# DreamBig
SEMICONDUCTOR

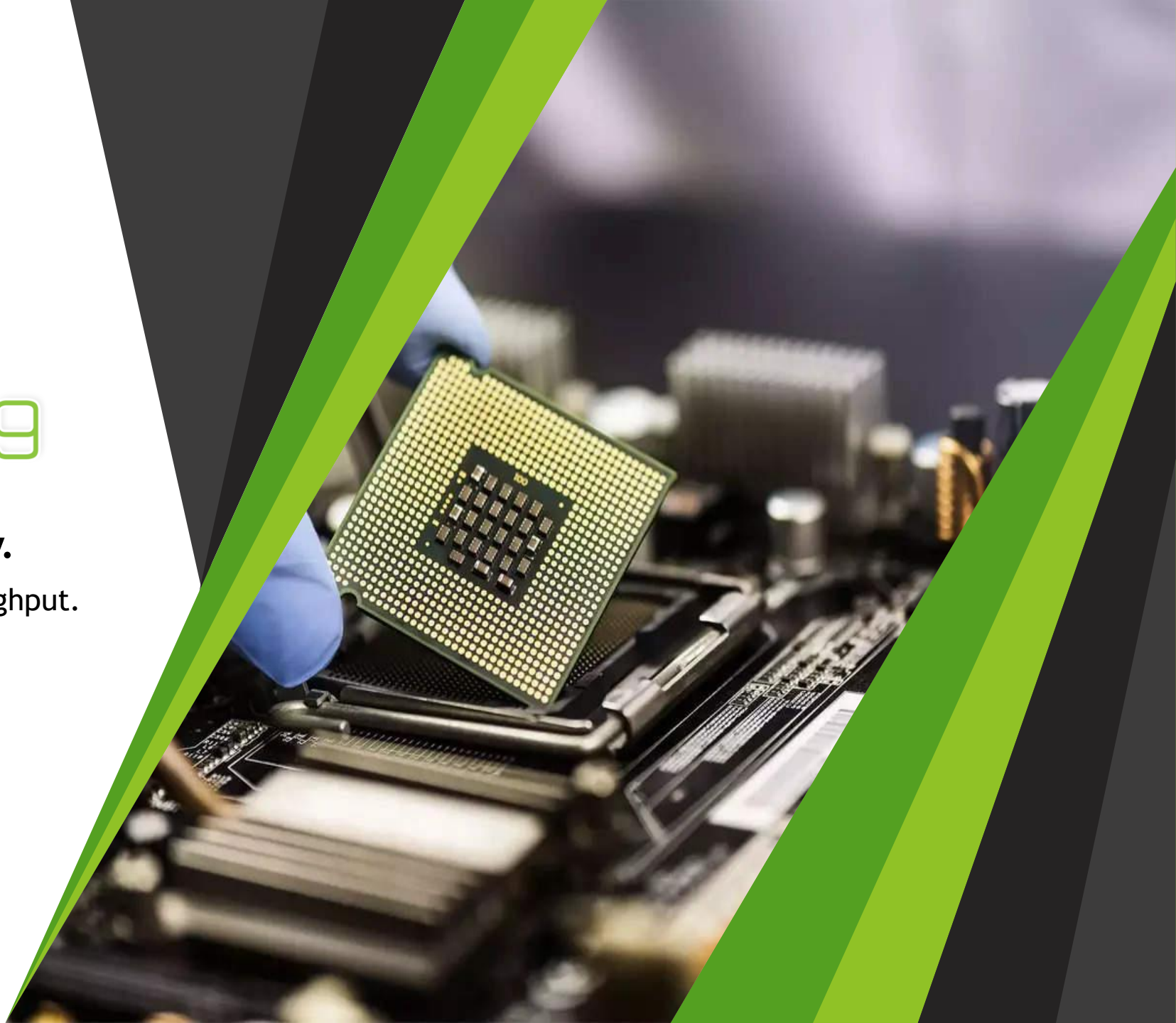## Disrupting the semiconductor industry.

Low cost, Low Latency, High Throughput.

# Importance of SmartNICs for CSP, Super Computing and AI

- Introduction
- What does offload mean
- What is a SmartNIC?
- Different offloads
- How the offload relates to different use case?
- Use cases – CSP, AI, HPC

# Founding team with stellar credentials

## The Semiconductor Founding Team

Team of top innovators of Semiconductor industry with proven track record and billion dollar exits.

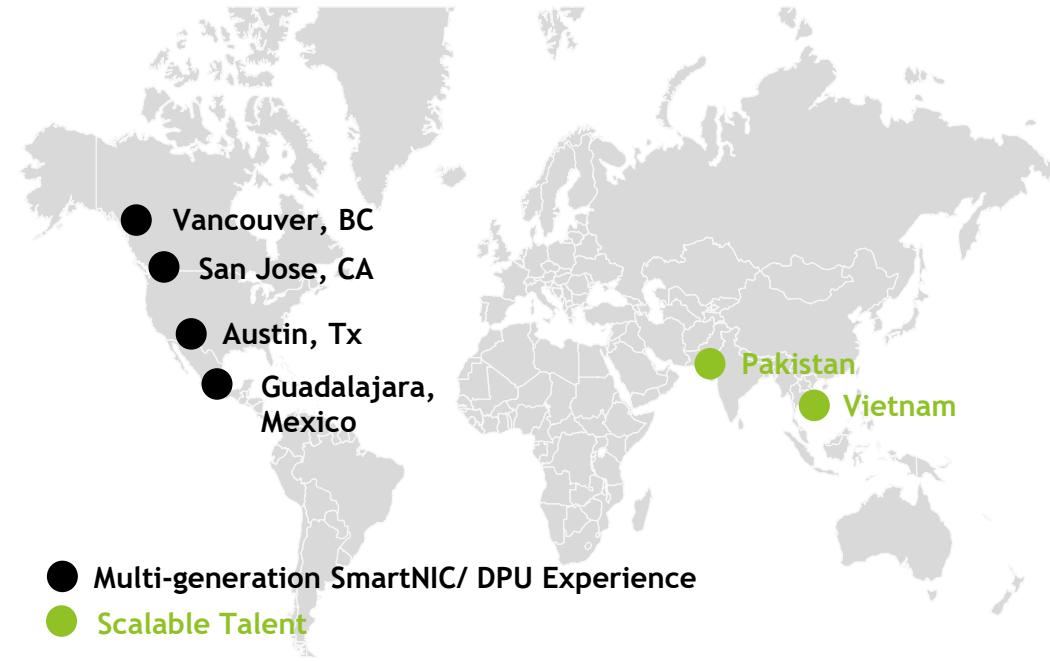**Sehat Sutardja**
Founder, Executive Chairman

**Weili Dai**
Founder, Executive vice-chairwoman

**Sohail Syed**
Founder, CEO
(NED Alumnus)

Sehat & Weili were the founders of Marvell Semiconductor MARVELL

Sohail is a Serial Entrepreneur with multiple exits

- Vancouver, BC
- San Jose, CA
- Austin, Tx
- Guadalajara, Mexico
- Pakistan
- Vietnam

● Multi-generation SmartNIC/ DPU Experience
● Scalable Talent

DreamBig **core team is formed of domain experts** required to be a leader in SmartNIC/DPU including LAN, RDMA, and NVMe.
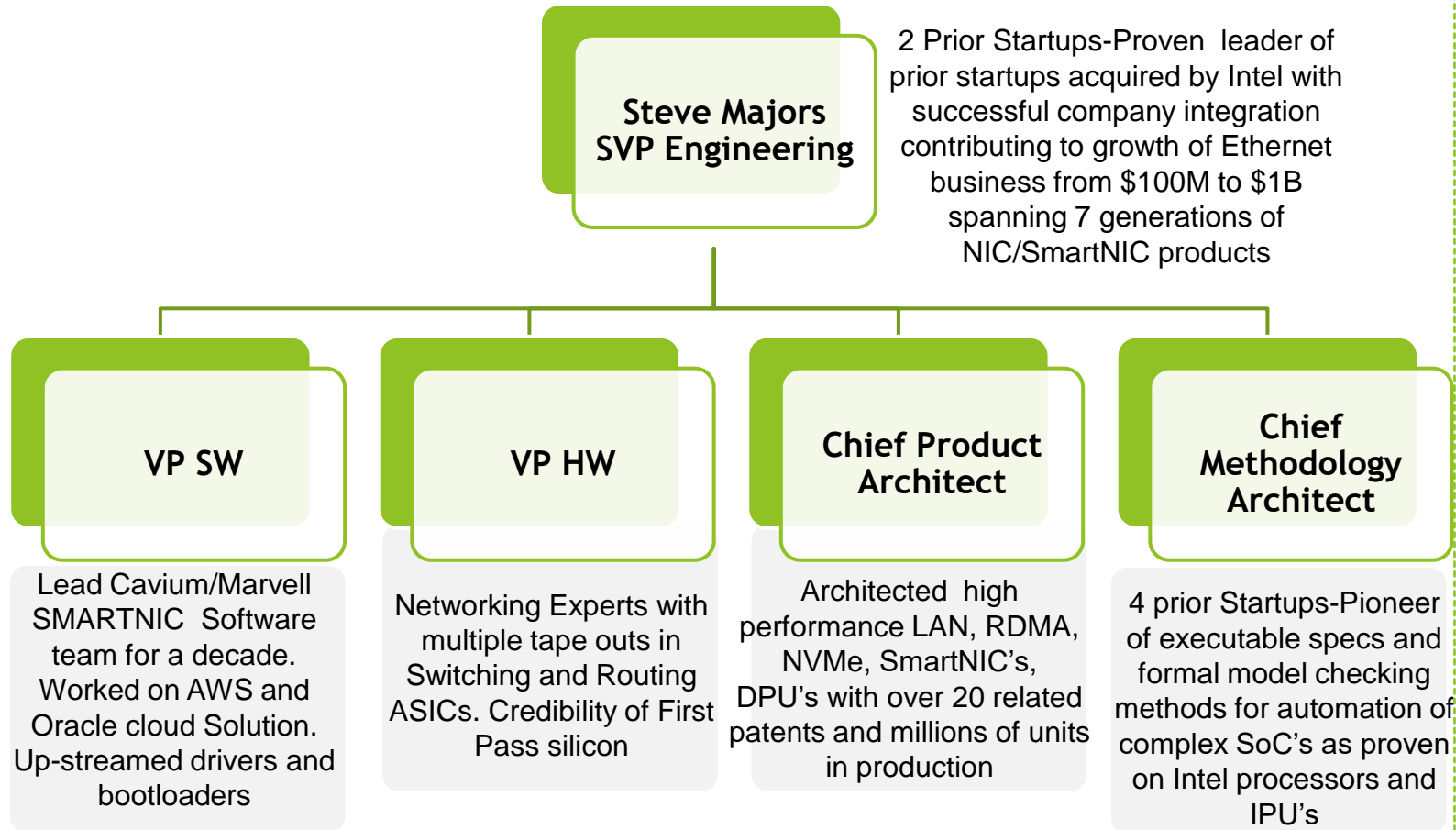
After developing multiple generations, the lessons, successes, and tight team bonds have been **forged to solve the next set of challenges** at 400/800Gbps and beyond.

Our scalable offshore talent is **strategically located in geographies with low turnover** enhancing predictable product execution.

# 170+ Strong Engineering Team led by Key market leaders.. with 400+ years Experience

## Steve Majors
### SVP Engineering

2 Prior Startups-Proven leader of prior startups acquired by Intel with successful company integration contributing to growth of Ethernet business from $100M to $1B spanning 7 generations of NIC/SmartNIC products

### VP SW

Lead Cavium/Marvell SMARTNIC Software team for a decade. Worked on AWS and Oracle cloud Solution. Up-streamed drivers and bootloaders

### VP HW

Networking Experts with multiple tape outs in Switching and Routing ASICs. Credibility of First Pass silicon

### Chief Product Architect

Architected high performance LAN, RDMA, NVMe, SmartNIC's, DPU's with over 20 related patents and millions of units in production

### Chief Methodology Architect

4 prior Startups-Pioneer of executable specs and formal model checking methods for automation of complex SoC's as proven on Intel processors and IPU's

## Overall profile

Team members with experience with **industry leaders**

MARVELL   BROADCOM   CISCO   juniper NETWORKS
CAVIUM   intel

**60+ experts in North America**
- 8 Architecture,
- 10 Design,
- 15 Verification,
- 10 SW/FW,
- 8 System Engineers

**100+ offshore engineers** with mix of design, verification, and SW skills

**Combined experience strength of**
- 250yrs Ethernet Networking,
- 100yrs RDMA,
- 50yrs NVMe product development
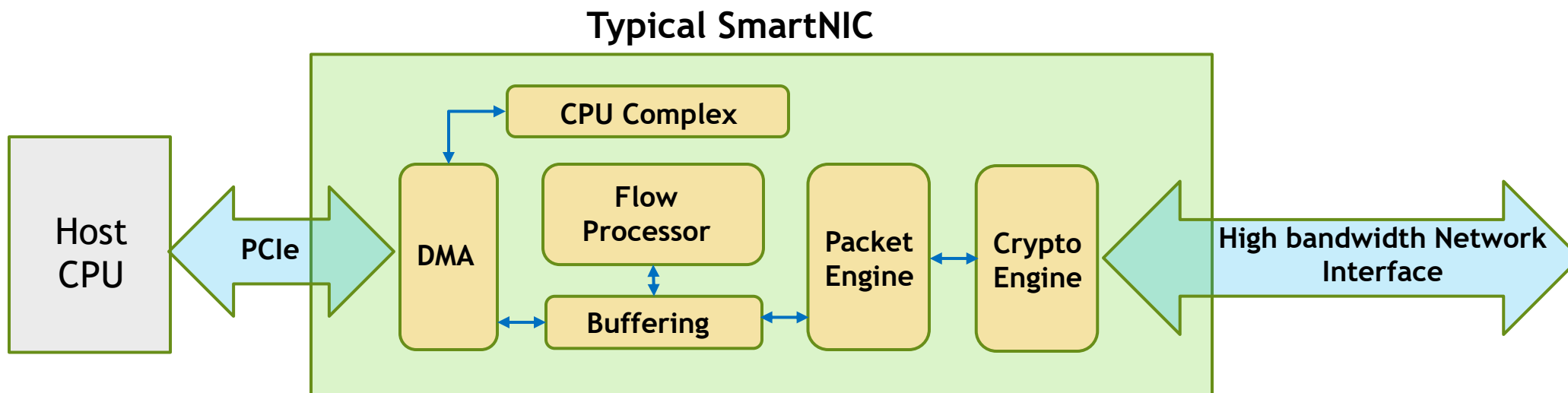- millions of units in production,
- and growing…

# Network Onload vs Offload

- Onload:
  - Host CPU manages and executes all networking operations.
  - With higher networking loads, the CPU cycles are consumed by these complex networking operations.
  - Less CPU is available for the application – which is the primary task for a CPU.
  - Performance bottleneck!

- Offload:
  - Overcomes onload performance bottlenecks by performing complex network operations on a NIC card (more precisely, SmartNIC!).
  - This results in more CPU available for the real applications – improving the overall efficiency of the system.

ONLOAD

Network

OFFLOAD

10Gbps

10Gbps

Regular NIC
(No offload)

SmartNIC
(Offload)

100 %

11 %
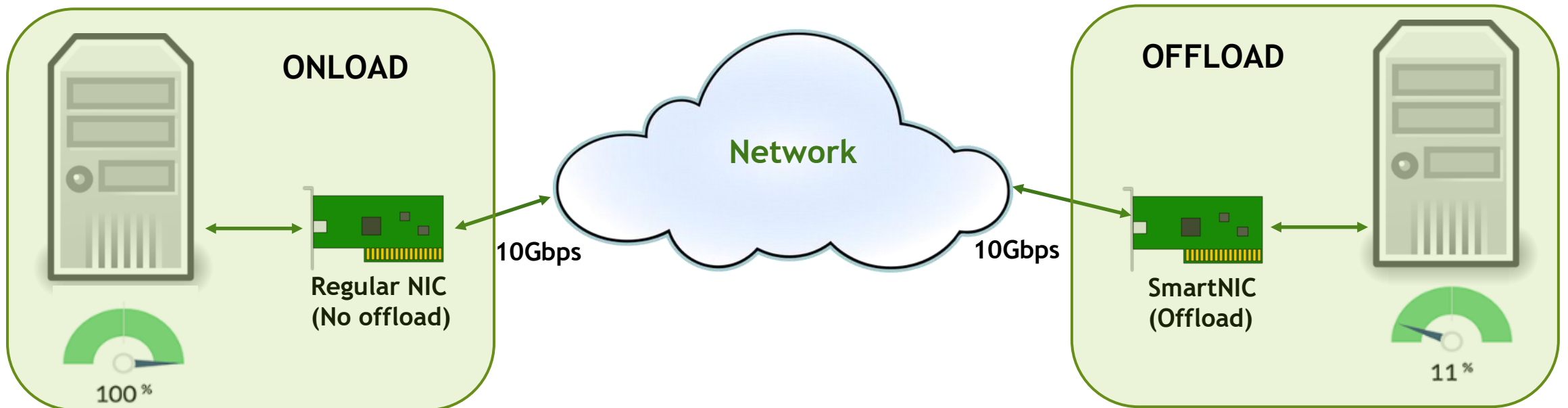
# What is a SmartNIC?

- SmartNICs are enhanced NICs which can offload complex workloads from the host CPU.
  - Some of these workloads are security, isolation, switching, complex packet processing, QoS, (Quality of Service), storage.
  - This enable the host CPU to efficiently use its cycles for executing the application.

- SmartNICs consists of specialized blocks (also knows as accelerators) which are capable of executing specific packet processing tasks much more efficiently than a general-purpose CPU.

- They are complex!

**Typical SmartNIC**

Host CPU ←→ PCIe ←→ DMA → CPU Complex, Flow Processor, Buffering → Packet Engine ←→ Crypto Engine ←→ High bandwidth Network Interface
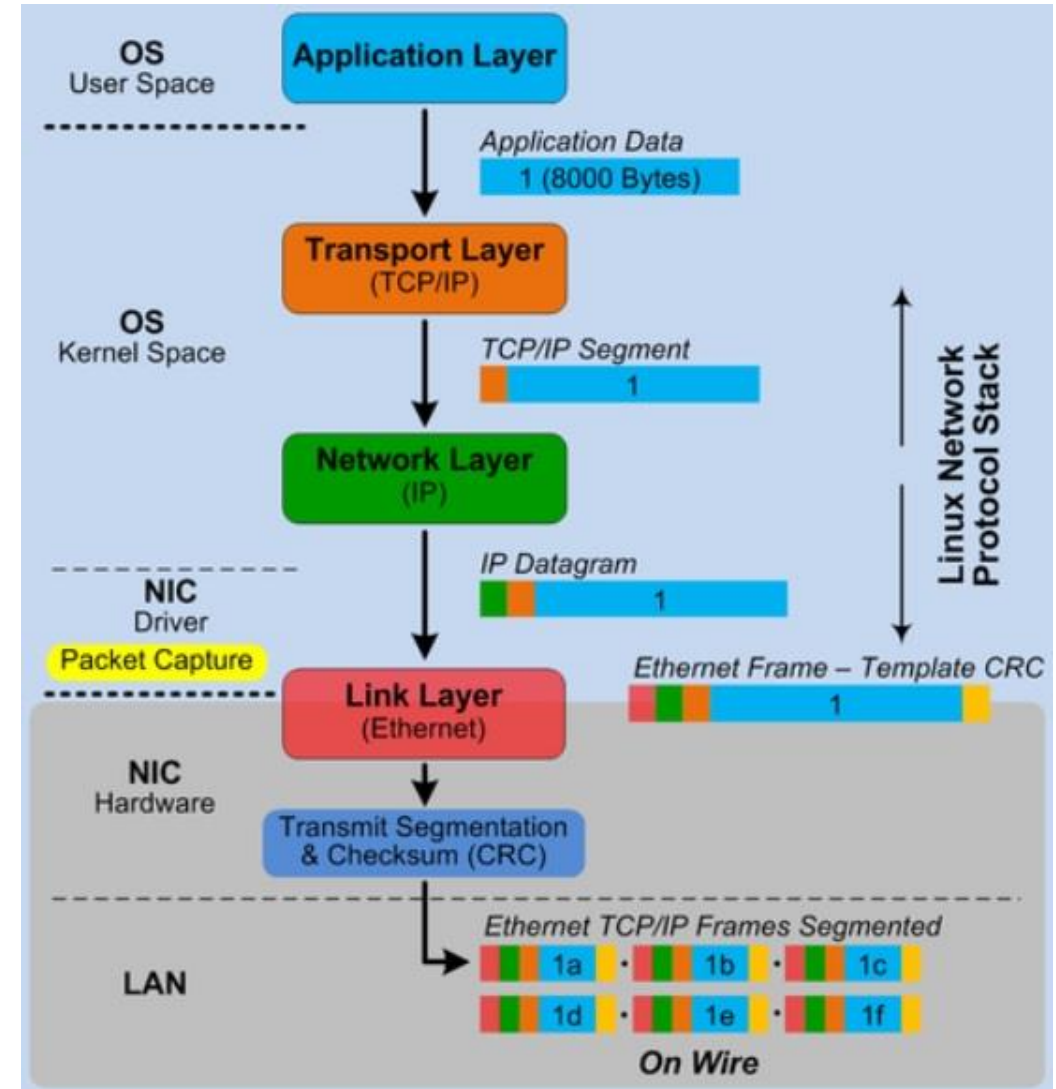
# DreamBig Designs and Develops SmartNICs

- Why SmartNIC?

  - NIC (Network Interface Card) connects a computer to a network.
  - Data bandwidth requirements are increasing exponentially with the emergence of Cloud Networking.
  - More data requires more host CPU cycles to process, resulting in less cycles for real applications.
  - Application workloads are getting complex; requiring even more CPU cycles.
  - Demand of security and isolation is also increasing.
  - A traditional NIC cannot cope with these requirements – this requires a new NIC technology; "SmartNIC".

- SmartNIC provides inline packet processing capabilities and offloads complex host CPU operations:

  - SmartNIC offloads security, isolation, switching, complex packet processing, QoS, (Quality of Service), storage from the host CPU.
  - IO virtualization provides workload isolation.
  - *Host CPU gets more cycles for the real work – executing applications.*

# SmartNIC Offloads (Stateless Offloads)

**LSO (Large Send Offload):**

- Outgoing packet segmentation takes host CPU cycles (e.g. segmenting a large TCP payload into smaller packets). NIC can perform this segmentation which saves CPU cycles!
- Linux TCP stack sends a large TCP packet to the NIC (much bigger than MSS).
- NIC performs the segmentation of this data based on the MSS value, adds TCP, IP and Ethernet headers to each segmented packet and sends the data on the wire.
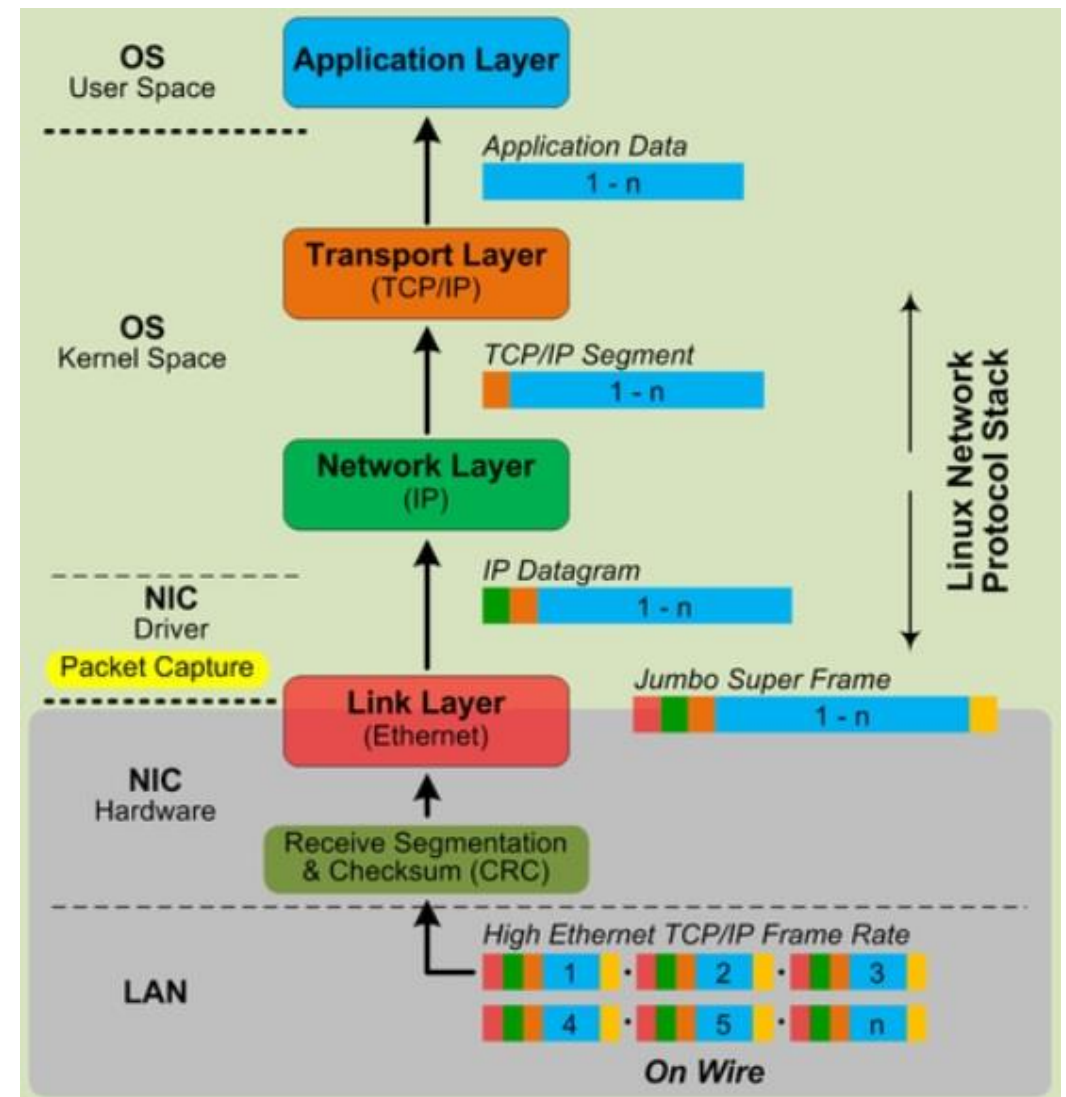


Diagram courtesy of https://the-linux-channel.the-toffee-project.org/index.php?page=8-links-network-packet-processing-hardware-offload&lang=en

# SmartNIC Offloads (Stateless Offloads)
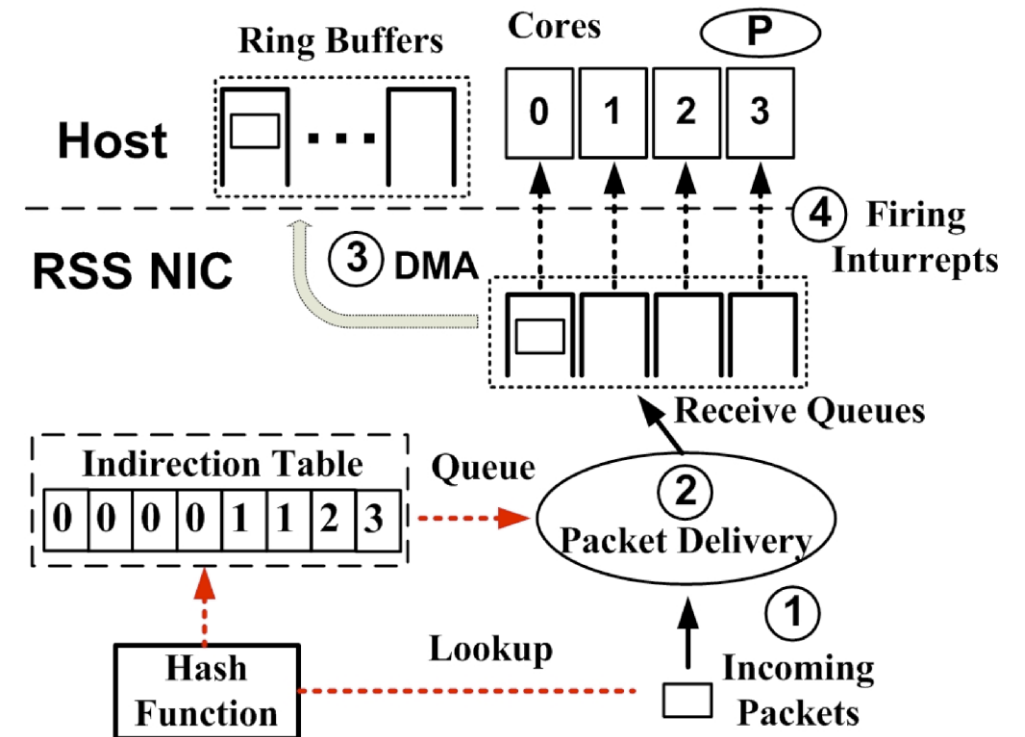
**LRO (Large Receive Offload):**

- Large Receive Offload (LRO) reduces the CPU overhead for processing packets that arrive from the network at a high rate.
- LRO reassembles incoming network packets into larger buffers and transfers the resulting larger but fewer packets to the host network stack.
- The CPU has to process fewer packets than when LRO is disabled, which reduces its utilization for networking especially in the case of connections that have high bandwidth.
- LRO is used for TCP.



Diagram courtesy of https://the-linux-channel.the-toffee-project.org/index.php?page=8-links-network-packet-processing-hardware-offload&lang=en

# SmartNIC Offloads (Stateless Offloads)

**RSS (Receive Side Scaling):**

- Receive side scaling (RSS) is a network driver technology that enables the efficient distribution of network receive processing across multiple CPUs in multiprocessor systems.
- A NIC uses a hashing function to compute a hash value over a pre-defined header fields of the received packet.
- A number of least significant bits (LSBs) of the hash value are used to index an indirection table.
- A host receive queue is selected based on this index.
- RSS improves the overall system performance by reducing:
  - Processing delays by distributing receive processing from a NIC across multiple CPUs.
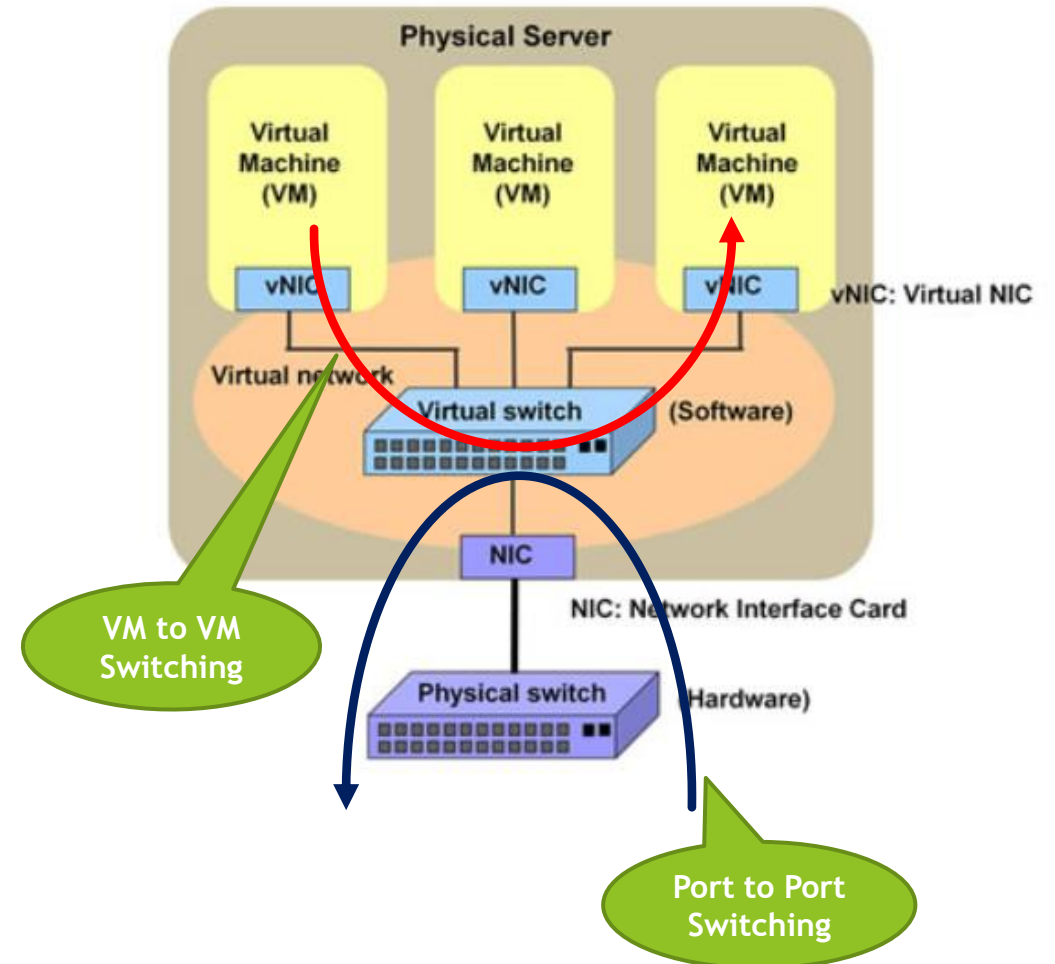  - Cache overheads since an incoming flow will be processed by the same core.

Diagram courtesy of "A Transport-Friendly NIC for Multicore/Multiprocessor Systems" by Wenji Wu, Phil DeMar, Matt Crawford

# Complex Offloads (Security)

- Data security is critical!
  - ✓ Packets on the wire should be encrypted – crypto operations are expensive.
  - ✓ SmartNIC have crypto accelerators which can perform expensive crypto operations.

- MACSec (L2) and IPsec (L3) and are two commonly used packet security protocols.
  - ✓ Both provides confidentiality, data integrity and authentication.

- Host CPU fully offloads these protocol processing to the SmartNIC.
- These offloads are done inline. Host only sees plain packets (Linux also supports a hybrid mode which involved CPU involvement for the offload).
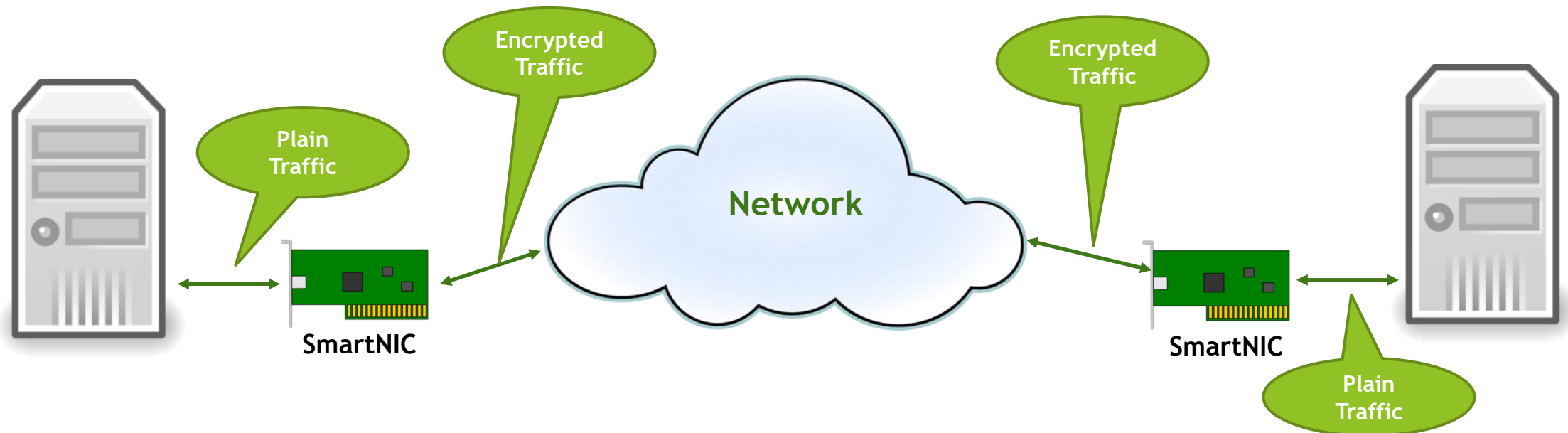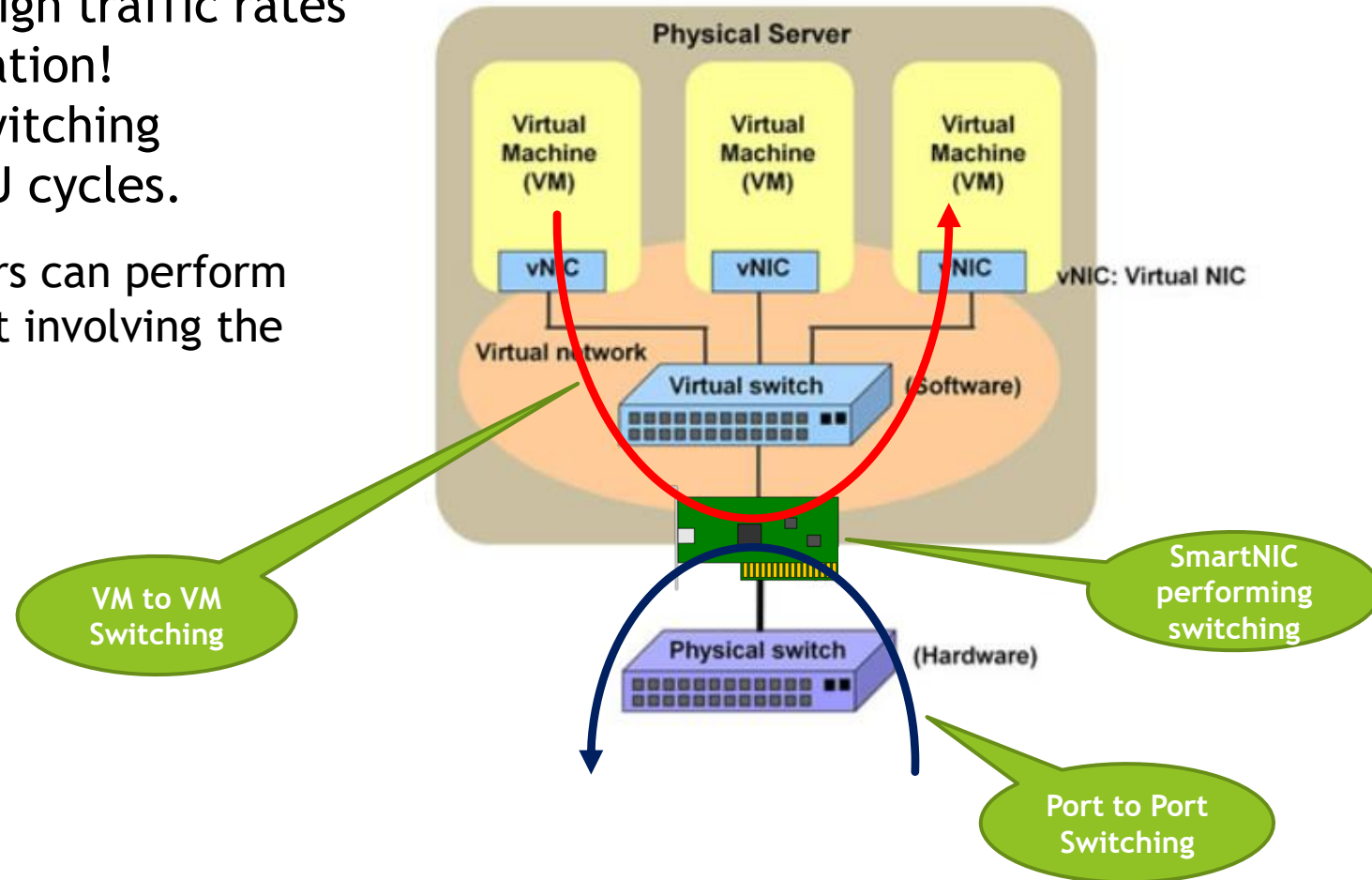
# Complex Offloads (Switching)

- Packet routing and switching can be done by the host.
- With the emergence of virtualization and Virtual Machines (VMs), the switching is becoming very common.
  - ✓ Two VMs can talk to each through a software switch.
- OVS (Open vSwitch) is a commonly used software switch.
  - ✓ Flexible switch which allows the switching to be done based on various packet header fields.
  - ✓ If can also perform set of actions before the switching (e.g. header manipulation, insertion/removal of VLANs etc).



Diagram courtesy of https://www.fujitsu.com/global/about/resources/news/press-releases/2010/0610-02.html
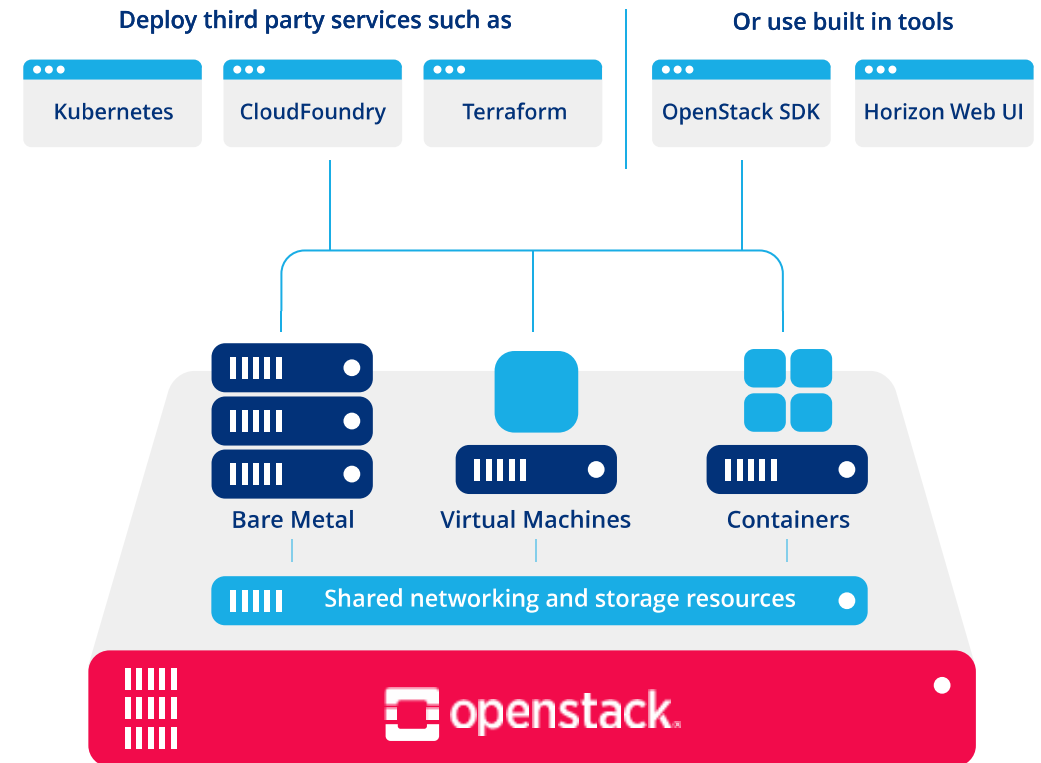
# Complex Offloads (Switching- cont...)

- Software based switching at high traffic rates results in high host CPU utilization!
- SmartNIC can offload these switching operations to free up host CPU cycles.
  - ✓ Dedicated lookup accelerators can perform high traffic switching without involving the host CPU.
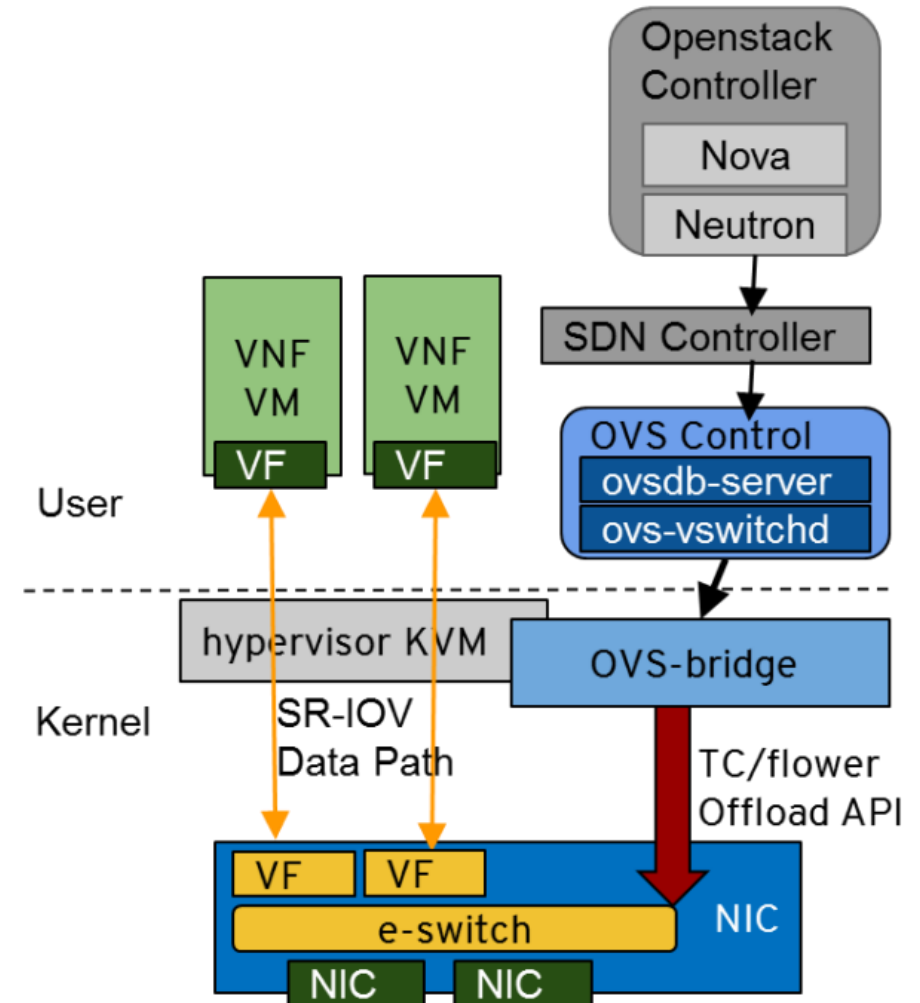


**Physical Server**

Virtual Machine (VM)

Virtual Machine (VM)

Virtual Machine (VM)

vNIC

vNIC

vNIC

vNIC: Virtual NIC

Virtual network

Virtual switch (Software)

VM to VM Switching

SmartNIC performing switching

Physical switch (Hardware)

Port to Port Switching

Diagram courtesy of https://www.fujitsu.com/global/about/resources/news/press-releases/2010/0610-02.html

# Where The Offload Works – CSP

- OpenStack
  - Open source cloud computing platform
  - Enables Infrastructure as a Service IaaS)
  - Can create bare metal instances, virtual machines and containers
  - Offers networking, storage, processing segregation
  - Automotive, healthcare, finance, e-commerce, telco are some of the 40 million plus deployed cores in different industries
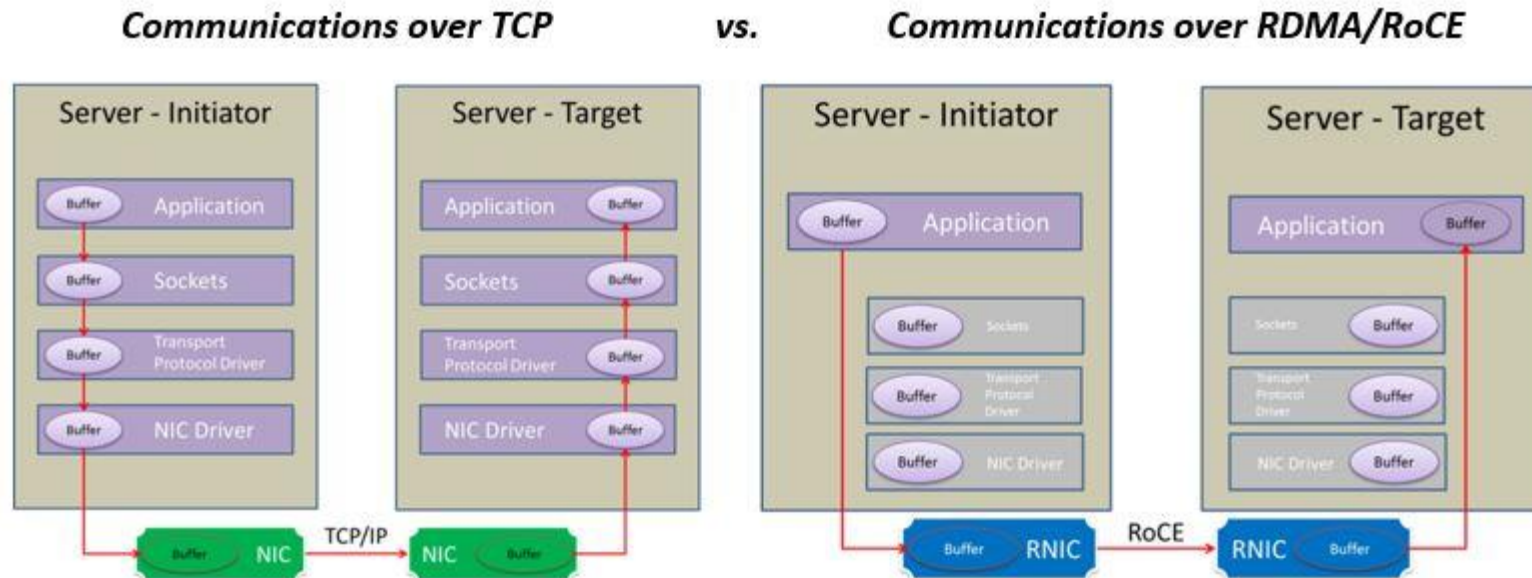


Diagram courtesy of https://www.openstack.org/

# Where The Offload Works – CSP

- Putting it together

  - Neutron is the Open Stack networking component

  - Integrates with OVS for layer-2 layer 3 functionality

  - Supports OVS rules offloading to SmartNIC



Diagram courtesy of https://resources.nvidia.com/en-us-accelerated-networking-resource-library/accelerating-cloud-ready

# Complex Offloads (RDMA)

- RDMA = Remote Direct Memory Access

- RDMA is the technology which allows a network host to access main memory of another host without involving the CPU.

- It improves data throughput and performance and frees up CPU and resources.
  - ✓ This results in higher data transfer rates and lower latencies.
  - ✓ It supports zero-copy operation by allowing the NIC to copy data directly from the wire to the application memory or from application memory to the wire – no data copy between application memory and kernel buffers.

- **SmartNIC fully offloads the RDMA protocol!**



Diagram courtesy of https://developer.nvidia.com/blog/doubling-network-file-system-performance-with-rdma-enabled-networking/

# Where The Offloads Work – AI (Meta)

- RDMAoverEthernet for Distributed AI Training at Meta Scale

    - https://dl.acm.org/doi/pdf/10.1145/3651890.3672233

- Backend network: The BE is a specialized fabric that connects all RDMANICs in anon-blocking architecture, providing high bandwidth, low latency, and lossless transport between any two GPUs in the cluster, regardless of their physical location. This backend fabric utilizes the RoCEv2 protocol, which encapsulates the RDMA service in UDP packets for transport over the network.
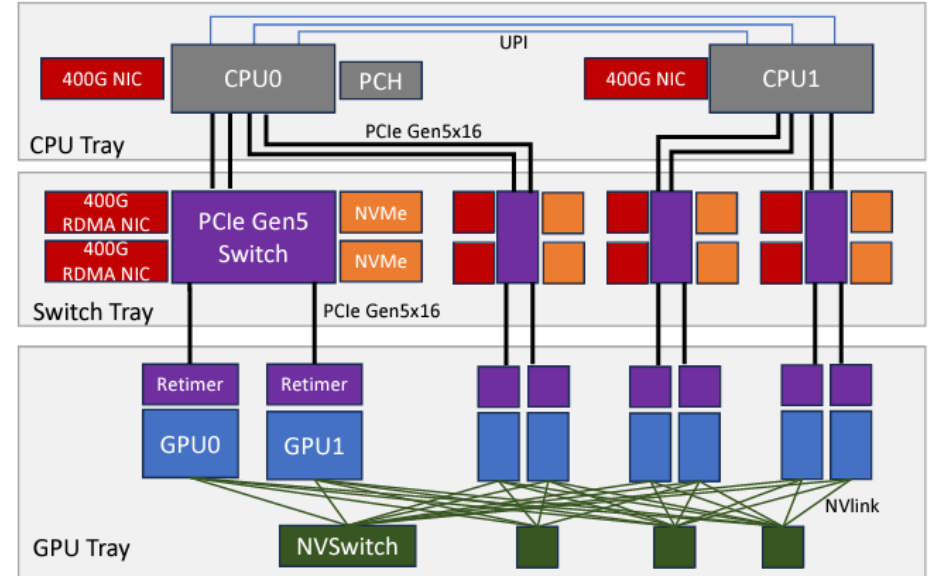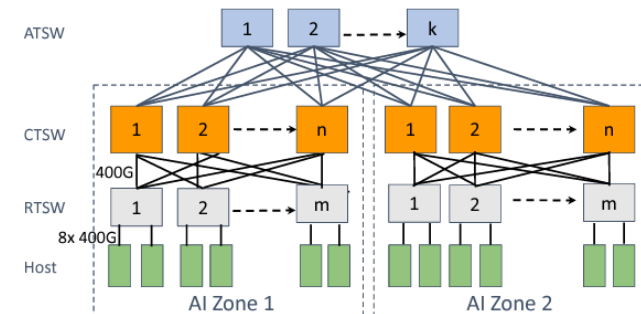


Figure 4: Grand Teton platform



Figure 6: Backend Network Topology

# Where The Offloads Work – AI (Meta)

- RDMAoverEthernet for Distributed AI Training at Meta Scale

  - https://dl.acm.org/doi/pdf/10.1145/3651890.3672233

- For larger jobs, RDMA NICs enable GPUDirect technology, so that GPU-to-GPU traffic can bypass host and host memory bottlenecks.

- GPUDirect RDMA provides direct communication between NVIDIA GPUs in remote systems. This eliminates the system CPUs and the required buffer copies of data via the system memory, resulting in 10X better performance.
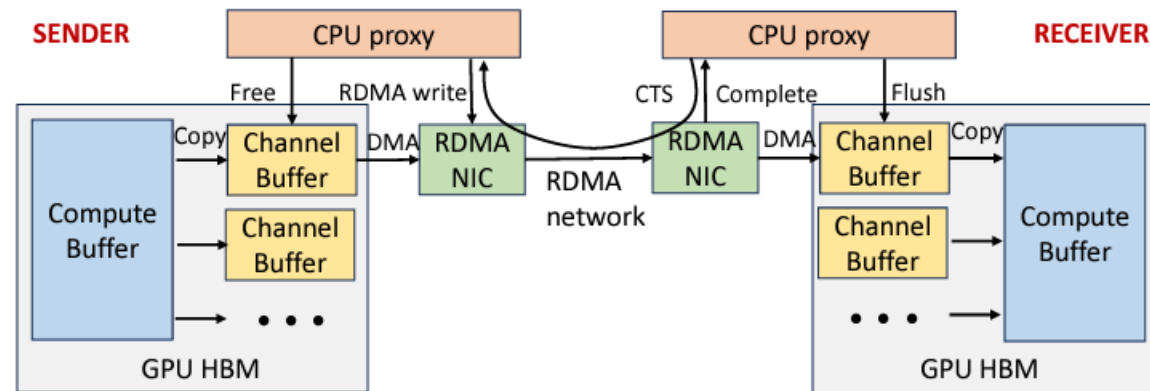


Figure 14: GPU to GPU communication architecture.

# Where The Offload Works – HPC

- Swift-X: Accelerating OpenStack Swift with RDMA for Building an Efficient HPC Cloud

  - https://shashankgugnani.github.io/publications/ccgrid_17.pdf

- The OpenStack Object Store project, known as **Swift**, offers cloud storage software so that you can store and retrieve lots of data with a simple API

- Introduced an RDMA-based communication module in the client, object server and proxy server for low latency communication
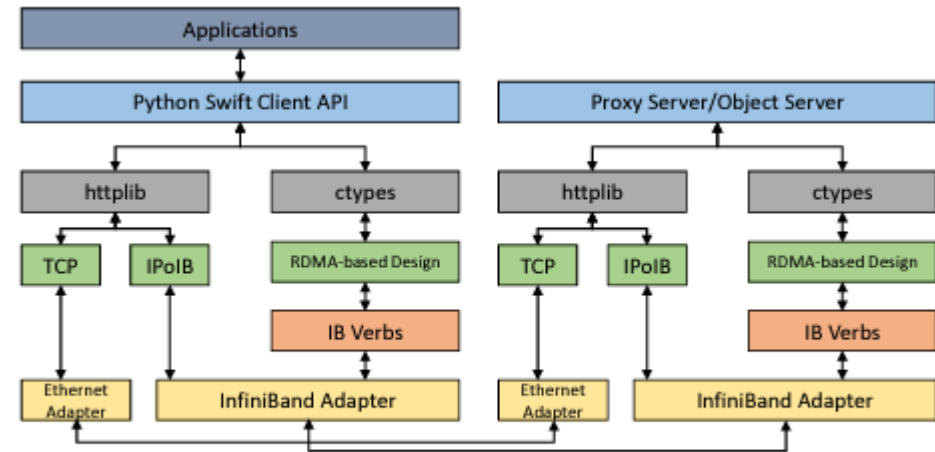


Figure 4. Technology Overview

# Where The Offload Works – AI (AliBaba)

- Alibaba HPN: A Data Center Network for Large Language Model Training

  - https://ennanzhai.github.io/pub/sigcomm24-hpn.pdf

  - We equip each host with 9 NICs each with 2×200Gbps

  - One is connected to the frontend network

  - Remaining eight NICs connect to the backend network to carry traffic during the LLM training.

  - Each of these eight NICs serves for a dedicated GPU (named rail), and thus each GPU has a dedicated 400Gbps of RDMA network throughput, resulting in a total bandwidth of 3.2Tbps.
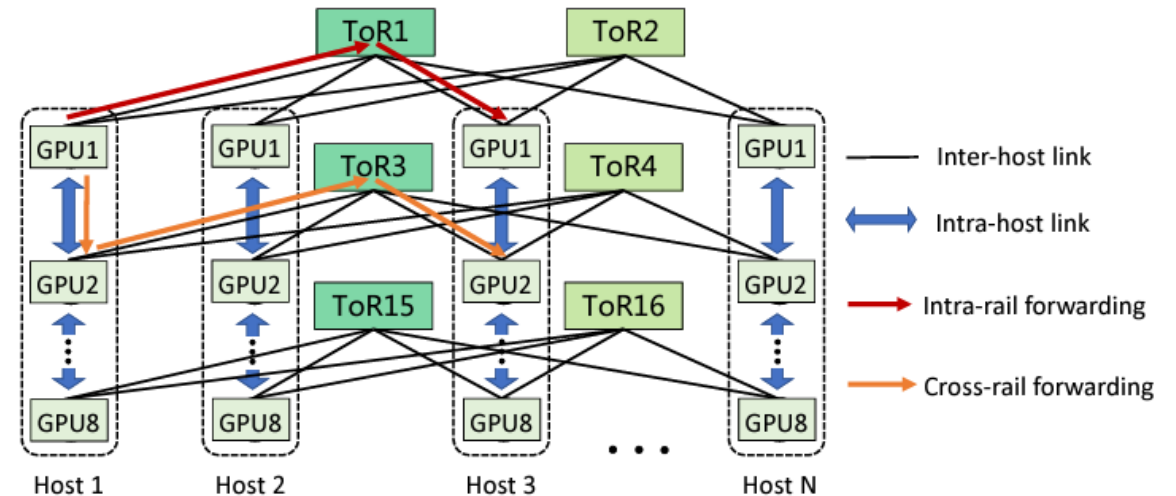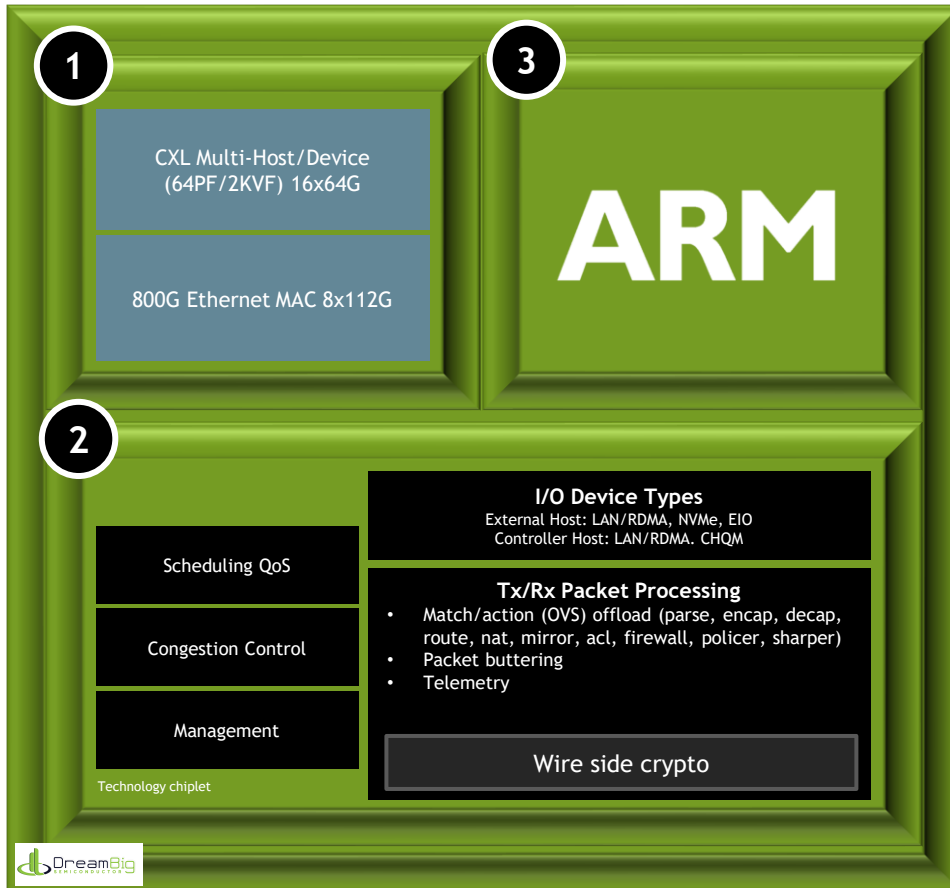


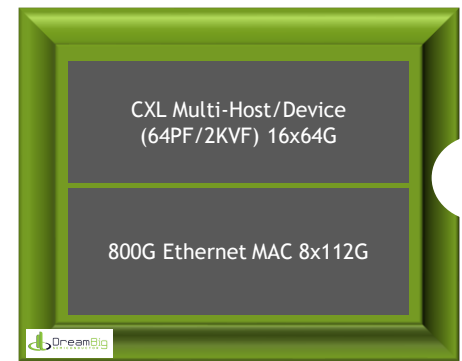Figure 11: Rail-optimized network under dual-ToR.

# Types of SmartNICs

- Vendors have different ideas when they implement SmartNIC:

    - FPGA based:
        - ✅ Fully programable
        - ❌ Restricted functionality, higher power, higher cost, difficult to program

    - CPU based: (Loop up APU based smartNIC. OCTEON is right example?)
        - ✅ Fully programable, easy to program
        - ❌ Less performance, higher cost

    - Dedicated ASICs:
        - ✅ Highest performance
        - ❌ Lower power, restricted programmability

    - Hybrid (ASIC/CPU):

        - ✅ Best of both worlds (DB SmartNIC comes under this category!)

# DreamBig pioneering a game changing Chiplet approach



**1** CXL Multi-Host/Device (64PF/2KVF) 16x64G

800G Ethernet MAC 8x112G

**3** ARM

**2** Technology chiplet

| Scheduling QoS |
| Congestion Control |
| Management |

**I/O Device Types**
External Host: LAN/RDMA, NVMe, EIO
Controller Host: LAN/RDMA. CHQM

**Tx/Rx Packet Processing**
- Match/action (OVS) offload (parse, encap, decap, route, nat, mirror, acl, firewall, policer, sharper)
- Packet buttering
- Telemetry

Wire side crypto

Dividing the SmartNIC chip into 3 chiplets

**1** DEIMOS (IO Complex)

CXL Multi-Host/Device (64PF/2KVF) 16x64G

800G Ethernet MAC 8x112G

**3** PHOBOS(Processor Complex)

ARM RISC-V®

**2** ARES (Networking Complex)

Technology chiplet

| Scheduling QoS |
| Congestion Control |
| Management |

**I/O Device Types**
External Host: LAN/RDMA, NVMe, EIO
Controller Host: LAN/RDMA. CHQM

**Tx/Rx Packet Processing**
- Match/action (OVS) offload (parse, encap, decap, route, nat, mirror, acl, firewall, policer, sharper)
- Packet buttering
- Telemetry

Wire side crypto
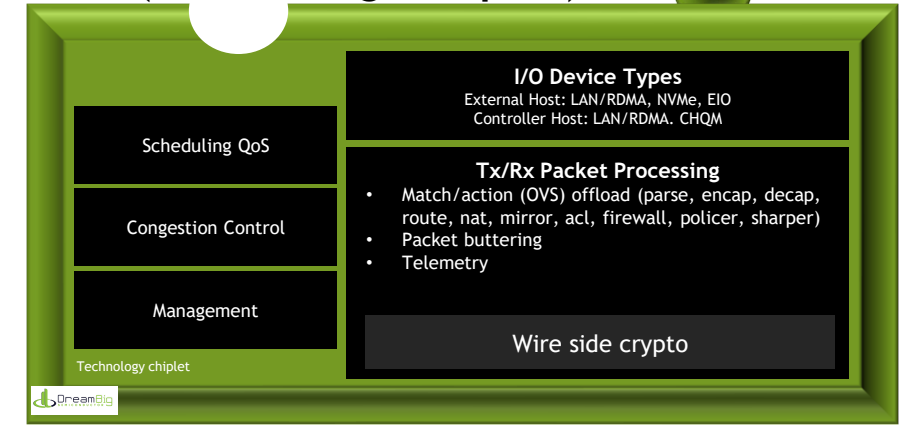
# Physical Design

- Physical design is the most resource intensive portfolio. It requires state-of-the-art technology and software to get the job done and that is why a lot of capital, human and machine resources are spent on physical design. That is why DreamBig has spent a lot of time and capital to develop world-class in-house capability to make sure the tape-out is as innovative as the design itself.

- **In-House Complete RTL-to-GDS Capability:**
  - Utilizing most advanced process nodes and EDA tools
  - In-house specialized techniques to overcome challenges in advance nodes
  - Expertise from around the world:
    - Constraint design
    - DFT Insertion
    - Topographical synthesis
    - Placement, CTS and routing
    - Signoff

# DreamBig SmartNIC Features

Connectivity:

- PCIe 5.0/CXL 2.0
- 25/50/100/200/400/800 GbE network ports

Performance:

- 800 Gbps packet throughput

Virtualization:

- SR-IOV with PFs and VFs

Offloads:

- Programable packet parser
- Programable hierarchal schedular
- Checksum, LSO (with/without tunneling) offloads
- RSS multi-queue packet receive logic
- SDN acceleration with Match/Action offload
- IPsec tunnel and transport offload with AES-GCM
- RDMA over Converged Ethernet (RoCE v2) with RC and UD

# Join Our World Class Team

- ## Hardware Development Areas:
  - High Performance, Low power ASIC Design
  - SoC - Integration of Cutting Edge IPs (PCIe 5.0/6.0, CXL 2.0/3.0, 800G Ethernet)
  - Micro Architecture and Logic Design
  - RTL Design using Verilog and SystemVerilog
  - HW Verification - UVM, Formal Verification
  - FPGA Prototyping
  - Design tools for Simulation, Synthesis, Timing Analysis and RTL Checking (Lint, CDC/RDC, LEC)
  - Silicon Validation and Board Design
  - Backend physical design team (4nm)
- ## Software Development Areas:
  - Linux kernel programing
  - Device drivers (specifically network drivers)
  - Networking stacks (L2/L3/L4)
  - Switching and routing (vSwitch and OVS)
  - RDMA (Remote Direct Memory Access)
  - Storage and NVMe
  - DPDK (Data Plane Development Kit)
  - Firmware Development
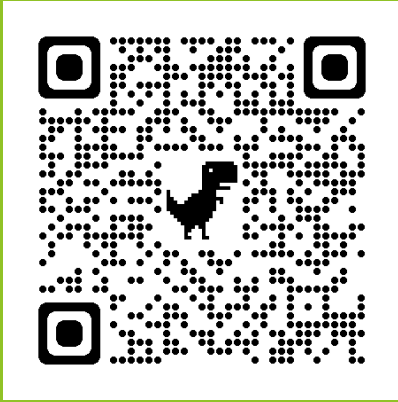
**Official Website**



**LinkedIn page**

For more information write to
info@dreambigsemi.com
hr@dreambigsemi.com

# Q & A