NETRONOME

WHITE PAPER

Open vSwitch Offload and Acceleration with Agilio® CX SmartNICs

DRIVING MASSIVE IMPROVEMENT IN DATA CENTER EFFICIENCY

THE CONCEPT OF SERVER-BASED VIRTUAL SWITCHING UNDER SOFTWARE-DEFINED NETWORKING (SDN) CONTROL ... HAS REVOLUTIONIZED THE DATA CENTER BY PROVIDING AN AGILE AND POWERFUL METHOD TO CONTROL OVERLAY TOPOLOGIES, MICRO-SEGMENTATION OF SECURITY POLICIES, AND PROVIDE ENHANCED NETWORK VISIBILITY

CONTENTS

INTRODUCTION

As virtual computing has grown, virtual switching has become an integral part of modern cloud networking architectures, providing necessary access for virtual machines (VMs) by linking virtual and physical network interfaces together. More recently, the concept of server-based virtual switching under software-defined networking (SDN) control, also known as server-based networking, has revolutionized the data center by providing an agile and powerful method to control overlay topologies, micro-segmentation of security policies, and provide enhanced network visibility. A widely deployed example of an SDN controlled virtual switch for server-based networking is the Open virtual Switch (OVS).

The benefits of OVS for server-based networking deployments have been well established, such as software-defined flexibility and control of datapath functions, fast feature rollouts, and the benefits of open source leverage. Cost-reduction through the use of a standard IT server infrastructure to implement networking functions is another goal of server-based networking which is problematic to achieve. This is because compute servers struggle when it comes to packet throughput for datapath functions and require a large number of expensive CPU cores, taking away a significant amount of server resources that could otherwise be used to run cloud applications.

A solution to this problem is to offload the OVS processing to a SmartNIC that has the ability to implement the OVS datapath directly on the adapter. This paper will show how OVS packet throughput can be significantly increased, while at the same time reclaiming massive CPU resources that would have otherwise been consumed. We will present a comparison of software-based implementations against an offloaded solution with the Netronome SmartNIC, showing detailed performance metrics and benchmarks, in addition to showing how such offload can be achieved without sacrificing software flexibility and feature velocity.

EXISTING OVS IMPLEMENTATIONS

Following its initial release as an open source project in 2009, OVS has become the most ubiquitous virtual switch (vSwitch) in OpenStack deployments. The standard OVS architecture consists of user space and kernel space components. The switch daemon runs in user space and controls the switch, while the kernel module implements the OVS packet data fast path. Over time, the user space and kernel components have evolved to introduce more sophisticated and scalable tunneling support by enabling network virtualization with overlays such as Virtual Extensible Local Area Networking (VXLAN). VXLAN is a network virtualization related specification being developed in the IETF.

Another variation that has become popular is to implement the OVS packet data fast path in user space. With this approach, some advantage can be gained by using the DPDK poll-mode driver to deliver packets directly into user space, bypassing the Linux kernel altogether. This eliminates some context-switching overhead and can enable additional optimizations for the vSwitch, such as loading packets directly into cache, and batch processing. Customizations can also be made but at the expense of losing leverage from standard OVS.

NETRONOME AGILIO® SMARTNICS PROVIDE A FRAMEWORK FOR TRANSPARENT OFFLOAD OF OVS.

AGILIO TRANSPARENT OVS OFFLOAD ARCHITECTURE

Netronome Agilio SmartNICs provide a framework for transparent offload of OVS. With this solution, the OVS software still runs on the server, but the OVS datapath match action tables are synchronized down to the Agilio SmartNIC via hooks in the Linux kernel. By running the OVS data functions on the Agilio SmartNIC the OVS datapath is dramatically accelerated, while leaving higher-level control and features under software control and retaining the leverage and benefits derived from a sizeable open source development community. The Agilio transparent OVS offload architecture is shown in Figure 1.

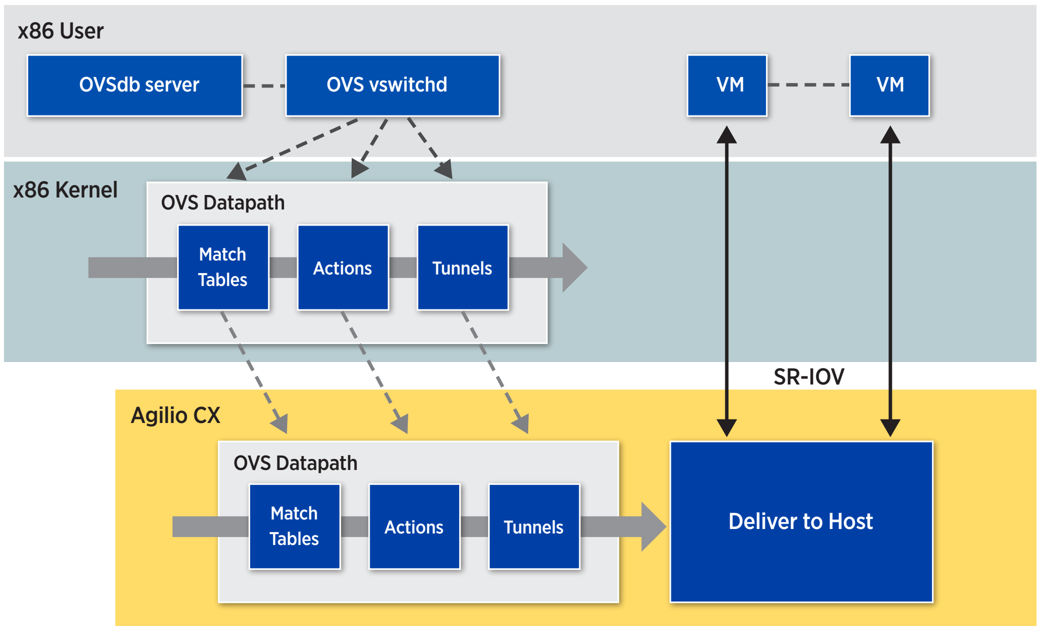

Figure 1. Agilio Transparent OVS Offload Architecture

BENCHMARK METHODOLOGY

The benchmark goals are to provide performance data for Netronome Agilio SmartNICs offloading OVS from the server CPU, and compare it to Kernel based OVS (Kernel OVS) as well as DPDK-based userspace OVS (User OVS). For the latter two approaches, we use a common non-SmartNIC, the Intel XL710, as a reference.

Two examples of datapath match action are used to provide results for a relatively simple L2 forwarding use case, as well as a more complex tunnel endpoint use case with VXLAN as the overlay tunnel encapsulation. For each case the OVS is configured with 1,000 wild card rules. For the L2 case the rules will match on destination MAC address, and determine which VM to forward the packet to. For the VXLAN case, the VXLAN header is parsed, and the virtual network identifier (VNI) and inner MAC address are used for forwarding to the destination VM. An IXIA traffic generator is used to distribute a total of 64,000 exact-match flows across the afore-mentioned target rule sets.

A key goal of the benchmarking is to quantify OVS datapath performance and the ability to deliver data to virtual machines (VMs). To make sure we are measuring OVS datapath performance and not driver or Linux IP stack performance, multiple destination VMs are used in parallel to eliminate any bottlenecks that are not in the OVS datapath. In the case of testing with Linux kernel (netdev) drivers, it was found empirically that using 8 target VMs was sufficient to show full OVS performance for the highest performing case. When using DPDK poll-mode drivers, only 2 target VMs were needed to saturate the datapath in the highest performing case, since the poll-mode drivers can sustain higher throughput.

TEST SETUP

The following hardware and software was used for the device under test:

Server: Dell PowerEdge R730
CPU: Intel Xeon CPU E5-2670 v3 @2.3Ghz with 64GB of DRAM
Operating System: Ubuntu 14.04.3 LTS
Kernel Version: 3.13.0-57-generic
Open vSwitch: v2.3.90
Server Adapters:
Intel XL710 for software (Kernel and User) OVS
Agilio CX 4000-40Q-10 with LNOD2.1 for accelerated OVS
Test Generator: IXIA/Spirent or Netronome Trafgen

The test setups used for Kernel and User software OVS on the XL710 and accelerated OVS on the Agilio CX are shown in the diagrams below:

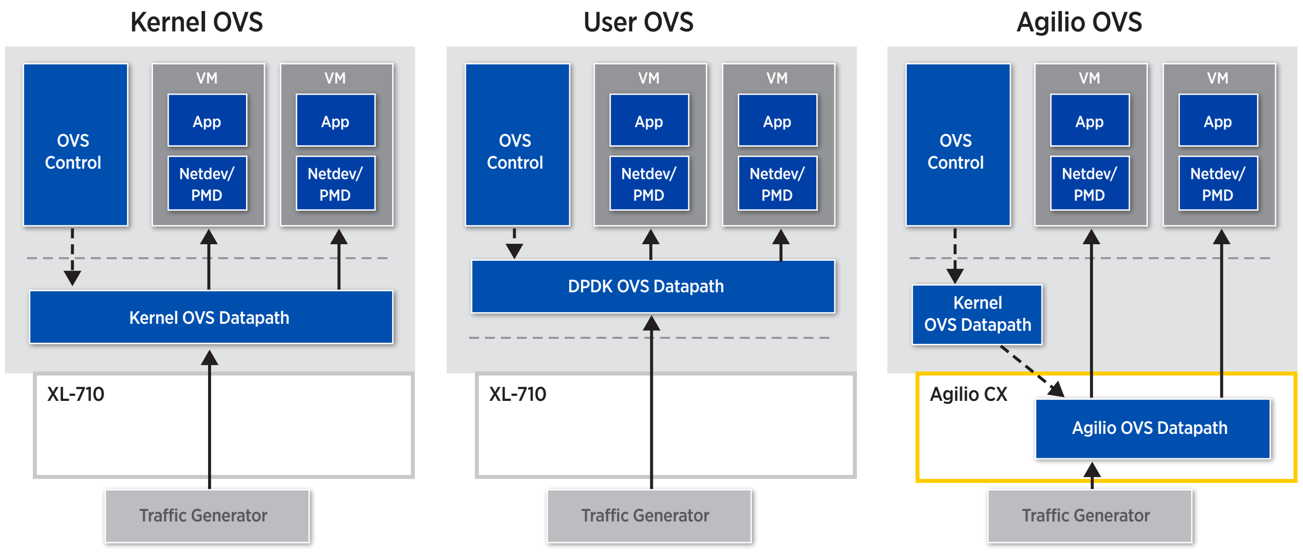

Figure 2. Test Setups for Kernel, User, and Agilio OVS

SERVER CPU CONFIGURATIONS

For an Intel Xeon Dual Socket Server with 12 physical cores per socket, we show test data that supports the following configurations. For software OVS running as Kernel OVS or User OVS, we tested with 12 physical cores dedicated to the OVS function. This configuration was chosen to show the maximum performance achievable in a balanced configuration, where the application throughput per core is equivalent to the OVS throughput per core. The software OVS reference configuration is show Figure 3 below:

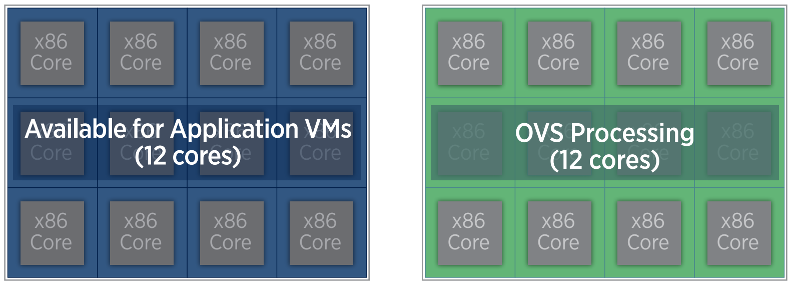
x86 Core
Available for Application VMs (12 cores)
OVS Processing (12 cores)

Figure 3. Dual Socket Kernel and User OVS configuration

For the case of accelerated OVS running on the Agilio CX SmartNIC, we only require 1 physical core dedicated to the OVS function. This frees up 11 additional cores to run more applications. This configuration still allows Agilio to reach maximum performance and deliver the highest data rates to applications via SR-IOV. The Agilio OVS reference configuration is show Figure 4 below:

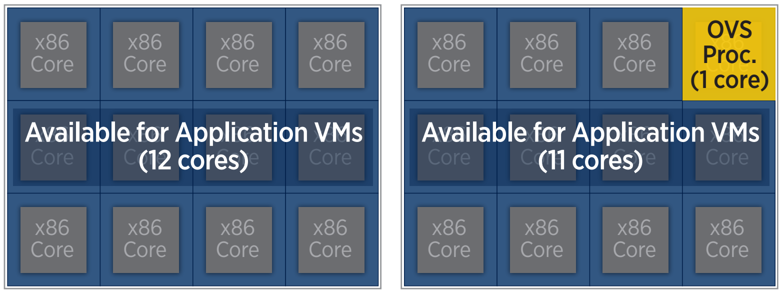
x86 Core
Available for Application VMs (12 cores)
OVS Proc. (1 core)
Available for Application VMs (11 cores)

Figure 4. Agilio OVS configuration

BENCHMARK RESULTS

For each of the test cases, the applied load is 40GbE line rate for packet sizes ranging from 64Byte to 1514Byte packets. Packets are injected from the traffic generator at the network interface and they are dropped and counted by a sink application in the VMs. The following graph displays the datapath performance for OVS with L2 Forwarding and delivery to VMs:

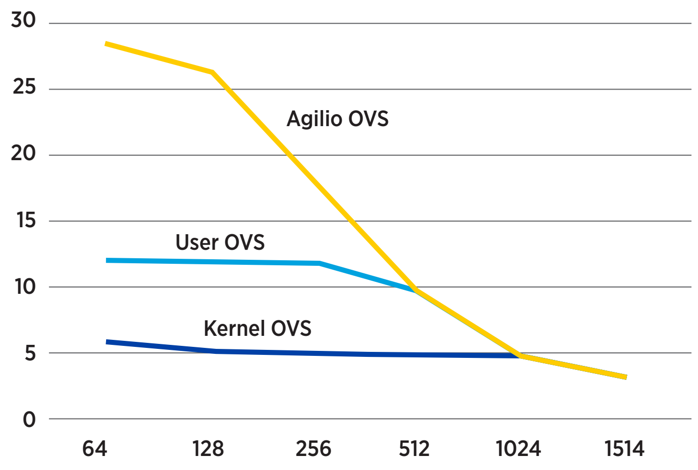

Figure 5. OVS L2 Delivery to VMs (MPPS)

We can see from the results that the standard OVS datapath running in the kernel achieves a maximum throughput of about 5.8 Mpps, and that the throughput is relatively constant regardless of packet size, until it finally hits the 40GbE line rate at packet sizes above about 1K Bytes. User space OVS provides some improvement, achieving a maximum throughput of about 12.1 Mpps and hitting line rate at packet sizes above about 400 Bytes. When we accelerate the OVS datapath by running it on the Agilio CX SmartNIC, the throughput reaches 28 Mpps, and reaches 40GbE line rate at 160 Bytes and above.

In the next test case, we use the same number of rules and flows while adding VXLAN tunnel processing. In this case, the VXLAN header is parsed and the VNI along with the inner destination MAC address are used to forward packets to VMs. The following graph displays the datapath throughput for OVS with VXLAN tunnel processing and forwarding to VMs:

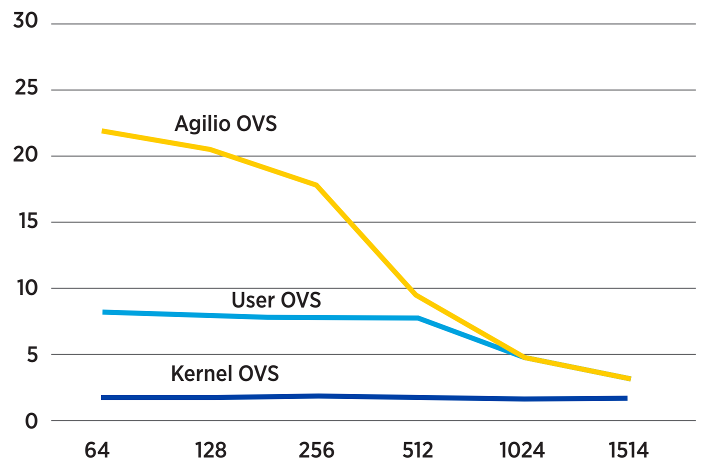

Figure 6. VXLAN Termination, L2 Forwarding, and delivery to VMs (MPPS)

We can see from these results that the standard OVS datapath running in the kernel degrades badly with VXLAN, only achieving a maximum throughput of about 1.5 Mpps, and that the throughput is relatively constant regardless of packet size, never reaching the 40GbE line rate even at the maximum packet size of 1514 Bytes. User OVS degrades also but not as steeply, achieving a maximum throughput of about 8 Mpps, but still only hitting line rate at packet sizes above about 512 Bytes. When we accelerate the OVS datapath by running it on the Agilio CX SmartNIC, the throughput stays relatively high, and we achieve 22 Mpps, hitting 40GbE line rate at packet sizes of 200 Bytes and higher.

These benchmark results clearly show the throughput improvement that is achieved when employing Agilio CX SmartNICs for offload of OVS compared to both kernel and user space software OVS implementations. An additional key finding is that throughput improvements become even more significant when datapath complexity increases. This can be observed when VXLAN tunnel endpoint processing is enabled, where packet-per-second throughput improvements for Agilio can reach as high as 15X over native OVS.

NORMALIZED PERFORMANCE AND WORK EFFICIENCY

The reader may have noticed that the previous comparisons are somewhat apples to oranges in the sense that dramatically different amounts of server CPU resources are being used in the software OVS case compared to Agilio OVS. In this section we will take a look at work efficiency. This can be done by normalizing the OVS throughput by dividing by the number of x86 cores needed to achieve the result. When we do that that, we see a dramatic difference in work accomplished per x86 core, as shown by the following graph:

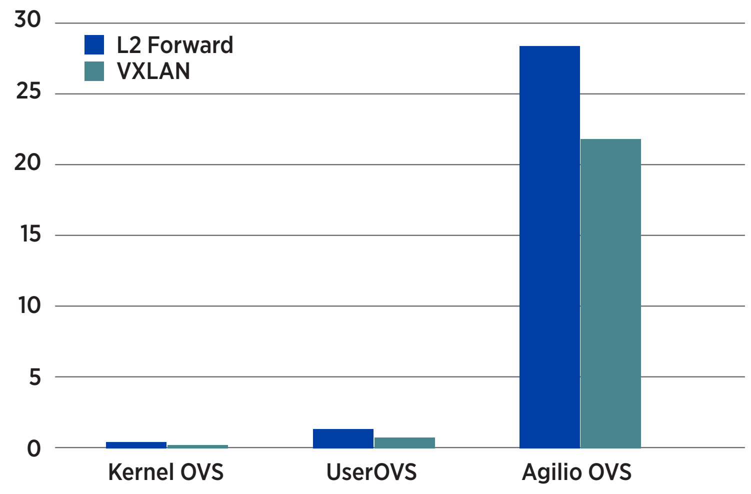

Figure 7. Normalized Performance (MPPS per single x86 Core)

CONCLUSIONS

The Agilio CX SmartNICs significantly outperform native software based implementations of OVS. Even when the maximum number of reasonably available server CPU cores is used for software OVS, the packets per second throughput of Agilio still outperforms Kernel OVS by at least 5X, and User OVS by at least 2X. When the server CPU resources are normalized, the Agilio CX advantage grows to over 50X over Kernel and 20X over User OVS, driving a massive improvement in datacenter efficiency.

This benchmarking confirms that Agilio SmartNICs can deliver more data to more applications running on a given server. This translates directly to improved server efficiency and a dramatic reduction in TCO, as less servers and data center infrastructure (such as switches, racks, and cabling) are needed to perform a given application workload.

Netronome Systems, Inc.
2903 Bunker Hill Lane, Suite 150 Santa Clara, CA 95054
Tel: 408.496.0022 | Fax: 408.586.0002
www.netronome.com

WP-OVS-BENCHMARKING-3/2017