AMD PENSANDO

WHITE PAPER | CASE STUDY

AMD PENSANDO™ DPUS OFFER HIGH PERFORMANCE WHILE HELPING LOWER TCO FOR CLOUD SERVICE PROVIDERS

February 2024

Introduction

This case study highlights the rapid adoption of the AMD Pensando DPU and Software-in-Silicon Development Kit (SSDK) by a cloud service provider (CSP), showcasing the advantages of employing a programmable software-defined data plane.

The modern cloud data center has converged on 100 Gb/s to the server, and the need for higher 200/400/800 Gb/s bandwidth driven by storage access for ML applications is present. As performance demands continue to increase, cloud infrastructures and associated network, storage, and security services must keep pace. CSPs offer a wide range of *stateful* services, (services that maintain state information about individual clients or sessions). Two common examples of stateful services are server load balancers (SLBs), and connection tracking for security.

Stateful services can be challenging to scale and manage because they require network devices to maintain per-session state. This can lead to performance bottlenecks and scalability challenges, especially in large and complex networks. Legacy data center architectures struggle to keep up with the packets per second, sessions, and connections per second (cps) needs as performance demands continue to increase. In addition to increasing throughput demands, security concerns require that all sessions in a multi-tenant environment be encrypted, which further burdens legacy architectures that rely primarily on CPUs and fixed-function ASICs.

Current Customer Environment

This CSP's network is one of its most important infrastructure components, responsible for delivering high-performance, reliable, and secure networking services to its customers. They have designed it on a foundation of three key principles:

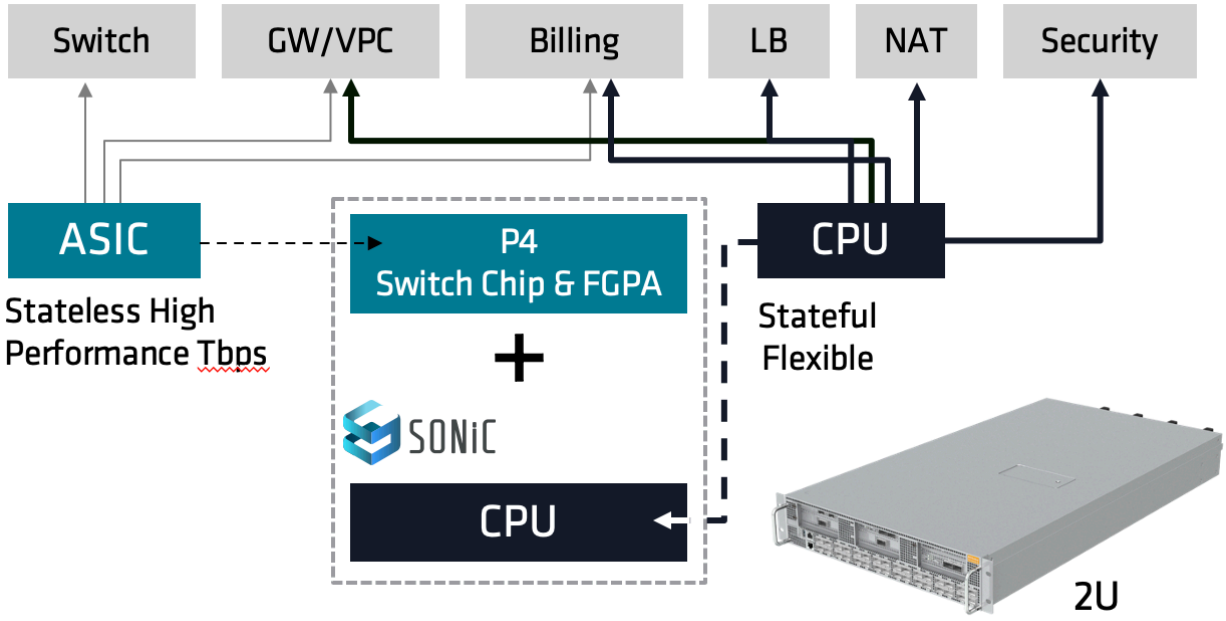
- **Scalability**: The network must scale to meet the demands of its growing customer base.
- **Reliability**: The network must be reliable and deliver high-performance services even during peak demand.
- **Security**: The network must be secure and protect customer data from unauthorized access.

Their architectural philosophy is based on the following characteristics:

- Open and programmable;
- Predictable performance to support multi-tenant and built-for-AI infrastructure; and,
- High-performance network functions (VPC virtual network gateways, server load-balancing, NAT, and security)

Looking at the last of these: the CSP started exploring hardware offloading to build network functions and gateways three-plus years ago. Their 1st generation hardware offloading platform, ("Gen 1", Figure 1), was built with a P4-based switch processor, multiple FPGA cards, and general purpose CPUs, in a single chassis.

Figure 1. The Gen 1 solution



Each component plays a unique role: The P4 switch processor offloads bulk traffic ("elephant flows") and consolidates switch functions into the Gen 1 appliance. The general-purpose x86 CPU runs the CSP's network operating system and provides virtualized network services such as SLB, NAT, or DDoS.

CSP networking deployments require handling millions of microflows, which cannot be handled by general-purpose CPUs or P4 switch processors alone: general-purpose CPUs are expensive and have too much latency and jitter for microflows, while the P4 switch processor has limited table scale. Instead, the CSP uses FPGA cards in the appliance to overcome these limitations.

However, FPGAs cannot offload or accelerate all the intelligence and packet processing required for CSP networking. As a result, their Gen 1 appliances still rely heavily on general-purpose CPUs for tasks such as metering, load balancing, network address translation, and security. This increased acquisition and operational expenses, resulting in an inefficient allocation of resources for networking purposes. Additionally, FPGAs do not provide IPsec offload, a growing demand in CSP networking.

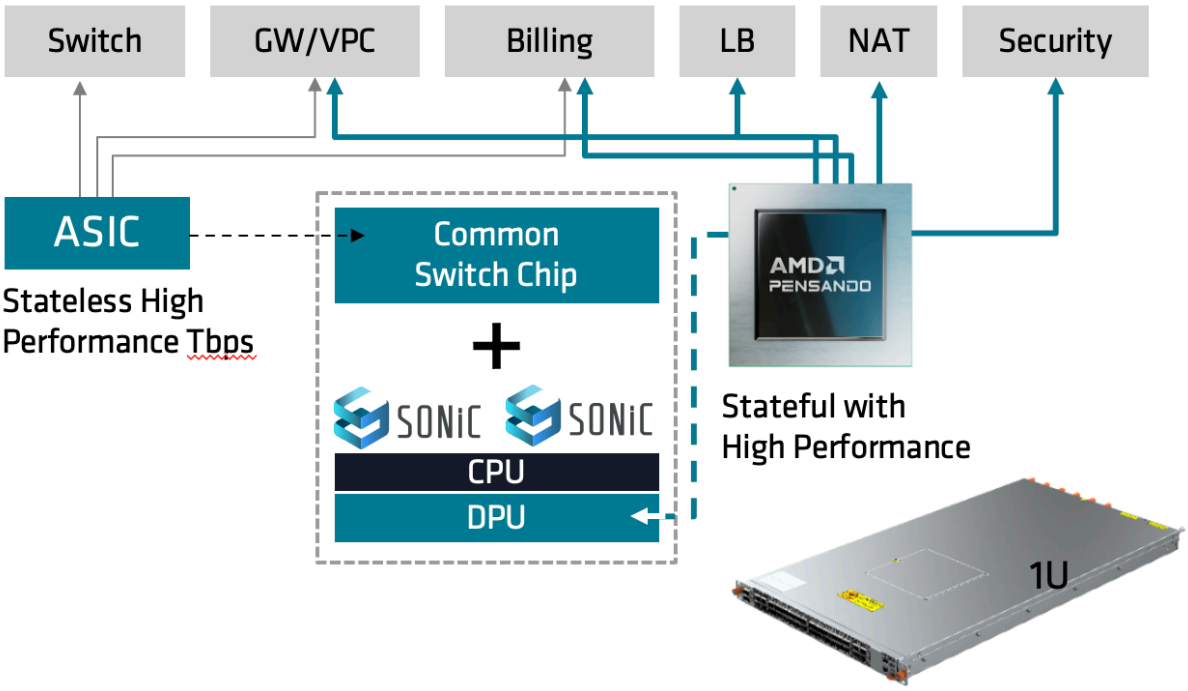
Even with the high cost of using multiple components and power requirements, the Gen 1 implementation allowed the CSP to replace traditional server-based VPC virtual network gateways and support additional network and security functions (security, billing, metering, load-balance, and NAT) with some hardware offloading. Gen 1 was well-suited to edge computing scenarios, where there is great cost sensitivity and physical space is highly restricted. Gen 1 is also used as a VPC gateway in their central locations, which need to provide terabits per second throughput for the stateless gateway functions. Without Gen 1, the CSP would need to deploy dozens or even hundreds of traditional server-based VPC gateways in central locations for their big customers.

Gen 1 was very successful for software-based network and security functions, and is a flexible solution, but it has one big limitation: it has very limited bandwidth per CPU core, which for several of their larger customers caused issues related to elephant flows. High component cost was also a limiting factor.

The CSP started analyzing and evaluating emerging xPU hardware technologies for their suitability to offload, accelerate, and forward stateful network. Several xPU vendors were evaluated to be coupled with a Broadcom Trident4 processor to provide the next-generation (“Gen 2”) platform.

The evaluation resulted in the selection of AMD Pensando DPUs for Gen 2, providing 800 Gb/s bandwidth for stateful network functions with one hundred million sessions per Gen 2 appliance. The Gen2 platform is an evolution to *Smart Switches* (Figure 2). Like Gen 1, Gen 2 runs SONiC on the switch ASIC and DPUs. The CSP has experienced increased bandwidth and has seen a dramatic reduction in network latency. They have also experienced a power reduction of approximately 5X savings, and have seen that each DPU can achieve 400 Gb/s and 70 million packets per second forwarding bandwidth, while migrating network and security functions to the DPU has reduced the bandwidth limitations per CPU core they experienced with Gen 1.

Figure 2. The Gen 2 solution



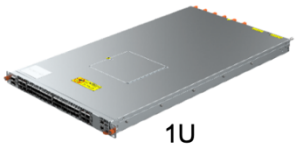
The performance gains are evident, but also, the ease and flexibility to migrate services from Gen 1 to Gen 2 was compelling. AMD Pensando DPUs use the open standard P4 language, and provide a powerful and flexible P4-based software development environment, including libraries and APIs to simplify use of the P4 engine.

The AMD Pensando Software-in-Silicon Development Kit (SSDK) includes ready-to-deploy reference pipelines; the CSP was able to leverage the feature-rich and flexible SSDK, specifically its SDN policy-offload reference policy, and integrate it with their existing P4 code. This made it possible for them to migrate and test load-balancing and NAT in only two weeks.

Figure 3 highlights the power, latency, cost savings (52 general-purpose CPU cores replaced by two 2nd-generation “Elba” DPUs), and performance and flexibility gains by migrating to a programmable Smart Switch architecture.

Gen 2 can fit into all Gen 1 (1 RU vs 2 RU) deployment scenarios but now also can be used as a ToR switch in the data center, further expanding its use cases and helping improve TCO. As bandwidth demands increase and new network and security services are requested, only a domain-specific architecture can provide the horsepower and flexibility they require.

Figure 3: Results experienced by CSP



Stateful LB/NAT/etc.	Gen 1 with CPU (52 Intel cores)	Gen 2 with DPU (2 AMD Pensando DPUs)
Bandwidth	200 G/s	800 G/s
Sessions	100 M	100 M
Latency	~100 µs	~2 µs
PPS	~70 Mpps	~140 Mpps
Elephant Flow	10 G/~1 Mpps per core	400 G/70 Mpps per DPU
Power Draw	5X	80% reduction

Why the AMD Pensando P4 Programmable DPU

Although flexible and agile, existing cloud architectures using general-purpose CPU and Multicore SoC do not have the needed performance (throughput, PPS, CPS, and sessions) for network, security, and storage functions. No merchant ASIC had the programmability or hardware accelerators required, especially in storage and security. Although FPGAs can have high performance and are programmable, the AMD Pensando DPU provides a programming model specifically designed for networking solutions, and simplifies programming tasks with a network-specific paradigm. This not only simplifies and speeds programming, it also is implemented with low power. The AMD Pensando DPU implements a domain-specific architecture on its custom silicon—a built-to-purpose P4-programmable processor that fits the CSP's requirements for flexibility, agility, and performance. It is designed to tackle a broad range of tasks, including storage, telemetry, statistics, SDN, security, congestion management protocols, and RDMA. The DPU P4 pipeline can handle a mixture of these tasks at once without compromising performance and throughput—the "power of *and"* in AMD Pensando.

The AMD DPU provides a programmable high-performance pipeline consisting of Rx and Tx policers, full Rx and Tx statistics (including all drop reasons), VxLAN tunnel encapsulation and decapsulation, IPv4/v6 routing, stateless and stateful security rules, network address translation (NAT), server load balancing, encryption services, VLAN to VPC mapping, and VPC peering (i.e., tunneling), IPsec, and storage offloads.

Benefits of Using a Software-Defined Data, Control, and Management Plane

- **Flexibility**: Software-defined data/control/management planes can be easily programmed to support new functions and services with performance and scale. This makes a DPU ideal for data centers, which need to be able to adapt to change.
- **Agility**: Software-defined data planes can be programmed quickly and easily, without having to go through a lengthy redesign and manufacturing process. This makes it possible to quickly deploy new features and services.
- **Cost savings**: Software-defined data planes can help to reduce costs by eliminating the need to respin hardware when new functions are needed.

How the SSDK Helps Accelerate Migration and Development

The AMD Pensando solution is entirely built from a set of P4 programs that interact with the DPU data plane. This approach allows for virtually unlimited flexibility and permits rapid deployment of fixes, enhancements, and major protocol changes via simple software upgrades. P4 makes it possible to define and implement network forwarding pipelines in software. This offers the same speed and flexibility as an FPGA or ASIC, but with the added benefit that P4 code is open source and can be modified by the entire software team. Because of the network-specific design of the P4-programmable match processing units, the AMD Pensando software team can use P4 to implement new features in a matter of days, which would have required months, if not a year plus, for reprogramming at a lower hardware level and years for a new ASIC to be spun. A P4 pipeline isn't rigid and can be reprogrammed to suit evolving networking, storage, and security needs.

The AMD software-provided P4 binaries and libraries can be deployed alongside the CSP's P4 code on the same DPU to integrate their functionality into the customer's system. This approach streamlines the implementation process and enables seamless compatibility.

Combining a programmable data plane, control plane and management plane with the SSDK provides true flexibility for building services. To enable stateful processing, AMD has extended the P4 language with mechanisms for locking tables and other synchronization primitives (required when multiple packets update shared records). AMD extended the architecture with Direct Memory Access (DMA) primitives to move data between the local memories, the host CPU, and hardware accelerators. In addition to the familiar Ingress and Egress pipelines, two new P4 DMA pipelines handle the Match-Action processing for DMA Rx (to host) and Tx (from host) transfers. The P4 DMA pipelines also provide DMA for the x86 or Arm® multicore complex and hardware accelerators.

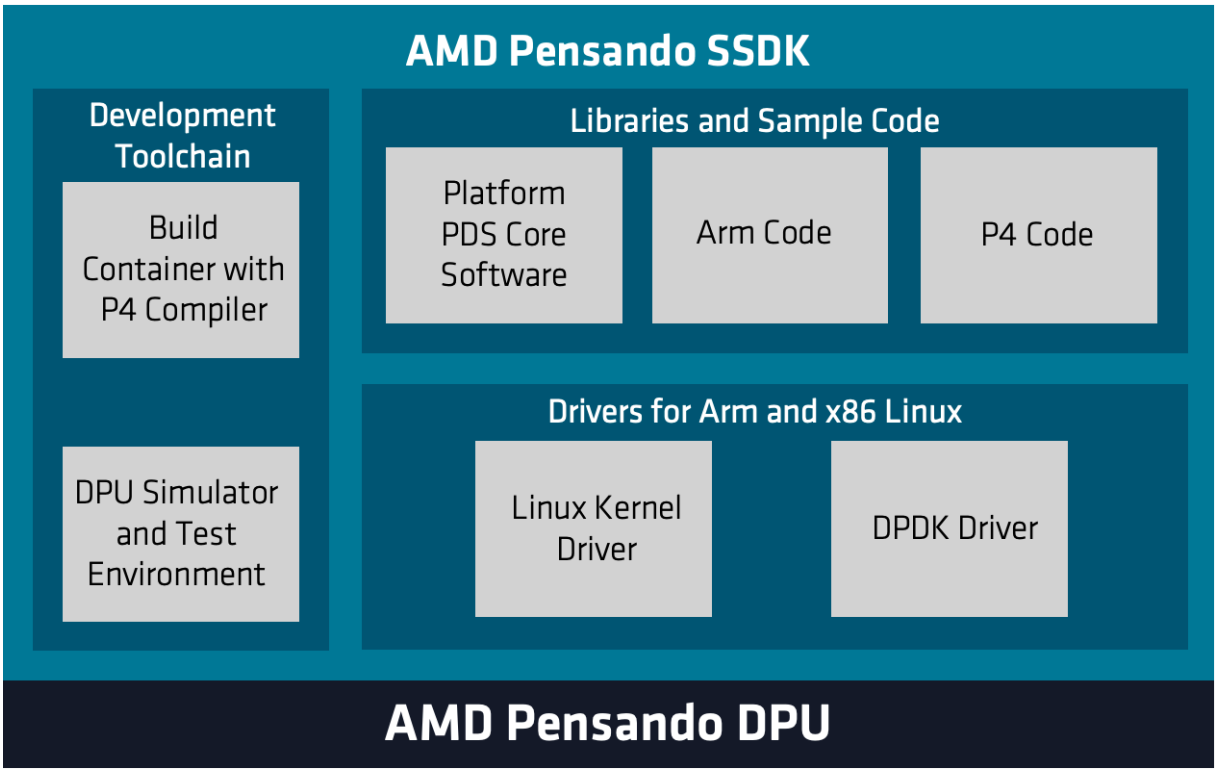
Key features include:

- Simplified setup that quickly installs the development environment
- Extensive examples and reference pipelines to get started with actual P4 development
- Readily available containerized infrastructure to build, test, and debug code, before cross-compiling the same code to DPU
- Compiled and ready to integrate system software supporting secure boot for DPU and applications
- Extensive documentation that helps navigate through the development environment.

The Software-in-Silicon Development Kit (SSDK) consists of (Figure 3):

- A build container includes the Linaro Arm® cross-compile toolchain, the AMD Pensando P4 compiler, and all the code and tools required to create executable images for the DPU.
- The build container doubles as a DPU simulator that executes any software running on a DPU and emulates its interfaces through which traffic can be sent and observed. Consequently, the development and functional testing of software for the DPU can be performed on an x86 platform. Only performance testing requires the actual DPU.
- Several libraries and executables are made available to developers for executing basic initialization and controlling the platform while it is running.
- Drivers are provided to enable the transmission and reception of packets from software running on the host and the Arm processors.
    - Linux® kernel drivers can be used by applications that operate as endpoints for packets. For example, a process implementing BGP running on Arm will use these drivers to send and receive routing updates. The packets sent and received will be processed by the P4 pipeline and sent out of the interfaces.
    - DPDK drivers can be used by Arm or x86 software that injects packets in the P4 pipelines and receives packets from P4 pipelines at high performance. One example is a module receiving the first packet of each flow, applying an access policy, possibly forwarding the packet, and installing an entry in a flow table within the P4 pipelines that will be used to process the following packets of the flow.
- Sample P4 and Arm programs called *reference pipelines*. These can be used to familiarize with the SSDK, as examples of how to write software for the AMD Pensando DPU, or as a starting point for code developed using the SSDK.

Figure 4. SSDK components



AMD Pensando Reference Pipelines

The customer accelerated their Gen 2 development efforts by using the SSDK-provided reference pipelines alongside their code. As mentioned earlier, the CSP uses SONiC. To further fast-track the development of Gen 2, the CSP customer used the SSDK SAI (Switch Abstraction Interface) reference pipeline, which runs the SONiC OS on the AMD Pensando DPU. It enables integrating an SSDK-based pipeline into a SONiC environment running on a DPU to accelerate customer development and deployment with SONiC-based switches. With this model, all the SONiC-provided services, like the routing stack, management interface, monitoring, etc., can be leveraged while also taking advantage of all the DPU capabilities via the SSDK. The reference app provides all necessary software components to integrate the base DPU hardware into SONiC and underlay functionality.

In addition, the SDN Policy Offload reference pipeline was utilized as a ready-to-deploy P4 library to build a high-performance, highly scalable data path application. The P4 libraries included the LPM routing table and ACL security lookup in P4 DMA, with the CSP reporting much higher performance than their legacy approach, where the routing table lookup and ACL

lookup occurred on general purpose cores. By storing the routing tables and ACL table on DDR memory, the solution can scale to millions or tens of millions of entries, without the high cost of HBM or TCAM used in ASIC designs. Unlike legacy designs, connection tracking and flow aging are handled in the P4 pipeline, further offloading the CPU. With the routing table lookup and ACL lookup offloaded and accelerated by P4 DMA and connection tracking offloaded by P4, the P4 library allows users to build a stateful services application with extremely high CPS, or stateless services applications (not flow-aware, or connection session aware) such as a cost-effective router with large scale routing table and ACL.

Key SDN Policy Offload Reference Pipeline Features Used:

- Ready-to-deploy libraries for IPv4/v6 routing, security policy evaluation, and hardware-assisted flow aging.
- P4 pipeline programmability for flexible software-defined constructs.
- Hardware accelerated flow offload, VLAN and VXLAN encap/decap, Route Guard, NAT, load balancing, and metering.

Conclusion

The AMD Pensando DPU, P4 SSDK, and reference pipelines are powerful technologies and tools for cloud services providers as they develop and deploy high-performance, highly scalable data path applications. In the case illustrated here, it provides a flexible and agile platform for building innovative new networking, storage, and security solutions and is the ideal programmable platform for their second generation and future evolutions.

End Notes

All performance and cost savings claims are provided by the third-party customer and have not been independently verified by AMD. Performance and cost benefits are impacted by a variety of variables. Results herein are specific to the third party and may not be typical. GD-181

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

amd.com/pensando

PWP24001