Kaloom™ Virtual Switch:

The path to improve data center networking performance

Abstract

This Solution Brief provides an overview of the Kaloom Virtual Switch (KVS) and the high-level details regarding the role it plays within the Software Defined Fabric™data center networking solution.

Introduction

The KVS is an enhanced feature of the Software Defined Fabric, an innovative data center networking fabric targeted at data center infrastructure providers (DCIPs) and virtual data center operators (vDCOs) such as telecom and cloud service providers, large enterprises, gaming and financial institutions. It provides enhanced switching functionality that enables containers and virtual machines (VMs) running on application servers to communicate more efficiently. The KVS delivers significant improvements over other existing virtual switch solutions on the market by providing key benefits to data center networking performance with regards to latency and data throughput. Furthermore, by interacting directly with Software Defined Fabric, KVS can leverage Software Defined Fabric's advanced networking functions and offload virtual switching functionality onto Software Defined Fabric, thus freeing up valuable resources from application servers.

High Level Architectural Overview

The KVS uses the Berkeley Extensible Software Switch (BESS)[1] as its foundation to implement a virtual switch for the Linux platform. The BESS is not a virtual switch, but rather an open source modular framework and programmable platform for software switches. BESS's functionality enables the packet processing data path to be defined and configured by chaining small "modules" in a "pay for what you use" approach. As BESS does not run in the kernel of a given application server, but rather entirely in user space, it bypasses the kernel and binds directly to network interfaces using the Data Plane Development Kit (DPDK)[2]. The DPDK consists of a set of libraries used to create optimized high-performance packet processing applications on CPUs. By using the DPDK primarily, as well as, avoiding the overhead of the kernel networking stack, BESS, and as a result KVS, benefits from extremely fast performance.

Designed specifically to support Network Functions Virtualization, BESS does not sacrifice performance for programmability; offering high performance such as sub-microsecond latency and 40Gbps line-rate with minimum-sized packets on two cores[3]. By designing KVS with BESS as its foundation, KVS inherits key performance parameters and enables the delivery of a programmable platform for virtual data plane functionality for Software Defined Fabric.

Figure 1 highlights KVS within the overall Software Defined Fabric. As an advanced feature of Software Defined Fabric, KVS uses a proprietary protocol to communicate with the directly attached Leaf switches and offers the following advanced network functions:

- Direct offloading of L2 to L7[4] network functions such as frame flooding and MAC learning to the Software Defined Fabric which provides the benefit of significantly reducing the size of the MAC tables on the server.
- Accelerated flow distribution performance between directly connected Software Defined Fabric nodes.

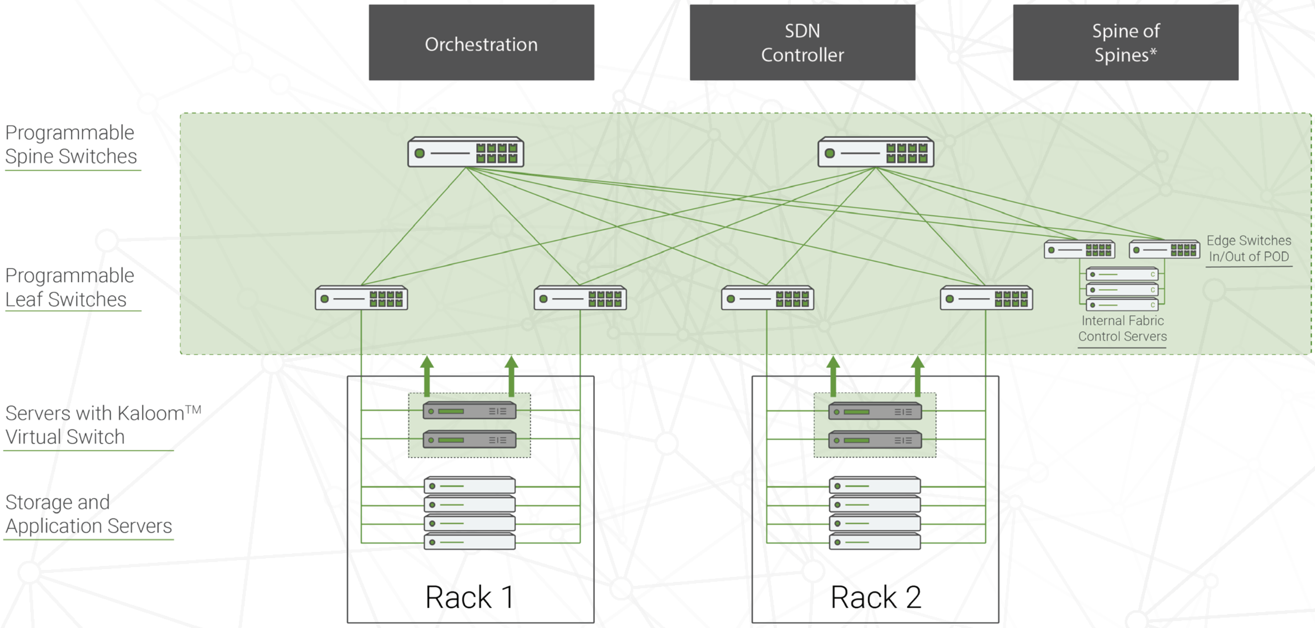
Orchestration
SDN Controller
Spine of Spines*
Programmable Spine Switches
Programmable Leaf Switches
Edge Switches In/Out of POD
Internal Fabric Control Servers
Servers with Kaloom™ Virtual Switch
Storage and Application Servers
Rack 1
Rack 2

Figure 1: Software Defined Fabric™ Topology.

KVS - Operations

Before elaborating on KVS's operational details, one is reminded that the essence of any virtual switch regardless of how it is implemented is to enable through software, virtual machine-to-virtual machine or application container-to-application container, communications. It accomplishes this by intelligently directing communications on the network by inspecting packets before passing them on. In the case of Kaloom's virtual switch functionality, its KVS implementation not only delivers on achieving the primary goal of virtual switch functionality, but also provides additional benefits to data center networking performance, as described further below.

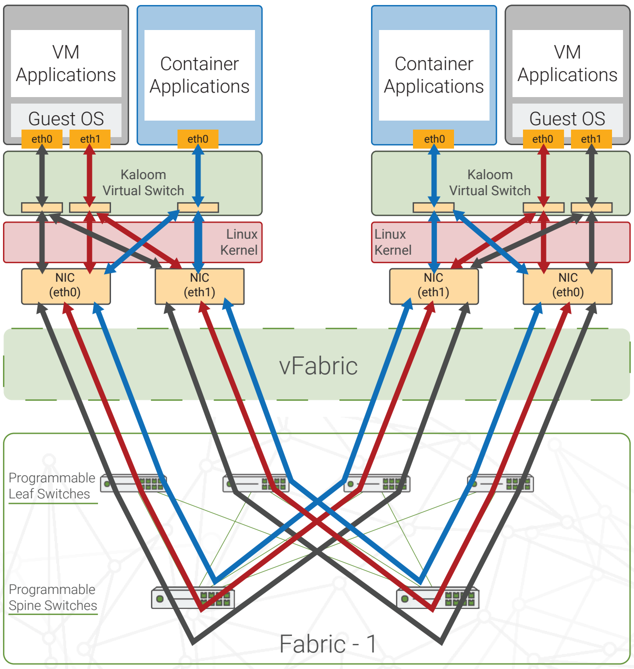

Figure 2: Kaloom Virtual Switch within the Fabric Topology.

From a deployment standpoint, the KVS software is configured via command line interface or RPC interface commands on the client's application servers located in the data center. The RPC interface is used by the ML2 plugin for integration with OpenStack management of virtual machines, and the CNI plugin for integration with Kubernetes management of containers. From an operational standpoint, KVS connects to a vfabric and can then attach containers or VMs to networks within the vfabric. Traffic between containers/VMs on the same server are forwarded by KVS to each other, with everything else being encapsulated in a Kaloom proprietary format known as Kaloom Normalized Format (KNF) and sent to the fabric which handles MAC learning and flooding. It should be noted that data traffic sent to the fabric does not need to terminate in another KVS but can terminate on an OVS instance using VXLAN or VLAN, or even a bare metal server that uses VLANs with a Linux bridge connected directly to the physical port. This operational paradigm allows the application servers to directly and more efficiently communicate with Software Defined Fabric, whereby the data plane specific virtual switching functionality is offloaded from the application servers and onto Software Defined Fabric.

Figure 2 provides an illustration of several data paths connecting various VM based and Kubernetes (a.k.a, k8s) container-based applications across the fabric.

KVS - A Replacement for OVS

Open vSwitch (OVS) is often regarded as a critical piece in a Software Defined Networking (SDN) solution. Serving as an open source virtual switch, it is typically used with hypervisors to interconnect virtual machines within a host and virtual machines between different hosts across the networks. OVS is integrated with Data Plane Development Kit (DPDK) and provides an option to use a DPDK optimized data path (OVS-DPDK), thereby providing a huge increase in network packet throughput and much lower latencies.

As part of its Software Defined Fabric offering, Kaloom provides support for this key networking functionality via its KVS, all the while ensuring interoperability with networks having OVS installations. As an embedded accelerated virtual switch with its own optimized algorithms, the KVS is an OVS-DPDK replacement that provides key performance benefits including reduced user-space-to-user-space latency and increased per-server data throughput.

The KVS Competitive Advantage

Kaloom has performed extensive testing to generate data comparing KVS versus OVS-DPDK. In brief, the test scenarios shown in figures 3a and 3b were used to produce the results provided later in this document. The test-beds had virtual switches; KVS or OVS-DPDK respectively, running on two different servers connected to Software Defined Fabric. One server had a virtual machine running a traffic generator while the other used DPDK's TestPMD application inside a virtual machine to forward packets between two interfaces. Measurements were made on traffic flowing from the traffic generator through the first virtual switch, the Software Defined Fabric, the second virtual switch, TestPMD and effectively looping back to the traffic generator using the same path in the opposite direction.

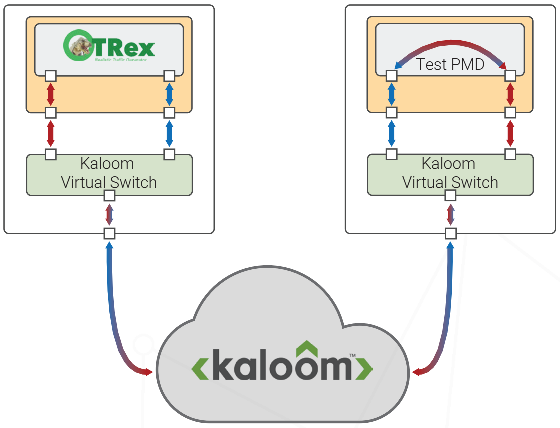

Figure 3a: Testbed using Kaloom Virtual Switch.

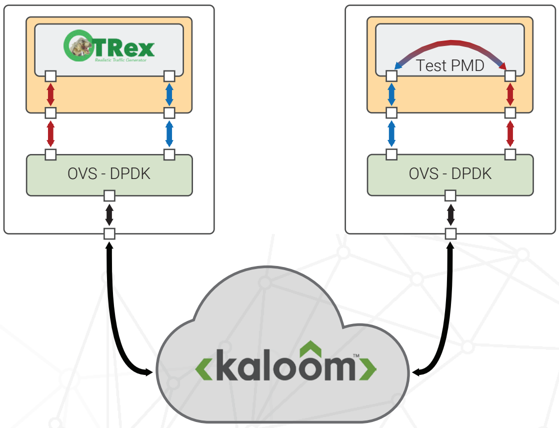

Figure 3b: Testbed using OVS-DPDK.

Figures 4a, 4b and 4c highlight KVS versus OVS performance comparisons related to average latencies measured in a network tested for various packet sizes, namely; 512, 1024 and 1514 bytes respectively. It can be clearly seen that in general, for larger packet sizes the latency in the network is much lower in the case of KVS versus OVS as the number of bidirectional flows is increased. For example, in the 1514 bytes packet size scenario, there is a factor of approximately seven times less latency observed for the network where KVS is installed versus that of OVS.

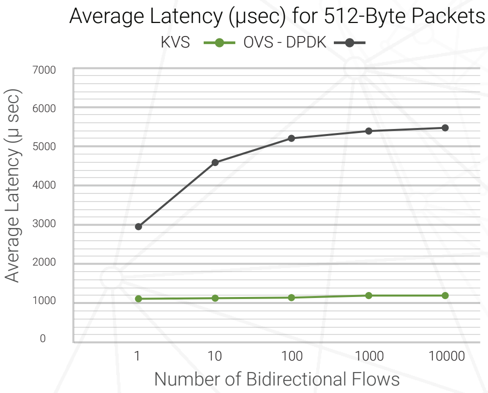

Figure 4a: Average latency performance comparison (KVS vs OVS) for 512-byte packets

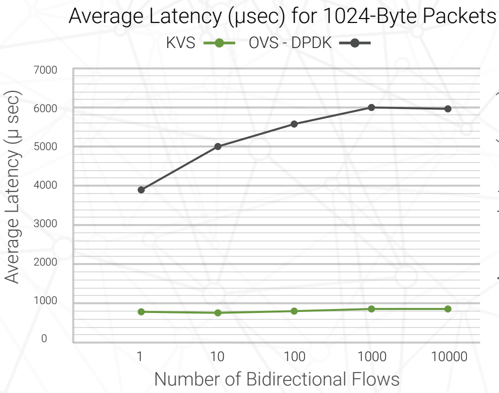

Figure 4b: Average latency performance comparison (KVS vs OVS) for 1024-byte packets

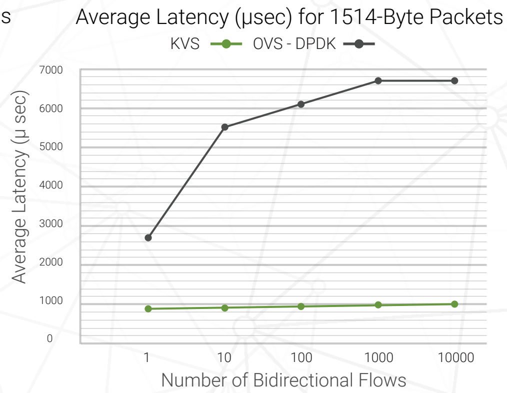

Figure 4c: Average latency performance comparion (KVS vs OVS) for 1514-byte packets

Figures 5a, 5b and 5c highlight and compare KVS versus OVS performance related to throughput. As can be seen for the same scenario as described previously, for the network with KVS installed, it can clearly be seen that for larger packet sizes, and as the number of bidirectional flows is increased, the throughput is higher versus that of OVS. For example, in the 1514 bytes packet size scenario, there is a factor of approximately two times the throughput.

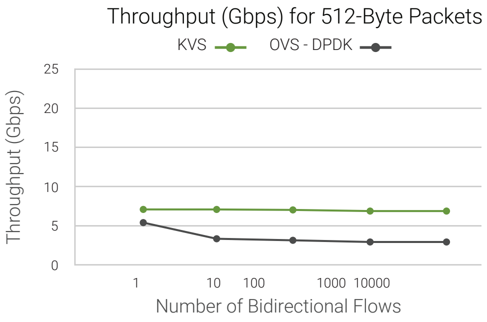

Figure 5a: Throughput performance comparison (KVS vs OVS) for 512-byte packets

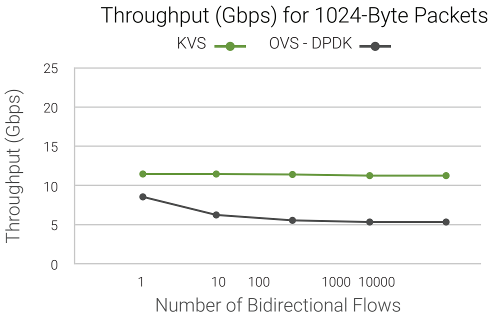

Figure 5b: Throughput performance comparison (KVS vs OVS) for 1024-byte packets

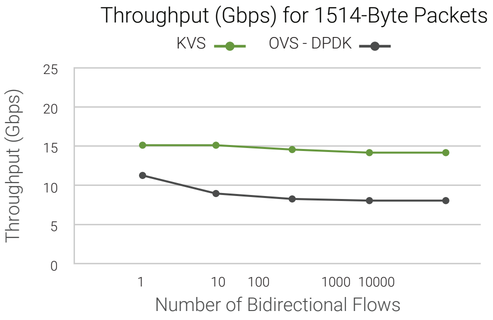

Figure 5c: Throughput performance comparison (KVS vs OVS) for 1514-byte packets

Conclusion

As an enhanced feature within the Software Defined Fabric product family, KVS offers an innovative virtual switch solution for data center infrastructure providers (DCIPs) and virtual data center operators (vDCOs) such as telecom and cloud service providers, large enterprises, financial institutions, etc. It extends networking functionalities by directly communicating with Software Defined Fabric to offload the virtual switching functionality from the application servers and onto Software Defined Fabric, thereby delivering more efficient execution of traditional and Software Defined Fabric advanced network functions. As an optimized virtual switch replacement for OVS-DPDK, KVS provides key benefits in networks where it is deployed. Its use enables the minimization of compute resources, while reducing user space to user space latency and increasing data throughput.

References & Notes

[1] For additional information on BESS, see https://github.com/NetSys/bess/

[2] For additional information on DPDK, see https://www.dpdk.org/

[3] For additional information on BESS, see https://software.intel.com/en-us/videos/bess-a-virtual-switch-tailored-for-nfv

[4] Currently support L2, with L3-L7 supported in the future (roadmap item)

For more information please visit: www.kaloom.com

Headquarters

355 Rue Peel, Suite 403
Montreal, Quebec, Canada
H3C 2G9

www.kaloom.com
info@kaloom.com