**DPDK在虚拟化环境中的应用**

* 实验目的：

进一步熟练KVM虚拟化技术的基本使用；

了解以DPDK为代表的网络优化技术；

了解DPDK在虚拟化环境中的应用。

* 实验背景知识

DPDK(Data Plane Development Kit)是Intel研发的一种网络优化框架，利用了批处理，零拷贝，无锁队列，用户空间网卡驱动，多队列网卡等多种网络优化技术来加速应用程序对网络数据包的处理。具体介绍如下：

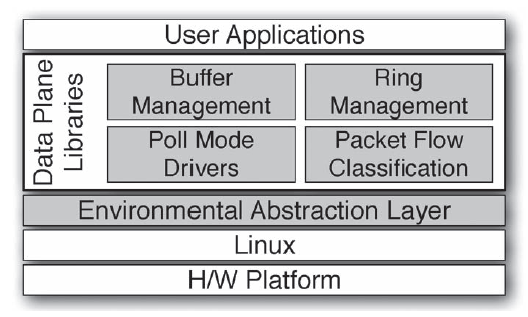
首先，用户空间网卡驱动让用户空间的应用程序可以直接使用轮询（Poll）的方式从“网卡中”读取数据（实际是从网卡驱动与应用程序共享的内存）。使用用户空间的网卡驱动意味着应用程序读取数据包时可以绕过内核，避免了不必要的内存拷贝和一系列系统调用，也就是避免了二者的空间切换；使用轮询而不是中断，避免了高速网络I/O中中断上下文切换的开销，避免了“活锁”现象，有利于高性能网络中数据包的处理。

其次，DPDK利用了内存巨页（huge page），可以减少DPDK程序所需要的TLB（Translation Lookaside Buffer）表项，从而有效地降低TLB表的miss率。

再次，DPDK使用无锁队列，其底层技术是支持多“生产者”和多“消费者”（Multiple producer and Multiple consumer）的、无锁的Ring，这使得DPDK应用程序的不同线程间可以无锁地进行数据通信，减少了线程等待时间也降低了编程难度。

最后，DPDK工作在用户空间，而DPDK应用程序的线程调度依然依赖内核。DPDK利用线程的CPU亲和绑定的方式，使特定任务可以被指定工作在某个核上，从而避免了线程在不同核间频繁的切换，减少了cache miss和cache write back造成的大量性能损失。

DPDK在用户程序看来是以一种库的形式出现的，其架构如下图所示。



**图1 DPDK的架构示意图**

总地来看DPDK应用程序运行在DPDK的库之上，而DPDK的底层是一个被称为“环境抽象层”（Environmental Abstraction Layer, EAL）的覆盖层，EAL将底层操作系统的功能进行抽象为上层的库提供相同的运行环境（虽然都使用Linux内核，但不同的操作系统所支持的功能不同）。

它的库主要有四部分：buffer管理相关的库、ring管理相关的库、轮询模式的驱动和数据包的分类相关的库。

其中，通过buffer相关的库应用程序可以申请和释放存放数据包的“缓冲区”（buffer），在DPDK中网络数据包的内容和包头都会存放到buffer中；ring如前文所述，支持多“生产者”和多“消费者”访问，并且是无锁实现的，应用程序可以通过ring实现线程间通信；轮询模式的网卡驱动工作在用户空间，这样便可绕过内核，而当网卡收到的数据包特别多时，轮询模式会明显优于中断模式；DPDK允许使用许多提前定义的规则来对收到的数据包进行分类，这可以用来实现访问控制、流的识别和QoS管理等功能。

总结一下，DPDK的核心技术是通过运行在用户空间的网卡驱动使用户程序可以“穿透”操作系统，直接与网卡交换数据。DPDK也可以使用在虚拟化环境下，在虚拟机中使用DPDK当然可以让虚拟机中的用户程序“穿透”虚拟机的操作系统。

* 实验概述：

本实验主要内容是在两台虚拟机上编译安装DPDK库，在其上分别运行DPDK应用程序，测试虚拟机间的“二层网络数据转发”性能。

其中，在虚拟机0上运行DPDK的“二层网络数据包转发”示例程序（L2fwd），这里，该示例程序将网口中接收的数据包原路转发回去。

在虚拟机1上运行pktgen-dpdk程序，这是一个运行在DPDK上的应用程序，用来进行高速地收发网络数据包，这里被用来向虚拟机0发送数据包，并接收虚拟机0转发过来的数据包，并统计。

* 实验步骤：

1. 参照“KVM基础实验”创建两台虚拟机，二者都运行Linux系统(建议使用Ubuntu Server 16.04.1 LTS amd64)；
2. 分别在两台虚拟机中编译安装DPDK库：
3. 下载DPDK源代码；
4. 根据官网指导（参考资料3）编译DPDK库；
5. 参考资料3，进行虚拟机huge page的设置，加载PMD（Poll Mode Driver，即前文中的用户态网卡驱动），对网卡进行绑定。
6. 在虚拟机1上编译安装、使用pktget-dpdk：
7. 下载pktgen-dpdk源代码；
8. 参照pktgen-dpdk官网编译并运行，将运行结果截图。
9. 在虚拟机2上编译使用DPDK库中的“二层网络数据转发”示例程序L2fwd：
10. 进入DPDK源代码目录的“examples/l2fwd”目录，使用make指令编译；
11. 运行l2fwd示例程序，将运行结果截图。
12. 同时在虚拟机1上运行pktgen-dpdk在虚拟机2上运行L2fwd，测试虚拟机间的“二层网络数据转发”性能：
13. 在pktgen-dpdk程序界面中通过指令通知所发数据包的大小（从64字节到1500字节），记录L2fwd所转发过来的数据包的数量；
14. 绘制L2fwd所转发数据包的数量随数据包大小的变化图，分析性能变化原因（实验重点之一）。

* 参考资料

1、虚拟机的创建：“KVM基础实验”指导书；

2、DPDK源代码下载：推荐16.07版本，DPDK官网（http://dpdk.org/download）；

3、DPDK的编译安装：DPDK官网指导（http://dpdk.org/doc/quick-start，http://dpdk.org/doc/guides/linux\_gsg/build\_dpdk.html）；

4、pktgen-dpdk源代码下载：推荐3.0.14版本，http://dpdk.org/browse/apps/pktgen-dpdk/refs/；

5、pktgen-dpdk的编译使用：pktgen-dpdk官网（http://pktgen.readthedocs.io/en/latest/）；

6、L2fwd示例程序的编译使用：DPDK官网示例程序使用指导

（http://dpdk.org/doc/guides/sample\_app\_ug/l2\_forward\_real\_virtual.html）；

* 要求与提示

重要实验步骤需给出截图。

编译安装DPDK、pktgen-dpdk、L2fwd时可能会遇到种种问题，请先根据报错提示在网上寻找解决方法，实在解决不了，请及时联系助教。