Published: 24 September 2017

Citation in BibTeX format

SoCC '17: ACM Symposium on Cloud Computing
September 24 - 27, 2017
California, Santa Clara

Conference Sponsors:
SIGOPS
SIGMOD

ABSTRACT

KVS: high-efficiency kernel-level virtual switch

HEUNGSIK CHOI, Korea University, Seoul, South Korea

GYEONGSIK YANG, Korea University, Seoul, South Korea

KYUNGWOON LEE, Korea University, Seoul, South Korea

CHUCK YOO, Korea University, Seoul, South Korea

SoCC '17: Proceedings of the 2017 Symposium on Cloud Computing (September 2017)
https://doi.org/10.1145/3127479.3131615
ISBN: 9781450350280

KVS: High-Efficiency Kernel-level Virtual Switch

Heungsik Choi, Gyeongsik Yang, Kyungwoon Lee, and Chuck Yoo
Department of Computer Science and Engineering, Korea University, Seoul, Korea
{hschoi,ksyang,kwlee,chuckyoo}@os.korea.ac.kr

CCS CONCEPTS

• **Networks → Cloud computing**; • **Software and its engineering → Memory management**;

KEYWORDS

Virtual switch, Kernel, DPDK, Open vSwitch

ACM Reference Format:
Heungsik Choi, Gyeongsik Yang, Kyungwoon Lee, and Chuck Yoo. 2017. KVS: High-Efficiency Kernel-level Virtual Switch. In *Proceedings of SoCC '17, Santa Clara, CA, USA, September 24–27, 2017,* 1 pages.
https://doi.org/10.1145/3127479.3131615

1 INTRODUCTION & DESIGN

In clouds, virtual switch (vSwitch) is in charge of packet forwarding between virtual machines (VMs). However, kernel-based vSwitches show throughput degradation for intensive packet processing; this becomes a bottleneck for the network performance of clouds. DPDK-based vSwitch (DPDK vSwitch) [1] has been developed to resolve the performance problem. Although it exhibits high throughput, DPDK vSwitch has two weak points. First, it consumes excessive memory. DPDK vSwitch uses huge page to reduce the number of memory operations, and this design causes high memory consumption even when the traffic is low. According to [2], memory determines the available number of VMs per single physical server. Thus, saving the memory decreases the capital expenditure of clouds. Second, security is another concern of the DPDK vSwitch, because its data plane is exposed to user space with the shared memory [3]. Therefore, the isolation of packets across VMs cannot be guaranteed. To overcome the excessive memory use and security concern, we propose a new kernel-level vSwitch (KVS) based on Linux. KVS do not use huge page nor bypass kernel stack. Instead, KVS applies the following key ideas to enhance the throughput.

Single SKB: Linux networking stack uses socket buffer (SKB) for packet processing, but the allocation of SKB is done per packet, which leads to high overhead in heavy traffic. SKB pre-allocation is a well-known optimization technique that allocates the multiple SKBs and data buffers in advance [3]. However, the Linux processes the incoming packets with only one instance of *SoftIRQ* handler per core at a time. Thus, when traffic comes at high speed, multiple SKBs cause unnecessary cache misses. To optimize the problem, we propose "single SKB" – that allocates just one dedicated SKB per core and reuses the one SKB for multiple data buffers of each core.

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA

ACM ISBN 978-1-4503-5028-0/17/09.
https://doi.org/10.1145/3127479.3131615

High-Performance EMC: Although the single SKB improves per-packet processing, vSwitch itself incurs overheads for packet forwarding. Most vSwitches use an exact match cache (EMC) [4], which consists of flow entries, to forward packets. Our profiling results for Open vSwitch (OVS) with single SKB show that "flow rule lookup" takes the most CPU cycles (14.34%). The main overhead for "flow rule lookup" we found is that vSwitch calculates the hash of each packet when it looks up the flow rule. So we propose a new "high-performance EMC" that uses RSS-hash-based indexing between the packet and flow rule. Specifically, when a flow rule is inserted, we implement the Toeplitz hash algorithm [5] to generate the RSS hash for the flow rule. When a packet looks up the flow rules, KVS utilizes RSS hash in the packet. Note that RSS hash is calculated in NIC hardware. Therefore, KVS saves CPU cycles by using RSS hash to look up the flow rule.

2 EVALUATION & CONCLUSION

We implement KVS based on OVS and deploy KVS on a server with an Intel Xeon E5-2630 V2 @ 2.60GHz, 48 GB memory, and a dual-port Intel 82599EB 10GbE NIC. We measure forwarding rate for 64B packets while generating network traffic at line rate on another machine connected via a 10 GbE link. The average number of last-level cache (LLC) misses per packet of the single SKB scheme is 0.55, while that of the pre-allocation scheme is 0.93. For memory usage, the KVS uses 645 MB of the memory per core, but the DPDK OVS consumes 1012 MB. In other words, KVS saves up to 36.3% of memory usage compared to DPDK OVS. On a single core, the throughput of KVS is 2.68 Gbps, which is 407% and 63.5% of the throughputs of Linux OVS (0.65 Gbps) and DPDK OVS (4.2 Gbps), respectively. Even though this throughput is lower than that of DPDK vSwitch, KVS is useful considering that memory is a critical resource in clouds [4]. In future, we plan to apply kernel based network optimization techniques on other network functions.

REFERENCES

[1] 2013. Intel DPDK vSwitch Getting Started Guide. (2013). https://01.org/sites/default/files/downloads/intel_dpdk_vswitch_040_gsg.pdf
[2] Sean Kenneth Barker, Timothy Wood, Prashant J Shenoy, and Ramesh K Sitaraman. 2012. An Empirical Study of Memory Sharing in Virtual Machines.. In *USENIX Annual Technical Conference*. 273–284.
[3] Sebastian Gallenmüller, Paul Emmerich, Florian Wohlfart, Daniel Raumer, and Georg Carle. 2015. Comparison of frameworks for high-performance packet IO. In *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*. IEEE, 29–38.
[4] Nick Shelly, Ethan J Jackson, Teemu Koponen, Nick McKeown, and Jarno Rajahalme. 2015. Flow caching for high entropy packet fields. *ACM SIGCOMM Computer Communication Review* 44, 4 (2015), 151–156.
[5] Wikipedia. 2017. Toeplitz Hash Algorithm — Wikipedia, The Free Encyclopedia. (2017). https://en.wikipedia.org/w/index.php?title=Toeplitz_Hash_Algorithm&oldid=775110540 [Online; accessed 6-July-2017].

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2016-0-00124, Research of Network Virtualization Platform and Service for SDN 2.0 Realization). This research was also supported by the MSIT(Ministry of Science and ICT), Korea, under the SW Starlab support program(2015-0-00280) supervised by the IITP(Institute for Information & communications Technology Promotion).