从计算机架构师的角度看DPDK性能

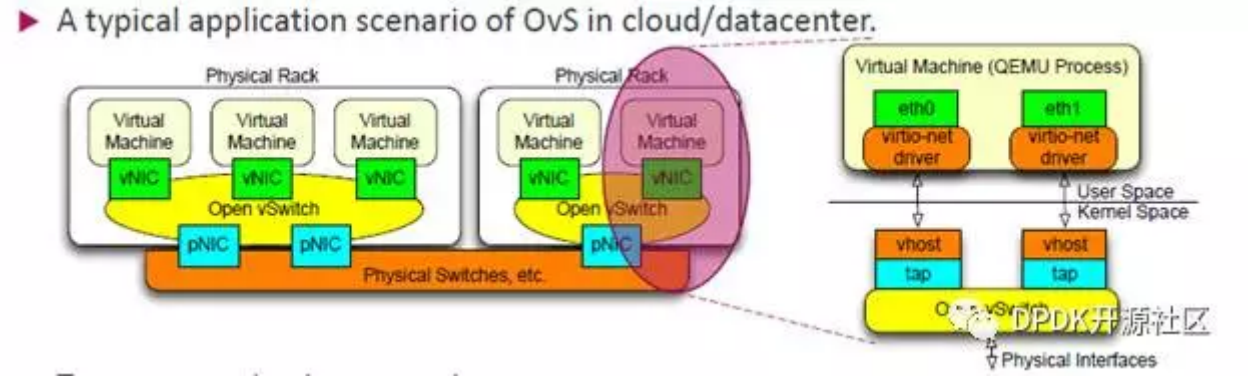
原创 DPDK开源社区 2016-12-21

作者 Dr. Peilong Li

对于典型的数据中心或者云，为了充分利用硬件，往往依赖于虚拟化技术。在这种情况下，OvS是云和数据中心在提供虚拟机网络方面的关键连接组件，比如OpenStack，以及OpenNebula。但是，问题是随着线路速率从10Gb增长到40Gb再到100Gb，OvS很难跟上这样的增长速度。因此，为解决这个问题，基于DPDK的OvS得以开发。确实，基于DPDK的OvS的性能高于Vanilla OvS，但这是为什么呢？接下去我们将一探究竟。

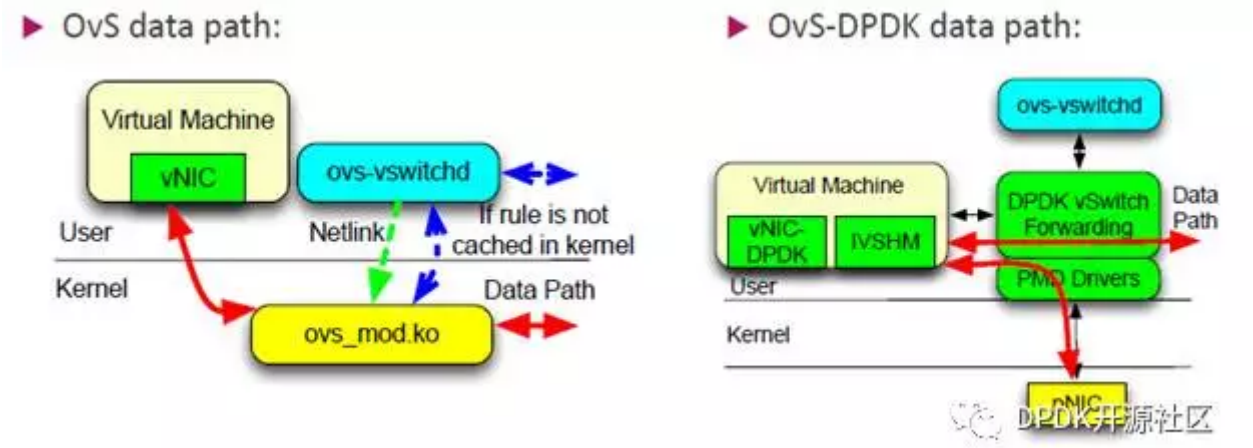
云/数据中心OvS典型应用场景

有两种基本的虚拟机通信场景：一种是不跨主机的虚拟机间通讯；另一种是需要经过物理网卡的跨主机的虚拟机间通讯。

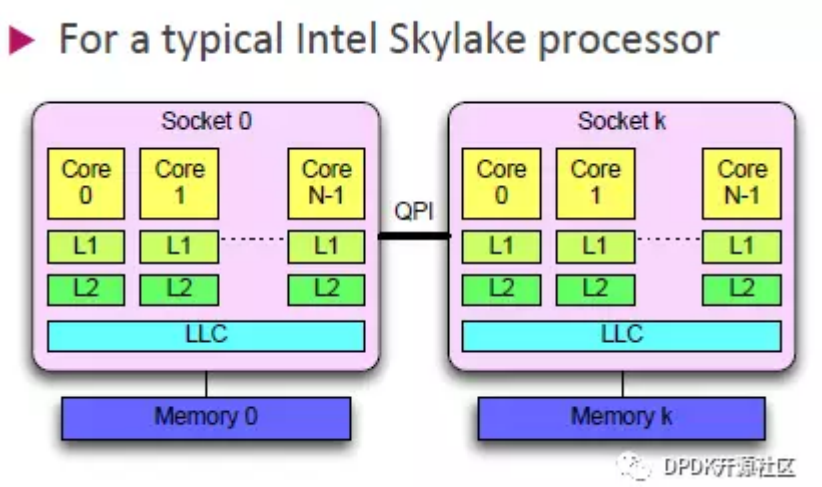


OvS，OvS-DPDK I/O比较

Vanilla OvS有两个典型的组件，一个是位于内核空间的kernel module，另一个是位于用户空间的OvS daemon。典型场景下，流量会从一台虚拟机进入内核module，然后去向别处。但是如果相应的rule没有缓存在内核module中，那么内核module就需要和OvS daemon通信来获取相应的rule。而这就会导致很多用户空间和内核空间的上下文切换。在OvS-DPDK中，整个data path都在用户空间。如果想要和外界通信，可以直接绕过内核，通过PM Ddriver与物理网卡通信。



典型的处理器有很多的core。每个core都有自己的L1 cache、L2 cache，并共享LLC cache。离core越远，就会有越大的访问开销。



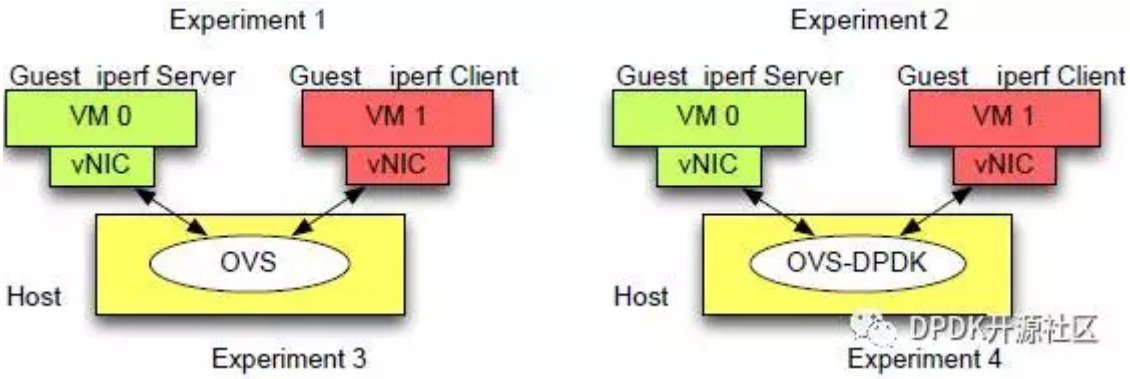
下图数据来源：*Intel 64 andIA-32 Architectures: Optimization Reference Manual*

Parameters	Value
L1 Peak Bandwidth (bytes/cycle)	2x32 Load 1x32 Store
L2 Data Access (cycles)	12
L2 Peak Bandwidth (bytes/cycle)	64
Shared L3 Access (cycles)	44
L3 Peak Bandwidth (bytes/cycle)	32
Memory Access (cycles)	~ 140 (for 2.0 GHz)

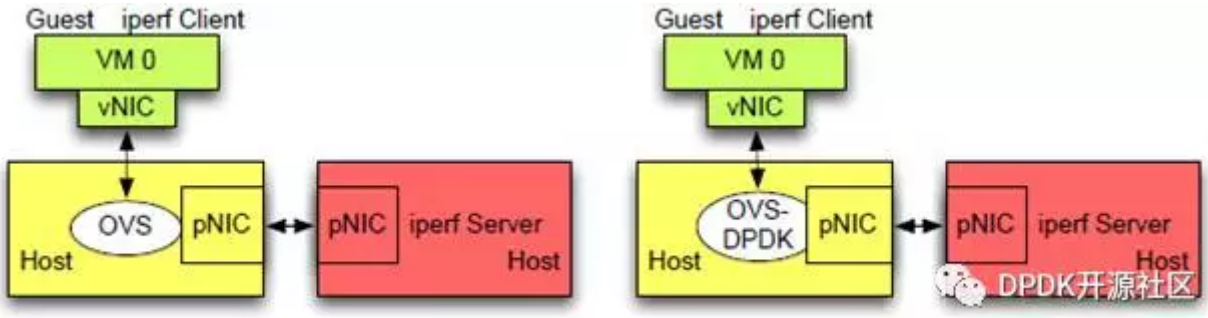
DPDK开源社区

测试方案

在Guest-Guest (VM2VM)中，我们有一个物理host，运行2个虚拟机，它们可以通过OvS或者OvS-DPDK通信。



在Guest-Host(VM2Host)中，有2 个物理host，一个运行虚拟机并运行OvS或者OvS-DPDK，另一个运行iperf服务器。



具体配置信息如下：

- ***Hardware*** - Intel SuperMicro Server
  - Intel Xeon D-1540, 8 Cores @ 2.0 GHz.
  - ***L1i***: 32 KB, ***L1d***: 32 KB, ***L2***: 256 KB, ***LLC***: 12 MB, ***Memory***: 64 GB.
  - ***NIC***: Intel 82599ES 10-Gigabit SFI/SFP+
- ***OS***: Ubuntu 16.04; ***OvS*** version: 2.5.0; ***DPDK*** version: 16.04
- All the VMs are created by KVM and emulated by QEMU.
- Run ***Iperf*** (version 2.0.5) test on the provided environment.
- Processor performance profiling tools:
  - Linux ***Perf*** version: 4.4.13
  - Intel ***VTune Amplifier XE*** version: 2016 Update 4

DPDK开源社区

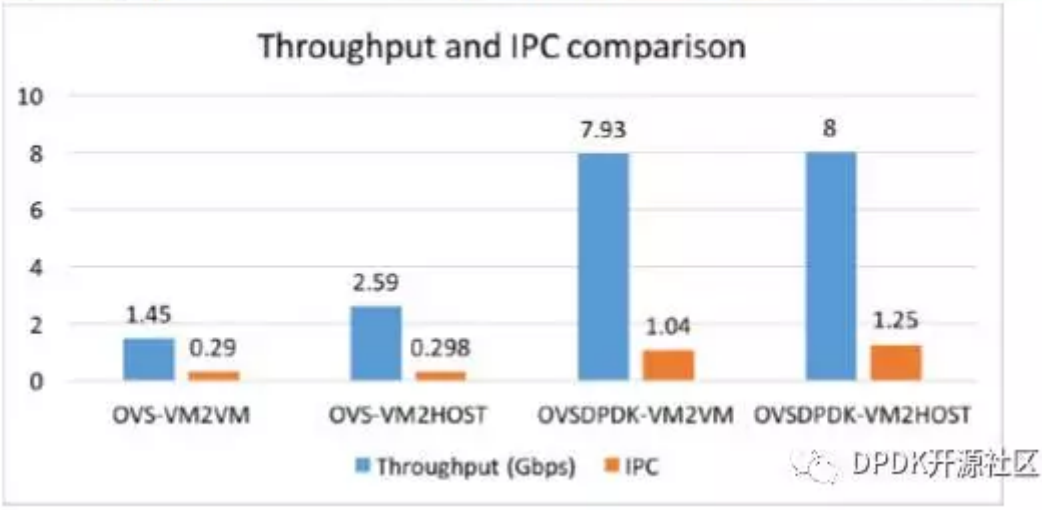
Iperf测试命令如下：

- Experiment 1 (*VM-OvS-VM*)
  - On VM0 (Iperf Server)
    - sudo iperf -s -w 512k -l 128k -p 1005 | grep SUM
  - On VM1 (Iperf Client)
    - iperf -c 10.0.0.1 -p 1005 -w 512k -l 128k -i2 –t60 -P4 | grep SUM
- Experiment 2 (*VM-OvSDPDK-VM*)
  - Same as experiment 1, but use OvS-DPDK
- Experiment 3 (*Host-OvS-VM*)
  - Same as experiment 1, but use another host machine as server
- Experiment 4 (*Host-OvSDPDK-VM*)
  - Same as experiment 3, but use OvS-DPDK.

Evaluation 1 四种场景下吞吐量和ICP比较

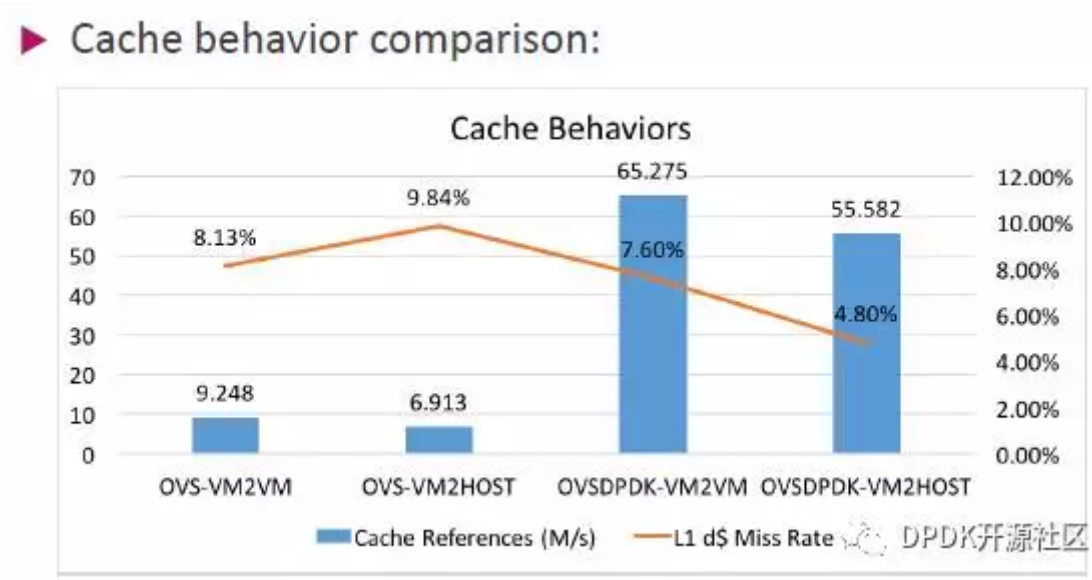
OvS-DPDK-VM2VM的吞吐量是OvS-VM2VM的5.5倍；OvS-DPDK-VM2HOST的吞吐量是OvS-VM2HOST的3倍左右。从IPC来看，对于典型的4-issue架构，理想的IPC是4，如果我们没有使用OvS-DPDK，IPC会低于1，差不多是0.3或0.2；而使用了OvS-DPDK，这一数值会大大提高。

- Throughput and IPC comparison for 4 different scenarios:



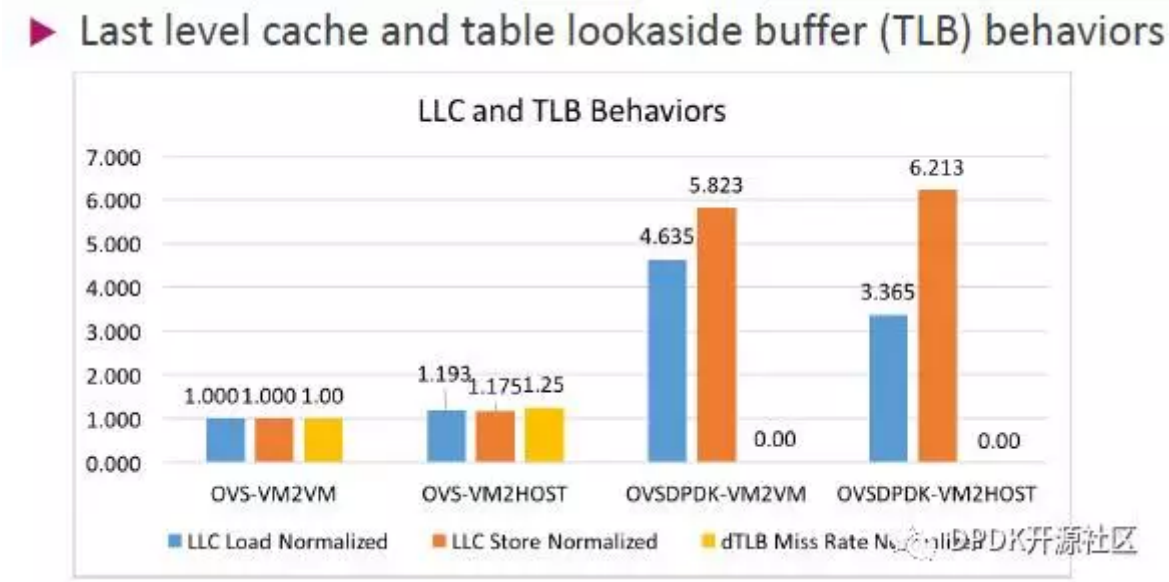
Evaluation 2 Cache 行为比较

第一个蓝色柱子和第三个蓝色柱子相比，我们可以看到有七倍左右的增长；第二个蓝色柱子和第四个蓝色柱子相比，我们可以看到有八倍左右的增长。也就是说当我们使用OvS-DPDK时会有更多的cache hits，更少的cache misses。图中的L1data miss rate，从9.84%下降到4.8%。这得益于DPDK的预取机制。



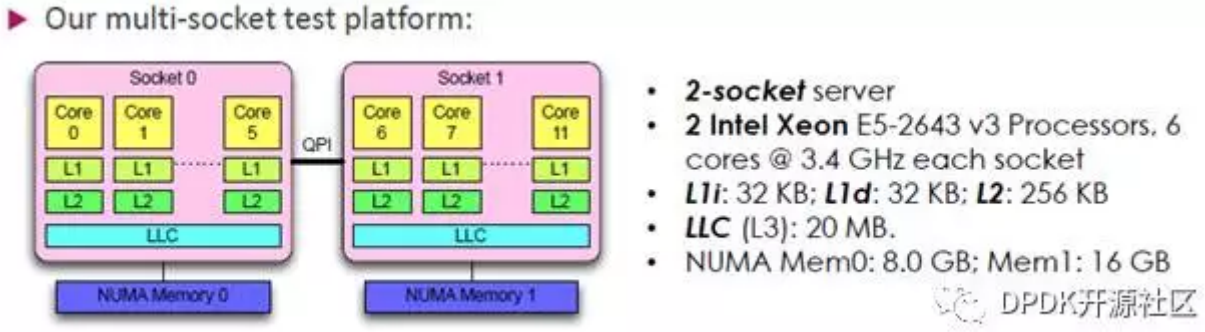
Evaluation 3 LLC Cache和TLB 行为

从图表中可以看出，使用OvS-DPDK时, TLB的 miss rate几乎为0。这得益于大页的应用。



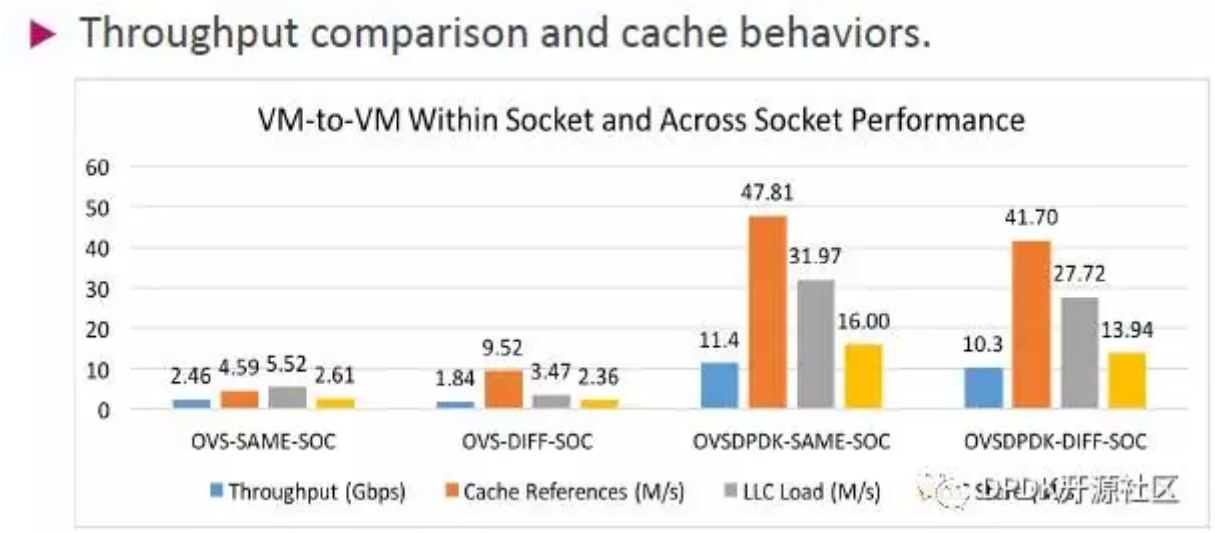
虚拟机之间的跨socket通信

现代的数据中心通过部署多路处理器平台来扩展性能。我们的测试平台如下：



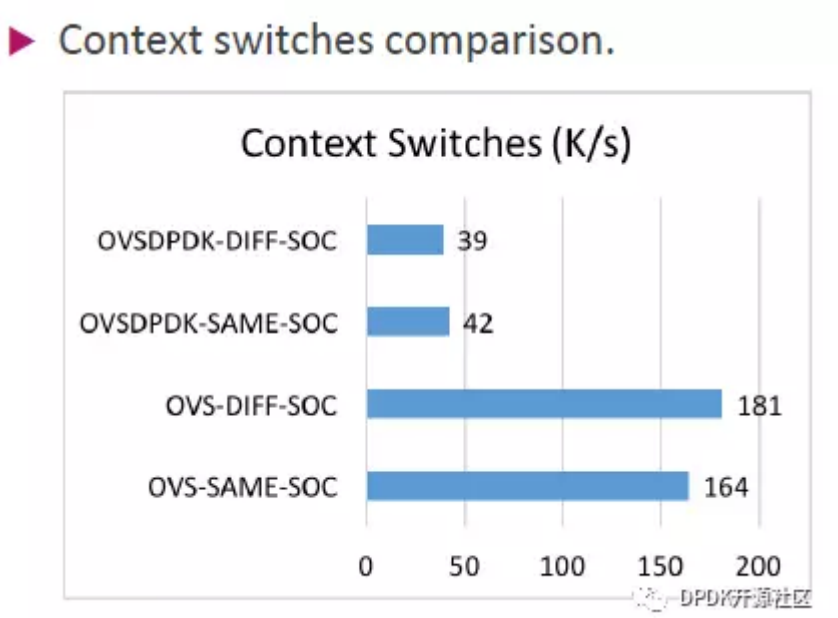
Evaluation 4吞吐量和cache行为比较

跨socket的设计会对性能有不利影响。在有多socket的平台中，比较好的做法是如果有很高的带宽，则使用同一个socket。在同一个socket中，会得到更高的吞吐量和更好的LLC行为。



Evaluation 5 上下文转换比较

相对于原始的OvS，OvS-DPDK有更少的上下文切换。此外， 跨socket的通信并不是导致上下文切换的根本原因。



总结

本文从计算机架构师的角度详细分析了Vanilla OvS 和OvS-DPDK的性能。总的来说有两点：A. OvS-DPDK通过以下两点提高系统性能：1.增加IPC和cache hits；2. 减少cache miss（软件预取功能），减少TLB miss（大页），减少上下文切换（用户空间驱动）。

B. 多socket 平台中可能导致：1. 更低的系统吞吐量和更少的LLC命中；2. 但跨socket的通信不是导致上下文切换的根本原因。

作者简介

Dr. Peilong Li，2016年获得马萨诸塞大学洛厄尔分校计算机工程博士学位。现今在该大学电气和计算机系从事博士后研究工作。他的研究领域包括异构和并行计算体系结构、分布式计算框架研究，软件定义网络的数据平面创新等等。

00:00/00:00 下载视频 倍速

前沿科技：超级高铁首测成功：2秒钟加速640千米每小时 用腾讯视频观看

DPDK开源社区 | 一个有用的公众号

长按，识别二维码，加关注

投诉