How Yahoo! JAPAN Used Open vSwitch* with DPDK to Accelerate L7 Performance in Large-Scale Deployment
Case Study

As cloud architects and developers know, it can be incredibly challenging to keep up with the rapidly increasing cloud infrastructure demands of both users and services. Many cloud providers are looking for proven and effective ways to improve network performance. This case study discusses one such collaborative project undertaken between Yahoo! JAPAN and Intel in which Yahoo! JAPAN implemented Open vSwitch* (OvS) with Data Plane Development Kit (OvS with DPDK) to deliver up to 2x practical cloud application L7 performance improvement while successfully completing a more than 500-node, large-scale deployment.

Introduction to Yahoo! JAPAN

Yahoo! JAPAN is a Japanese Internet company that was originally formed as a joint venture between Yahoo! Inc. and Softbank. The Yahoo! JAPAN portal is one of the most frequently visited websites in Japan, and its many services have been running on OpenStack* Private Cloud since 2012. Yahoo! JAPAN receives over 69 billion monthly page views, of which more than 39 billion come from smartphones alone. Yahoo! JAPAN also has over 380 million total app downloads, and it currently runs more than 100 services.

Network Performance Challenges

As a result of rapid cloud expansion, Yahoo! JAPAN began observing some network bottlenecks in its environment beginning in 2015. At that time, both cloud resources and users were doubling year by year, causing a rapid increase in virtual machine (VM) density. Yahoo! JAPAN was also noticing huge spikes in network traffic and burst traffic when breaking news, weather updates, or public service announcements, related to an earthquake for example, would happen. This dynamic put an additional burden on the network environment.

As these network performance challenges arose, Yahoo! JAPAN began experiencing some difficulties meeting service-level agreements (SLAs) for its many services. Engineers from the network infrastructure team at Yahoo! JAPAN noticed that noisy VMs (also known as “noisy neighbors”) were disrupting the network environment.

When that phenomenon occurs, a rogue VM may monopolize bandwidth, disk I/O, CPU, and other resources, which then impacts other VMs and applications in the environment.

Yahoo! JAPAN also noticed that the compute nodes were processing a large volume of short packets and that the network was handling a very heavy load (see Figure 1). Consequently, decreased network performance was affecting the SLAs.

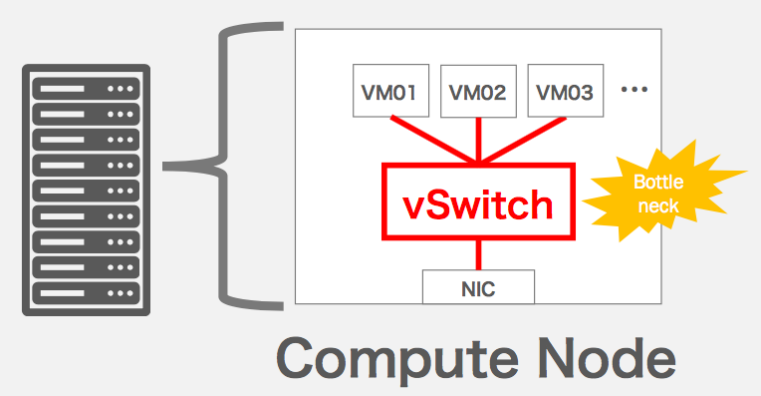

Figure 1. *A compute node showing a potential network bottleneck in a virtual switch.*

Yahoo! JAPAN determined that its cloud infrastructure required a higher level of network performance in order to meet its application and SLAs. In the course of its research Yahoo! JAPAN had noticed that the Linux* Bridge overrun counter was increasing, which meant that the cause of its network difficulties was located in the network kernel. As a result, the company decided it needed to find a new solution to meet its needs going forward.

About OvS with DPDK

OvS with DPDK could be a potential solution to such network performance issues in cloud environments that are already using OpenStack Cloud, since it features OvS as a virtual switch. Native OvS uses kernel space for packet forwarding, which imposes a performance overhead and can limit network performance. DPDK, however, accelerates packet forwarding by bypassing the kernel.

DPDK integration with OvS offers other beneficial performance enhancements as well. For example, DPDK's Poll Mode Driver eliminates context switch overhead. DPDK also uses direct user memory access to and from the NIC to eliminate kernel-user memory copy overhead. Both optimizations can greatly boost network performance. Overall, DPDK maintains compatibility with OvS while accelerating packet forwarding performance. Refer to Intel Developer Zone's article, Open vSwitch with DPDK

Overview, for more information.

Collaboration between Intel and Yahoo! JAPAN

As Yahoo! JAPAN was encountering network performance issues, Intel suggested that the company consider OvS with DPDK since it was now possible to use the two technologies in combination with one another. Yahoo! JAPAN was already aware that DPDK offered network performance benefits for a variety of telecommunications use cases but, being a web-based company, the company thought that it would not be able to take advantage of that particular solution. After discussing the project with Intel and learning about ways in which the technologies could work for a cloud service provider, Yahoo! JAPAN decided to try OvS with DPDK in their OpenStack environment.

For optimal performance deployment in OvS with DPDK, Yahoo! JAPAN enabled 1 GB hugepages. This step was important from a performance perspective, because it enabled Yahoo! JAPAN to reduce Translation Lookaside Buffer (TLB) misses and prevent page faults. The company also paid special attention to its CPU affinity design, carefully identifying ideal resource settings for each function. Without that step, Yahoo! JAPAN would not have been able to ensure stable network performance.

OpenStack's Mitaka release offered the features required for Yahoo! JAPAN's OvS with DPDK implementation, so the company decided to build a Mitaka cluster running with the configurations mentioned above. The first cluster includes over 150 nodes and uses Open Compute Project (OCP) servers.

Benchmark Test Results

Yahoo! JAPAN achieved impressive performance results after implementing OvS with DPDK in its cloud environment. To demonstrate these gains, the engineers measured two benchmarks: the network layer (L2) and the application layer (L7).

Table 1. Benchmark test configuration.

Hardware		Software	
CPU	Intel® Xeon™ processor E5-2683 v3 2S	Host OS	CentOS* 7.2
Memory	512 GB DDR4-2400 RDIMM	Guest OS	CentOS 7.2
NIC	Intel® Ethernet Converged Network Adapter X520-DA2	OpenStack*	Mitaka
		QEMU*	2.6.2
		Open vSwitch	2.5.90 + TSO patch (a6be657)

		Data Plane Development Kit	16.04

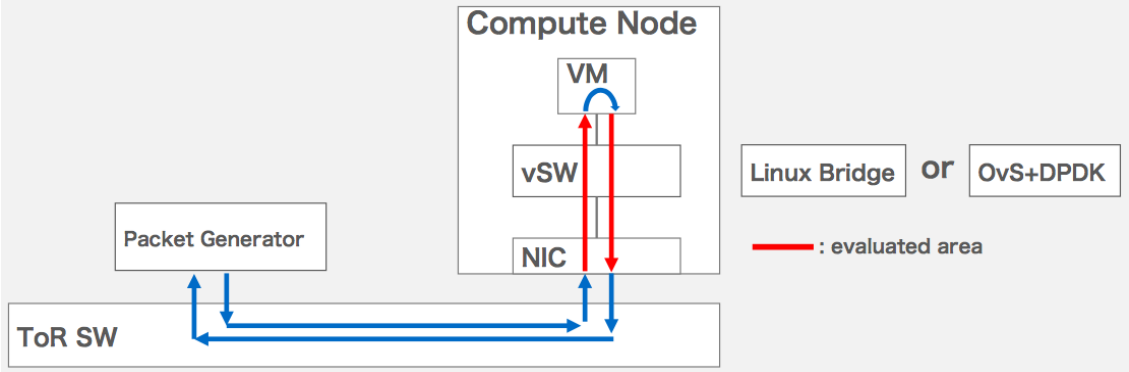

Figure 2. *L2 network benchmark test.*

L2 Network Benchmark Test Results

In the L2 benchmark test, Yahoo! JAPAN used Ixia IxNetwork* as a packet generator. Upon measuring L2 performance (see Figure 2), Yahoo! JAPAN observed 10x network throughput performance improvement in its short packet traffic. The company also found that OvS with DPDK reduced latency up to ~1/20x (1/20th). With these results, Yahoo! JAPAN successfully confirmed that OvS with DPDK accelerates the L2 path to the VM. These results were about in line with what Yahoo! JAPAN expected to find, as telecommunications companies had achieved similar results in their benchmark tests.

L7 Network Benchmark Test Results

The L7 single VM benchmark results for the application layer, however, exceeded Yahoo! JAPAN's expectations. In this test, Yahoo! JAPAN instructed one VM to send a query and another VM to return a response. All applications (HTTP, MQ, DNS, RDB) demonstrated significant performance gains in this scenario (see Figure 3). Particularly in the MySQL* sysbench result, Yahoo! JAPAN saw simultaneous improvement in two important metrics: 1.5x better throughput (transaction/sec) and 1/1.5x less latency (response time).

Apps	Linux Bridge -> OvS+DPDK
HTTP (nginx 1.10.2, apachebench 2.3)	2.0x
MQ (RabbitMQ 3.6.3, oslo.messaging/tools/simulator.py)	1.1x
DNS (bind 9.9.4, queryperf 1.12)	1.25x
RDB (MySQL/Percona 56-5.6.29, sysbench)	1.5x

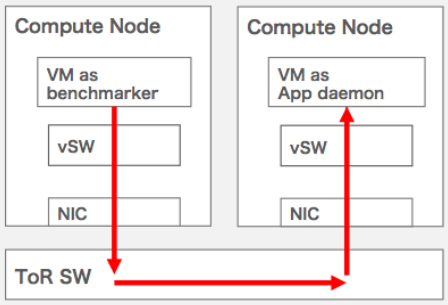

Figure 3. *Various application benchmark test results.*

Application Benchmark Test Results

Why did network performance improve so dramatically? In the case of HTTP, for example, Yahoo! JAPAN saw a 2.0x improvement in OvS with DPDK when compared to Linux Bridge. Yahoo! JAPAN determined that this performance metric improved because OvS with DPDK reduces the number of context switches by 45 percent when compared with Linux Bridge.

The benchmark results for RabbitMQ* revealed another promising discovery. When Yahoo! JAPAN ran their first stress test on RabbitMQ under Linux Bridge, it observed degraded performance. When it ran the same stress test under OvS with DPDK, the application environment maintained a much more consistent and satisfactory level of performance (see Figure 4).

	Linux Bridge	OvS+DPDK
1VM -> 1VM	1.0 (base)	1.1
10VMs -> 10VMs	0.49	0.98

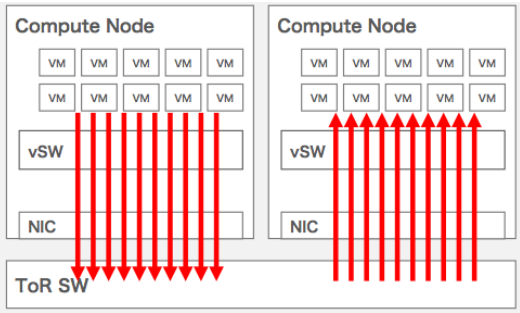

Figure 4. *RabbitMQ stress test results.*

RabbitMQ Stress Test Results

How was this possible? In both tests, noisy conditions created a high degree of context switching. In the Linux Bridge world, it's necessary to pay a 50 percent tax to the kernel. But in the OvS with DPDK world, that tax is only 10 percent. This is because OvS with DPDK suppresses context switching, which prevents network performance from degrading even under challenging real world conditions. Yahoo! JAPAN found that CPU

pinning relaxes interference between multiple noisy neighbor VMs and the critical OvS process, which also contributed to the performance improvements observed in this test. Which world would you want to live in: Linux Bridge or OvS with DPDK?

Ultimately, Yahoo! JAPAN found that OvS with DPDK delivers terrific network performance improvements for cloud environments. This finding was key to resolving Yahoo! JAPAN's network performance issues and meeting the company's SLA requirements.

Summary

Despite what you might think, deploying OvS with DPDK is actually not so difficult. Yahoo! JAPAN is already successfully using this technology in a production system with over 500 nodes. OvS with DPDK offers powerful performance benefits and provides a stable network environment, which enables Yahoo! JAPAN to meet its SLAs and easily support the demands placed on its cloud infrastructure. The impressive results that Yahoo! JAPAN has achieved through its implementation of OvS with DPDK can be enjoyed by other cloud service providers too.

When assessing whether OvS with DPDK will meet your requirements, it is important to carefully investigate what is causing the bottlenecks in your cloud environment. Once you fully understand the problem, you can identify which solution will best fit your specific needs.

To accomplish this task, Yahoo! JAPAN performed a thorough analysis of its network traffic before deciding how to proceed. The company learned that there was a high volume of short packets traveling throughout its network. This discovery indicated that OvS with DPDK might be a good solution for its problem, since OvS with DPDK is known to improve performance in network environments where a high volume of short packets is present. For this reason, Yahoo! JAPAN concluded that it is necessary to not only benchmark your results but also have a full understanding of your network's characteristics in order to find the right solution.

Now that you've learned about the performance improvements that Yahoo! JAPAN achieved by implementing OvS with DPDK, have you considered deploying OvS with DPDK within your own cloud? To learn more about enabling OvS with DPDK on OpenStack, read these articles: Using Open vSwitch and DPDK with Neutron in DevStack, Using OpenSwitch with DPDK, and DPDK vHost User Ports.

Acknowledgment

Thanks to this successful collaboration with Intel, Yusuke Tatsumi, network engineer for Yahoo! JAPAN's infrastructure team, said: "We found out that the OvS and

DPDK combination definitely improves application performance for cloud service providers. It strengthened our cloud architecture and made it more robust." Yahoo! JAPAN is pleased to have demonstrated that OvS with DPDK is a valuable technology that can achieve impressive network performance results and meet the demanding daily traffic requirements of a leading Japanese Internet company.

About the Author

Rose de Fremery is a New York-based writer and technologist. She is the former Managing Editor of *The Social Media Monthly*, the world's first print magazine devoted to the social media revolution. Rose currently writes about a range of business IT topics including cloud infrastructure, VoIP, UC, CRM, business innovation, and teleworking.

Notices

Testing conducted on Yahoo! JAPAN. Testing done by Yahoo! JAPAN.

Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at intel.com.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.