SmartNIC readout for VD-TDE

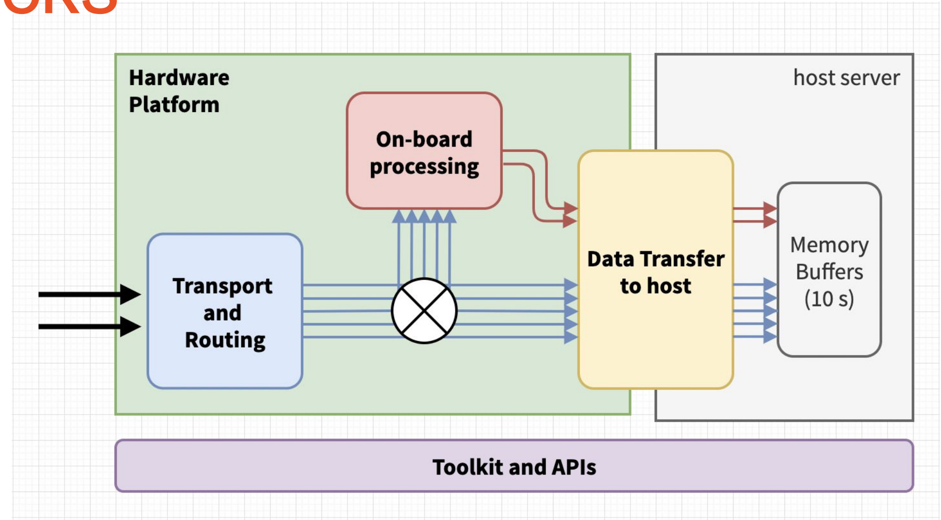
Roland Sipos
CERN

Ethernet based readout for FD-VD
2nd August 2021

Overview

- I/O device functional blocks
- SmartNIC - I/O device for VD-TDE
    - Different kinds
- Platform functionalities
    - Reception and routing
    - Interfaces to pre- and post-processing
    - API
- Summary

I/O device functional blocks



- **Hardware platform**
  - Physical network interface: QSFP+ 100Gb, Processing capabilities (FPGA), PCIe, etc.
- **Transport and routing capabilities**
  - Transport layer: UDP receiver and separation of incoming streams
- **To-Host interface**
  - DMA over PCIe: firmware, driver, software APIs
- **Pre- and post-processing interfaces**
  - *Architecture and platform specific*
- **Toolkit and API (development tools, workflow, build strategies, development model)**
  - *Architecture and platform specific*

Smart- Network Interface Controller (NIC)

- **Definition**
  - Simplest: a programmable NIC.
  - Better def.: A NIC that includes computational resources exposed to the end user, providing necessary open-source tools to utilize these resources. These resources process network traffic both on input and output of the server as well as offload the host CPU at the application level.

- **Motivation**
  - Bumping up the computing power while maintaining high bandwidth and low latency
  - Offload computationally intensive tasks from CPU (e.g.: blockchain hash comp.)
  - Number of market segments are growing, pushing the technology to evolve
    - integration with storage control
    - electronic and high-frequency trading
    - security applications, etc.

SmartNIC kinds

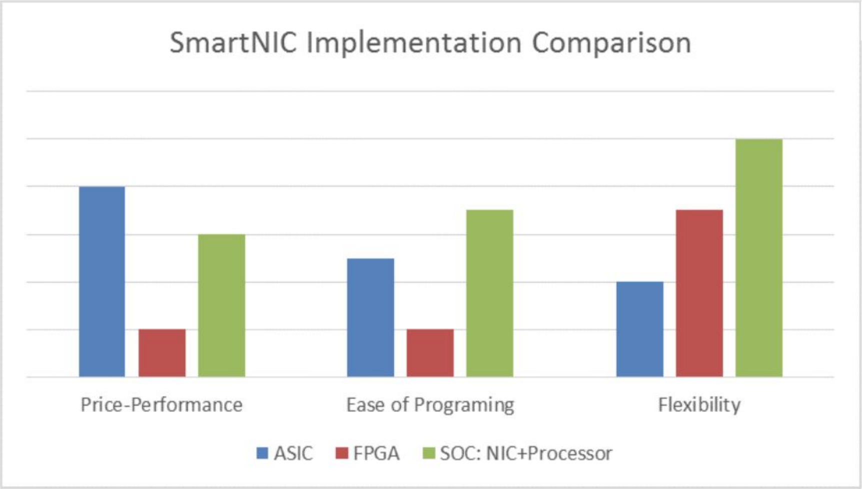
Source: https://blogs.nvidia.com/

- ASIC Based
  - Excellent price-performance
  - Vendor development cost high
  - Programmable and extensible
    - Easy to program but flexibility is limited to pre-defined capabilities

- FPGA Based
  - Good performance but expensive
  - Very difficult to program
  - Workload specific optimization

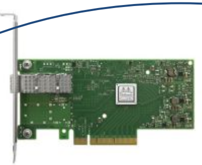
- SOC Based

System on Chip - NIC + CPU
  - Good price-performance
  - C Programmable Processors
  - Highest Flexibility
  - Easiest programmability



Source: https://blogs.nvidia.com/

SmartNIC kinds



- ASIC Based
  - Excellent price-performance
  - Vendor development cost high
  - Programmable and extensible
    - Easy to program but flexibility is limited to pre-defined capabilities

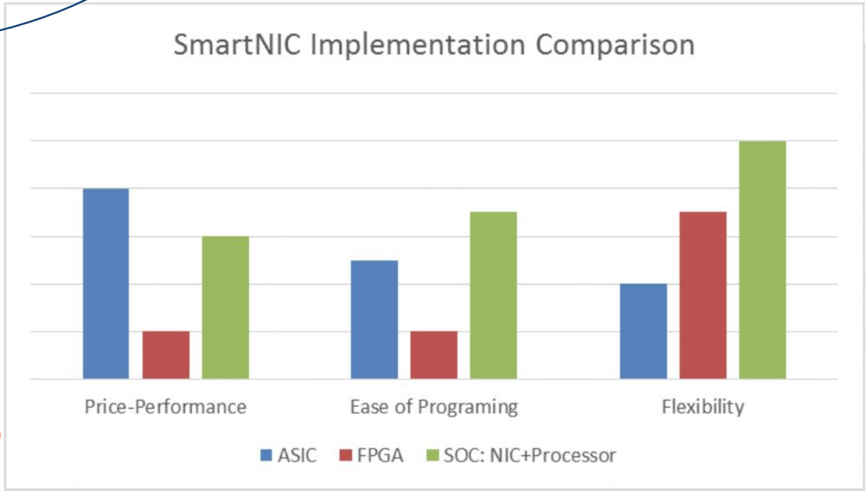
- FPGA Based
  - Good performance but expensive
  - Very difficult to program
  - Workload specific optimization

- SOC Based

System on Chip - NIC + CPU

- Good price-performance
- C Programmable Processors
- Highest Flexibility
- Easiest programmability

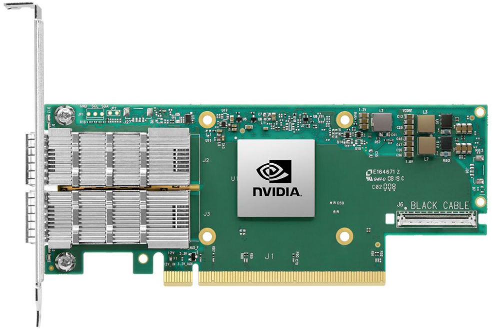
Experience from NP04: RDMA -> Infiniband over Ethernet between FELIX host and P2P connected data processing server.

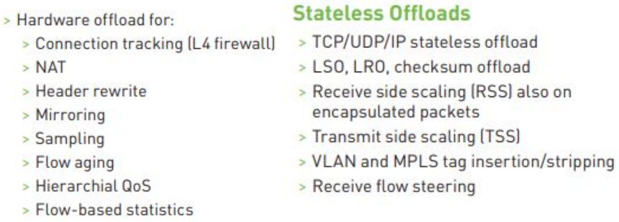


Routing and stream separation

- **Transport layer** is implemented in dedicated network chip, that also provides low level functionalities for a wide variety of features: flow steering, switching, application offload, etc.

- **Incoming data stream separation**
  Explore advanced routing capabilities and built-in mechanisms of SmartNICs and available data plane libraries and polling-mode drivers for user-space development.





> Hardware offload for:
  > Connection tracking (L4 firewall)
  > NAT
  > Header rewrite
  > Mirroring
  > Sampling
  > Flow aging
  > Hierarchial QoS
  > Flow-based statistics

Stateless Offloads

> TCP/UDP/IP stateless offload
> LSO, LRO, checksum offload
> Receive side scaling (RSS) also on encapsulated packets
> Transmit side scaling (TSS)
> VLAN and MPLS tag insertion/stripping
> Receive flow steering

<u>We will develop/test the initial VD-TDE readout (ex. TPG) on ASIC “Intelligent” NICs!</u>

Pre- and post-processing interfaces

- **Network data plane development**
  Data plane libraries and polling-mode drivers available for offloading packet processing from the kernel to processes running in user space. (Higher computing efficiency and packet throughput than is possible with interrupt-driven processing provided in the kernel.)

DPDK
DATA PLANE DEVELOPMENT KIT

- **Most probably also the interface to the FPGA based ones and emerging SmartNIC technologies**

- **This is that we already started to look at in order to have some examples and basic application to program the network data plane**

Pre- and post-processing interfaces

What we started to develop so far:

- UDP sender DAQModule
- Receiver DAQModule
- Basic benchmarking and saturating NIC ports
- Low-level test application using DPDK for packet routing and filtering

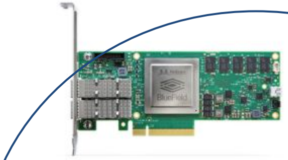
Very initial, but these are the ingredients to continue the work on the interface to the application framework.

SmartNIC kinds

Source: https://blogs.nvidia.com/

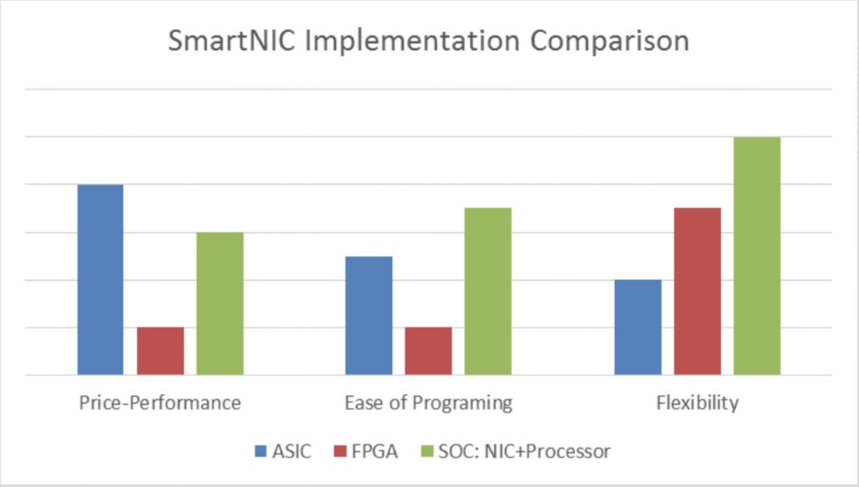
- ASIC Based
  - Excellent price-performance
  - Vendor development cost high
  - Programmable and extensible
    - Easy to program but flexibility is limited to pre-defined capabilities

- FPGA Based
  - Good performance but expensive
  - Very difficult to program
  - Workload specific optimization



- SOC Based

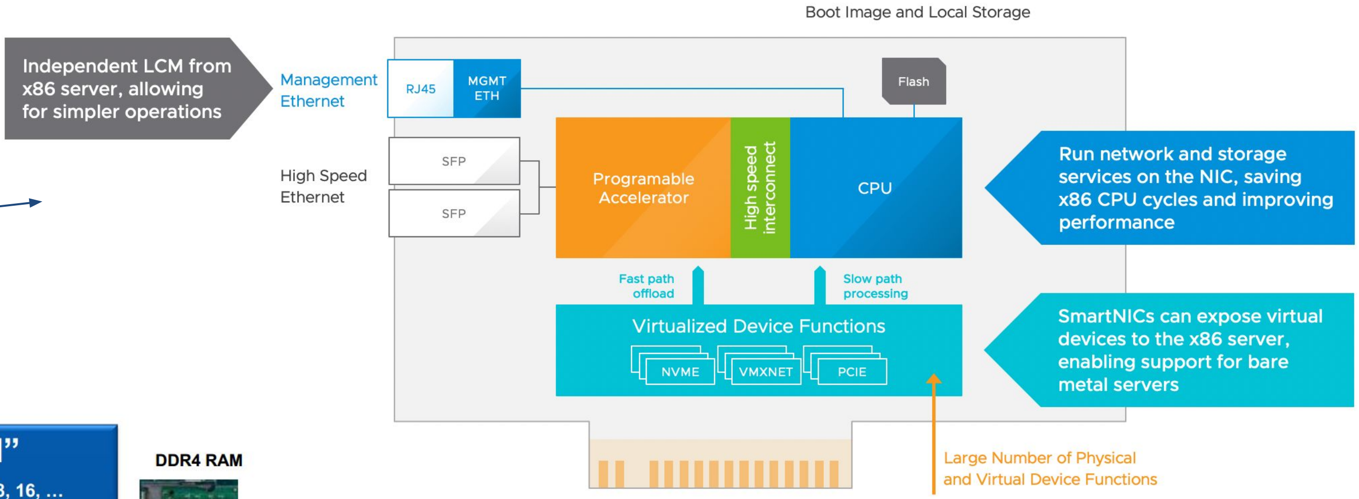
  System on Chip - NIC + CPU
  - Good price-performance
  - C Programmable Processors
  - Highest Flexibility
  - Easiest programmability



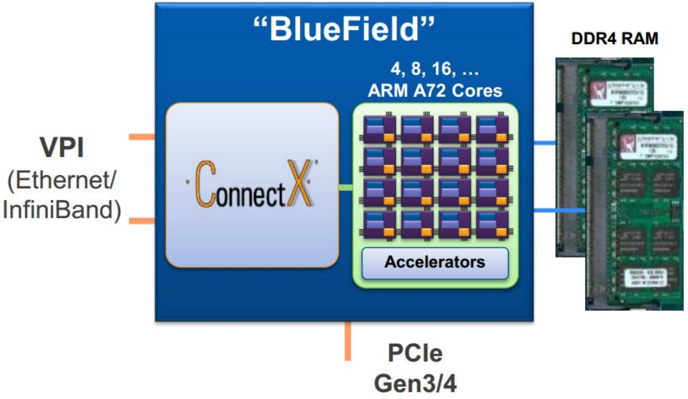
Emerging technology:
SoC SmartNIC

Possible R&D target

SOC SmartNIC architecture


Boot Image and Local Storage
Independent LCM from x86 server, allowing for simpler operations
Management Ethernet
RJ45
MGMT ETH
Flash
High Speed Ethernet
SFP
SFP
Programable Accelerator
High speed interconnect
CPU
Run network and storage services on the NIC, saving x86 CPU cycles and improving performance
Fast path offload
Slow path processing
Virtualized Device Functions
NVME
VMXNET
PCIE
SmartNICs can expose virtual devices to the x86 server, enabling support for bare metal servers
Large Number of Physical and Virtual Device Functions

Logical

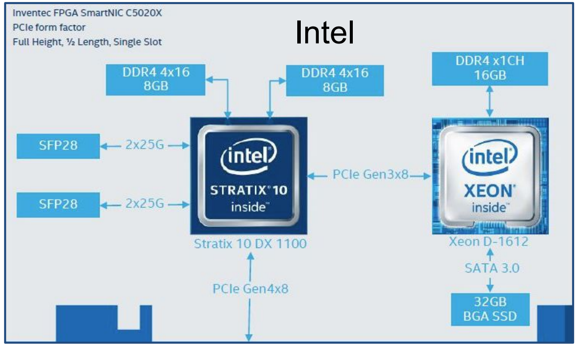


An implementation example

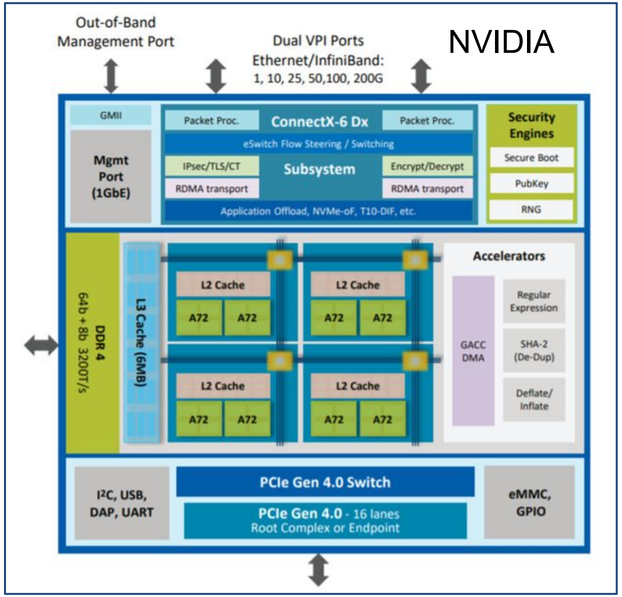
SoC Hardware platform

- **Multiple vendors**
  - Mellanox (now NVIDIA)
  - Xilinx
  - Intel
  - Broadcom

- **Versatile feature sets**, but all there what we need
  - SFP28, SFP56, QSFP28, QSFP56, DSFP...
  - PCIe Gen3, Gen4, Gen5
  - ASIC/FPGA network chip (E.g.: CX-5 or -6 or X710+)
  - CPU: ARM A72 (Armv8-A arch)



Intel



NVIDIA

From: https://www.evaluationengineering.com

Toolkits and APIs

- **Vendor specific SDKs and APIs examples**
    - Netronome Agilio P4C
    - NVIDIA DOCA
        - Industry standard APIs: DPDK, SPDK, P4

- The **need for standardization** was realized in the community, and there are some initiatives already for an “OpenSmartNIC” SDK

- We can explore how usable they are, and how easier do they make the the programming of the processing unit and the NIC

Summary

- To have the first integration with the VD-TDE readout, we will use “Intelligent” NICs with minimal programming of the network data plane

- For the longer term, SoC SmartNICs look interesting and their use case could be investigated

- The introduction of data plane development toolkits (e.g.: DPDK) will be useful irrespective of the final platform choice

End

Thank you for your attention!