2022 WHITE PAPER

How Adaptable SmartNICs Will Drive Next Generation Data Centers

WHITE PAPER

How Adaptable SmartNICs Will Drive Next Generation Data Centers

TRENDS IN DATA CENTER TECHNOLOGY

Over the past few years, the pace of data center architecture change has been exciting, and we continue to move quickly towards a new paradigm based on heterogeneous hardware and the convergence of acceleration technologies for computation, networking and storage.

Some of the technology drivers for this shift have been building for a while and critical pieces of technology are falling into place, unlocking IO bottlenecks and bringing forward the era of Terabit computing for endpoint processing devices (servers).

In particular, the move of server class CPUs towards a Chiplet-based architecture has allowed processor vendors to restore the innovation cycle of individual component technologies. For example, the dominant IO bus in deployment is based on the PCI-SIG PCIe Gen 3 standard, which has been available in servers since 2012. The roll-out of new generations since PCIe Gen 3 was stalled by the cadence of monolithic CPU silicon development, but the drive toward Chiplet-based CPUs has generated a more rapid development cycle for IO. Industry expectations are that server adoption will jump to PCIe Gen 5 and then see a normal (approximate 3 year) cycle to the adoption of PCIe Gen 6[1].

To put into context, an IO card using a 16-lane PCIe Gen 3 interface will be bottlenecked at around 100Gb/s of bidirectional throughput, whereas a similar Gen 5 interface will deliver around 500Gb/s followed by 1000Tb/s for Gen 6.

Data centers themselves over the past decade have adopted leaf-spine scale out architectures and are hungry to exploit this bandwidth at the server if doing so generates performance or efficiency gains. However, data center architects also need to balance processing efficiency with programmer productivity and ease of deployment. All need to be in place to deliver application velocity at hyper-scale[2]. These requirements are often in conflict, and at each technology node the question as to how much bandwidth ought to be delivered to most efficiently handle the data center's workload must be revisited.

[1] PCIe Gen 6 has adopted PAM4 signalling and FEC which are well tried and tested and together enable a doubling of performance over PCIe Gen 5.
[2] The constant development of features and on-line upgrade to application components .

General purpose CPUs which over the last decade have scaled performance through the adoption of increasing numbers of cores are not suited to handling very high network bandwidths or high packet rates. Simply directing a network firehose at the general purpose CPU caches and leaving it to horizontally layered operating systems and threaded applications to deal with would simply result in most of the data either backed up in large buffers or dropped. Instead, server architecture is changing and the NIC (Network Interface Card) component has itself evolved to become a SmartNIC which is optimized to perform the first tier of application processing at wire speed, right at the network ingress/egress point.

In essence, the SmartNIC should be considered as the evolution of the NIC from a role as peripheral device to become the IO Hub at the heart of the server[3].

THE XILINX SMARTNIC VISION

The Xilinx architectural philosophy for our SmartNIC products is to enable control-plane processing to be distilled out from the data plane, resulting in the composition of an application specific wire-speed processing pipeline as shown in Figure 1.

This fundamentally is a new vertically-integrated approach with the benefits of optimal per-bit processing efficiency and the removal of redundant data movement.

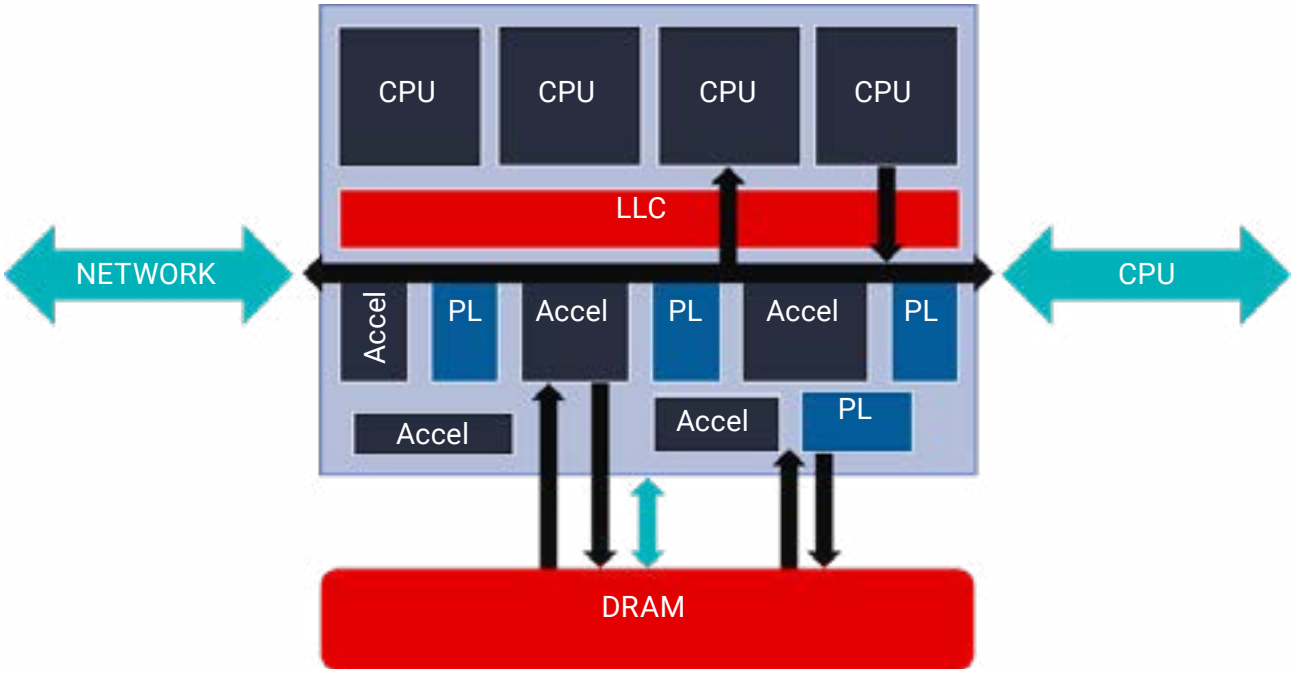

Figure 1: Composable Data Plane

[3] The SmartNIC may also be called a Data Processing Unit (DPU). It is very similar in role to a Network Processing Unit (NPU), with the (somewhat artificial) distinction that the SmartNIC is optimised towards endpoint computing and accelerating the applications of the server rather to perform computing operations within the network.

The Xilinx architecture enables:

> A high-speed application-specific data plane to be dynamically composed from a mix of accelerators and programmable logic.

> The programmable logic used to express the application specific processing that allows data from one accelerator to be chained directly into another, effectively handling the discontinuities introduced by layered protocol and applications structure which would otherwise require CPU (software) intervention.

> High-level language support to express the application specific elements of the composable pipeline. By directly supporting languages such as P4 the Xilinx composable NIC pipelines may be quickly constructed and customised to keep pace with high velocity application development.

> Our approach allows many layers of protocol and application processing to be combined into one streaming data plane with greatly reduced CPU interaction. In so doing we significantly reduce the control-flow problem and memory bandwidth requirements suffered by existing DPU and SmartNIC approaches.

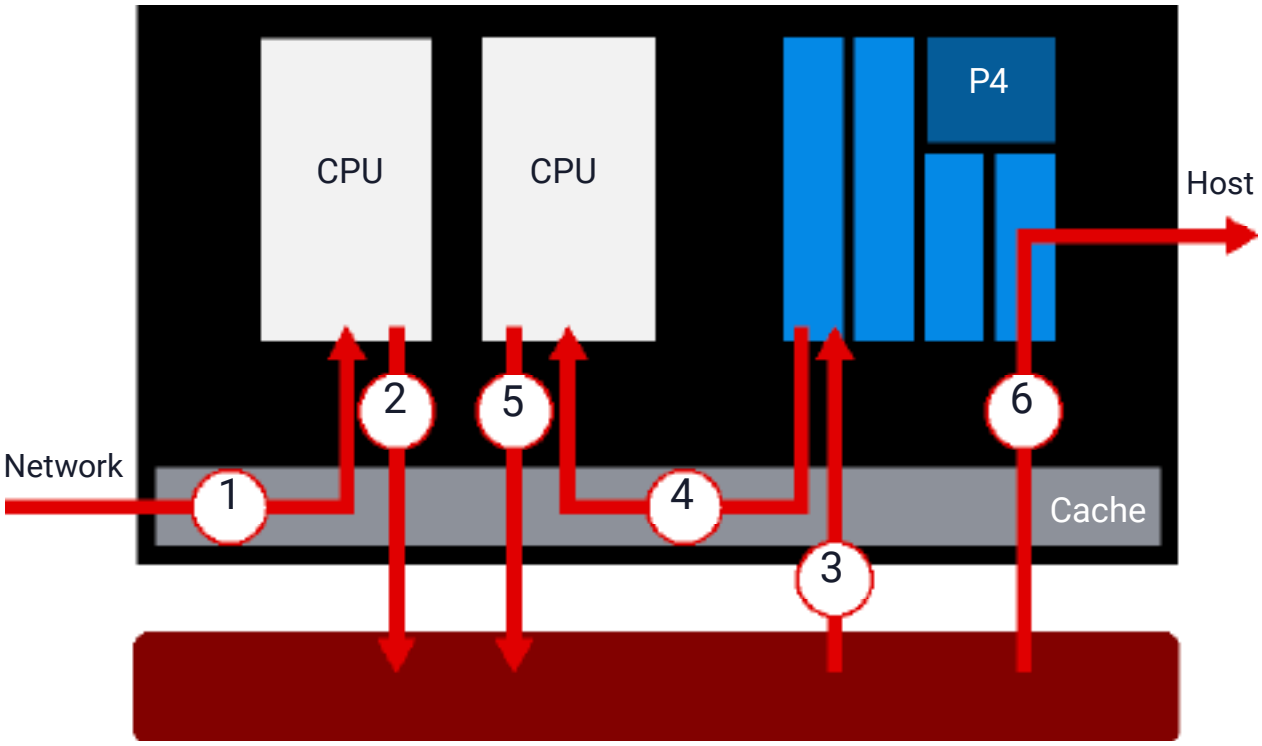

Figure 2: CPU Centric SmartNIC Inefficiencies

Figure 2 shows how a CPU-centric SmartNIC approach causes a control path to be interleaved between hardware accelerators which ultimately causes performance bottlenecks. Adding programmable accelerators can solve part of the problem but leads to a non-uniform programming model for the device and is very hard to scale. Devices based on fixed ASIC programmable accelerators are notoriously difficult to scale to the right number, type or topology and simply can't keep up with the required software development velocity.

In contrast, Xilinx believes that hardware should not impose a fixed architecture on software, but that software should drive the generation and composition of the application specific data plane. We use high-level languages to create these pipelines. Since programmable hardware can be tailored to the software, our composable accelerator pipelines are integrated easily with existing operating systems, libraries and applications without calling for the re-writing of hundreds of man years of software or forcing software onto fixed hardware.

An example SmartNIC configuration is shown in Figure 3. Here a data plane is composed for two applications: (a) Block Storage Virtualization and (b) Computational Storage with Vitis hardware acceleration.

The Block Storage Virtualization application is composed of a pipeline consisting of a sequence of kernels:

> NVMe: standard compliant device interface.

> NVMeOF: services NVMe requests by sending/receiving NVMe Fabric Capsules over TCP network flows.

> TCP: implements TCP data plane operations including segmentation and reassembly.

> Virtual Switch: routes network segments and offloads virtual networking functions such as tunnel encapsulation and de-encapsulation.

> AES: encryption and decryption of network flows.

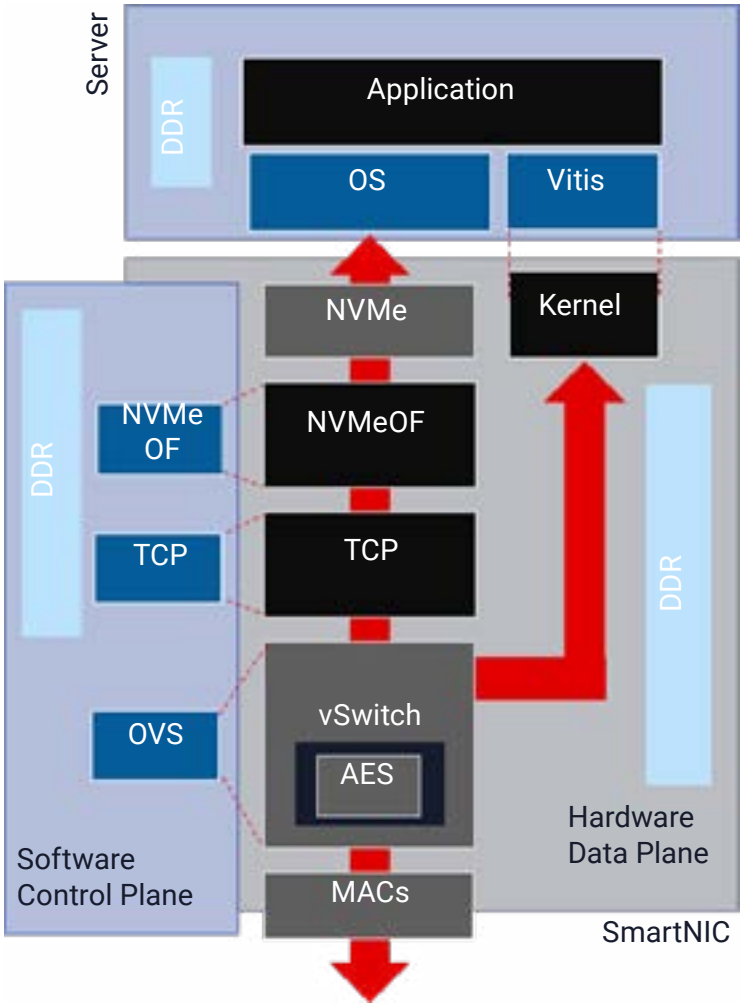

Figure 3 Example SmartNIC Configuration

Data is streamed directly between each of the kernels at wire-speed according to the software control plane. Each kernel is implemented in the most appropriate language (P4, HLS, RTL) and may be dynamically loaded and unloaded while traffic is passing through the SmartNIC. Each kernel can be considered as the accelerated component of an individual software application. These control plane applications themselves run on the embedded CPUs within the (usually Linux) operating system environment and care has been taken by Xilinx to ensure that hardware acceleration is abstracted within standard software subsystems such as OVS, SPDK, DPDK and OpenOnload, and through support of standard offload interfaces such as tc-flower and XDP.

The second application, computational storage, illustrates that an application running on the server may itself be hardware accelerated and that its hardware accelerated component may be a part of a composable pipeline. In this example, the hardware accelerated application is implementing a computational storage application (such as a key-value interface to storage). To this application, the composable pipeline is simply presented as a socket, conceptually just as software would use a socket library or operating system interface. Both the NVMe and computational storage applications may co-exist, with each application's data plane logically attached to a port on the vSwitch.

In Figure 3, the gray boxes indicate platform components and the dark blue boxes indicate application specific kernels. In this example, a standard AES core is wrapped using a protocol handling component implemented using a high-level language. This fine-grained interleaving of application specific logic between standard accelerator functions prevents the SmartNIC having to switch to a CPU in order to execute software.

A PORTABLE NIC ARCHITECTURE

The platform components of the Xilinx SmartNIC are themselves composed using the high-level programming language P4. In this way the entire processing pipeline of the NIC itself is software-defined around a portable hardware scaffold.

Figure 4 illustrates the main processing components of the Xilinx SmartNIC and their main function. For example: the VNIC RX handles processing operations required for operating system ingress: checksum offload, flow steering and flow spreading. Whereas the Virtual Switch (vSwitch) provides the network virtualisation processing operations typically performed by a hypervisor such as tunnel encapsulation/de-encapsulation.

Each component is implemented by P4 programs with hardware extensions for acceleration or functions not easily expressed by the language. This approach provides tremendous flexibility and productivity. For example, a new custom network overlay protocol could be quickly designed, expressed in P4 and uploaded into the SmartNIC, just as previously described for an application kernel.

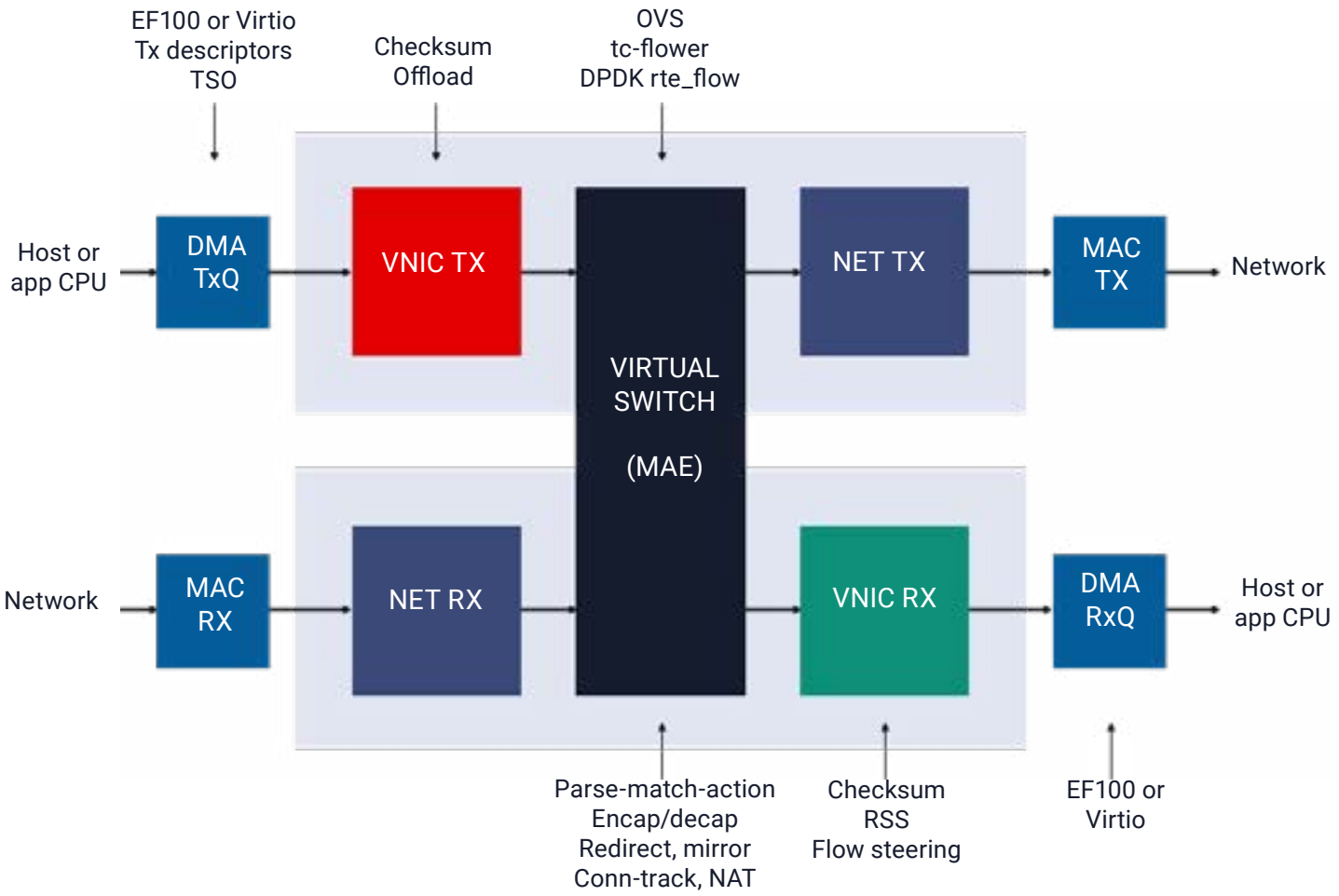

Figure 4: A Portable NIC Architecture

The hardware scaffold includes standard Ethernet Layer2 functions such as Priority Flow Control (PFC), accelerated flow lookups using Xilinx's Algorithmic CAM IP, and the host device interface (DMA).

The host device interface itself is designed to be composable, initially supporting virtio and ef100 device types but in the future extendible to other device types such as RDMA and NVMe.

The host device interface provides symmetric Queue Pair support for drivers running on any CPU (the host CPU or embedded SmartNIC cores). This enables the same operating system environment to be provisioned within the SmartNIC as for the server and enables rapid migration of software to the SmartNIC environment. Support for hardware extensions and new application specific hardware features can be mapped onto a Queue Pair on demand, under firmware control.

PLUGIN INTERFACE

The plugin interface is defined to create composable accelerator pipelines by allowing kernels to be connected to the hardware scaffold. For example, to allow a storage virtualisation pipeline to be connected to the port of the vSwitch.

The plugin interface provides credit-based flow control and scheduling support so that any kernel (or pipeline) can be created of arbitrary topology and inject or terminate network flows.

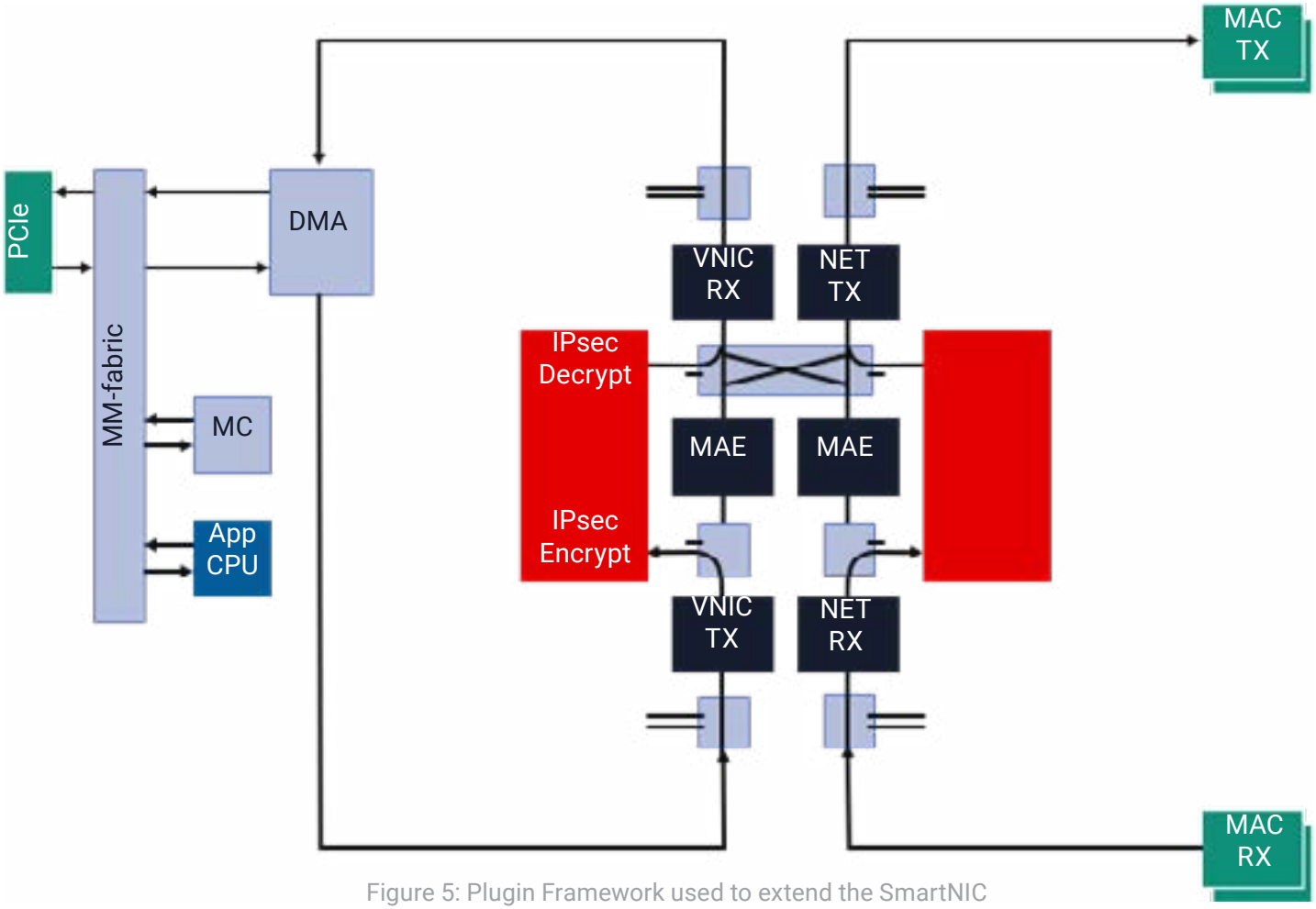

Figure 5: Plugin Framework used to extend the SmartNIC

The same plugin interface is supported at each of the components of the SmartNIC and may be used to extend and augment processing of the SmartNIC itself. For example, to support inline encryption/decryption as shown in Figure 5. Other examples may include network functions such as load-balancers or firewalls.

While the plugin interface is used to allow arbitrary kernels to be loaded and unloaded, it generally is abstracted within the run time support provided by the Xilinx language level tools and is transparent to the high-level programs themselves. It can be considered as the glue which bridges between hardware and software and allows data to stream smoothly through the device.

THE XILINX SN1000

The Alveo SN1000 is Xilinx's first SmartNIC to support the composable data plane and portable NIC architecture. It is presented as a PCI card supporting 100G Ethernet networking assisted by 8 or 16 A72 ARM cores. The SN1000 supports hardware accelerated virtual networking interfaces: virtio and ef100 with vDPA, DPDK, SPDK, OpenOnload, and Linux XDP accelerated software support.

The SN1000 is a complete SmartNIC providing the latest networking offloads including embedded OVS offload and accelerated virtio-blk storage virtualization (to Ceph/TCP) as part of the platform.

Figure 6: The Xilinx Alveo SN1000 SmartNIC

In summary, the old notion of a server containing network interfaces and application accelerators both as peripheral devices is dead and buried and the future lies in converged adaptive platforms. The launch of the SN1000 is the first of a multigenerational strategy of these products, delivering Terabit processing and driving next generation data centers.

Steve Pope, PhD
Xilinx Fellow

Steve Pope is a Xilinx Fellow and is responsible for the direction and leadership of the Xilinx data center networking architecture team. Steve founded Level5 Networks, staying with the company as CTO through its merger with Solarflare Communications, which was acquired by Xilinx in 2019. Steve received his PhD in Networks and Operating Systems from the University of Cambridge.

TAKE THE NEXT STEP > Visit www.xilinx.com/smartnic