a known issue that impacts data plane)

* Do not use a single PCI NIC for non-DPDK and DPDK interfaces (having i40e kernel driver used together with i40e PMD driver for interfaces on the same PCI NIC cause problems)
* Upgrade NIC FW to 6.01 and i40e kernel driver to 2.4.6 (all necessary information you find in that document, table 9. <https://www.intel.com/content/dam/www/public/us/en/documents/release-notes/xl710-ethernet-controller-feature-matrix.pdf>
* LLDP is supported from 6.01 NIC FW but Intel also suggested to disable (ethtool -set-priv-flags <interface name> disable-fw-lldp on)

If your server manufacturer has no support for the latest firmware please contact him asap (i.e. The latest Lenovo server firmware includes 5.05 for X710 NICs)

* Redhat supports 1.6.27 i40 kernel driver version.
* Canonical supports 1.4.25 i40e kernel version.

## Discover hardware architecture topology

….