Introducing Virtual Switching

A virtual switch (vSwitch) is a software-based switch that allows communication between virtual machines (VMs) and containers. A popular option is Open vSwitch* (OVS), an open-source implementation of a virtual switch and a critical component for network virtualization in cloud computing.

Although Open vSwitch is an essential element for cloud computing, concerns exist about meeting cloud network performance and scalability needs. By taking advantage of the libraries provided by the Data Plane Development Kit (DPDK), shortcomings in OVS network performance are addressed and utilization of CPU cores is optimized. As a result, infrastructure cost is minimized and availability of CPU cores is maximized for cloud service delivery.

Data Plane Performance Gains with DPDK-Accelerated OVS

OVS, when built with DPDK (aka OVS-DPDK) shows greatly improved performance over native OVS against the following indicators:

- **Network performance**: throughput, latency, and jitter
- **Network stability**: reliability and scaling
- **Efficiency**: performance per core

OVS-DPDK, depicted in Figure 1, is a DPDK-based application running on top of DPDK libraries. As a result, infrastructure cost is minimized and availability of CPU cores is maximized for cloud service delivery.

With the substantial performance gained by DPDK-accelerated OVS, Yahoo! JAPAN* acknowledged:

“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.”

![Figure 1. OVS-DPDK Overview](image-url)
When compared to native OVS, the forwarding plane in OVS-DPDK moves from kernel to user space, which helps to improve system stability. OVS-DPDK uses poll mode drivers (PMD) instead of the interrupt-based kernel mode drivers used in native OVS. Since a PMD polls for network packets, it eliminates the needs for network traffic interrupts and interrupt service routines for network data capture. Interrupt elimination helps improve network performance by avoiding expensive context switching overhead. In contrast, a PMD running in user space communicates directly with the hardware, bypassing kernel completely. This mechanism not only reduces network latency by removing data copy overhead from kernel to user space but also achieves better resource usage by not allocating an intermediate memory buffer in the kernel.

Implementing OVS-DPDK, Cloud Service Providers are able to achieve significant performance benefits over native OVS.

Yahoo! JAPAN Case Study: Throughput Increase and Latency Reduction

- For L2 network performance, Yahoo! JAPAN observed approximately 10 times greater throughput performance in its short packet traffic and about 1/20th the latency.
- Table 1 shows significant performance gains for other application L7 traffic types, including HTTP, MQ, DNS, and RDB.

<table>
<thead>
<tr>
<th>Applications</th>
<th>Linux* Bridge-&gt; OVS-DPDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP (nginx, apachebench)</td>
<td>2.0 times</td>
</tr>
<tr>
<td>MQ (RabbitMQ, oslo.messaging/tools/simulator.py)</td>
<td>1.1 times</td>
</tr>
<tr>
<td>DNS (bind, queryperf)</td>
<td>1.25 times</td>
</tr>
<tr>
<td>RDB (MySQL, sysbench)</td>
<td>1.5 times</td>
</tr>
</tbody>
</table>

Table 1. Performance Gains from Implementing OVS-DPDK

• Figure 2 shows how the vSwitch running on a compute node was tested using a packet generator for L2 testing.
• Figure 3 shows the use of the vSwitch in a simplified network topography for L7 testing.
When VM scaling was tested using RabbitMQ (Figure 4), the performance of the compute node with OVS-DPDK scaled linearly while the compute node with a Linux Bridge had half the application performance (Figure 4).

![Figure 4](Image)

<table>
<thead>
<tr>
<th>Number of VMs</th>
<th>Linux* Bridge</th>
<th>OVS-DPDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1VM</td>
<td>1.0 (base)</td>
<td>Perf Degraded</td>
</tr>
<tr>
<td>10 VMs</td>
<td>0.49</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Figure 4. Comparative performance of VM scaling between Linux Bridge and OVS-DPDK

With the substantial performance gained by DPDK-accelerated OVS, Yahoo! JAPAN acknowledged:

“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.”

![Figure 5](Image)
Meituan* Case Study: Throughput Increase

Meituan tested the vSwitch performance of an OVS loopback configuration and found the OVS-DPDK throughput performance was 4.2 times higher than native OVS. The results and test setup are shown in Figure 6.

- The vSwitch performance test of the OVS loopback configuration also showed the throughput performance for both non-tunnel and VxLAN networks were substantially higher with OVS-DPDK than with native OVS. It also scaled well with an increasing number of VMs, as shown in Figure 7.

Meituan states:

“OVS-DPDK removes many performance pain points, especially for VM live migration and VM vhost-user reconnection.”

Figure 6. Test Results and Simplified Topology for Throughput Test

Figure 7. Throughput Results and Simplified Topology for VxLAN Tunneled With an Increasing Number of VMs
Intel® Architecture Capabilities for DPDK-Based vSwitch Acceleration

Many Intel® architecture instruction sets and Intel® platform technologies are used to optimize DPDK-accelerated OVS performance.

Intel® Transactional Synchronization Extensions (Intel® TSX)

Intel TSX is a new synchronization feature that replaces thread serialization used by traditional locks. This feature can reduce thread serialization, resulting in significant latency improvements.

Figure 8 shows the use of Intel TSX in OVS can increase the insertion rate of the flow table by up to 11 times. In this example, performance can also scale linearly with the number of cores used. To learn more about optimizations for flow table insertion using Intel TSX, click here.

Intel® Advanced Vector Extensions (Intel® AVX)

“Vectorization of loops” with Intel AVX can help improve flow table lookup performance by allowing software to inspect multiple packets in parallel instead of single packets sequentially.

Using Intel AVX, lookup throughput performance can be improved by up to 35 percent and maintained with scaling the number of flows. More flow table lookup performance optimizations using Intel AVX can be found here.

Other Intel platform technologies and tools to improve vSwitch performance

Setting Huge Pages to 1 GB can help reduce memory access overhead by decreasing the number of page table lookups, TLB misses, and page faults.

CPU isolation/affinity mechanism can be used to remove certain CPUs from the OS scheduler and pin them to desired VMs and PMDs. This approach can increase performance and network stability by reducing interference from “noisy neighbor” VMs.

Non-uniform memory access (NUMA) awareness can help increase performance by enabling the CPU/Memory/PCI to run on the same node, which can reduce the overhead from transferring data from one node to another through the Intel® UtraPath Interconnect (UPI) bus.

Intel® VTune™ Amplifier, a long-established tool for Intel architecture development provides powerful capabilities to monitor/debug key system metrics like cache misses, branch mis-prediction, DTLB misses, long latency instructions, and exceptions. These metrics help engineers identify network performance bottlenecks and develop optimizations for OVS-DPDK performance.

Figure 8. Improvement in Insert Rate Following TSX Implementation in OVS
OVS-DPDK for Cloud Service Providers

vSwitch is an essential component of network solutions for Cloud Service Providers. As these technical case studies have shown, DPDK-accelerated OVS solves critical performance and bottleneck issues caused by native OVS. Removing these issues allows Cloud Service Providers to optimize their virtualized network and CPU core allocation for maximum efficiency, scalability and monetization potential.

For more information about Intel solutions for network infrastructure, visit:

- Intel® Network Builders – Find information related to Network Functions Virtualization (NFV), DPDK, etc.
- Intel® Developer Zone – Use OVS in conjunction with OpenStack®
- OpenvSwitch with DPDK Overview
- OVS with DPDK Installation Guide
- Yahoo! JAPAN Used Open vSwitch with DPDK to Accelerate L7 Performance in Large-Scale Deployment Case Study
- OVS-DPDK Practices in Meituan Cloud