DPDK 测试报告

编　号：ZKJS-Q-2-14-01-JT-SB 2017-02/0 V1.0

版本/修订：V1.0

编　制　何晓伟

审　核　王海霞

会　签

批　准　王凤丽

文档修订历史

版本	作者	版本变化对象	变化内容描述	审核人	批准人	修订日期

目 录

1 概述

。

1.1 文档目的

当高速网卡传输大量数据时，会产生频繁的中断，影响操作系统的执行效率。采用 DPDK 加速工具，利用 DPDK 的大页内存技术，CPU 亲和技术，轮询技术来优化网卡和操作系统的性能。本次测试利用 DPDK 的 testpmd 测试工具，主要测试 DPDK 在 intel 服务器（centOS 7 系统）上的数据吞吐量，以便为后期使用提供参考。

1.2 术语及缩略语

英文缩写	英文全称	中文全称

2 测试环境搭建

2.1 大页内存设置

echo1024>/sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr_hugepages

echo1024>/sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages

也可通过运行 usertools/dpdk-setup.选择【18】设置所需要的大页内存

挂载大页内存：

mkdir -p /mnt/huge

mount -t hugetlbfs nodev /mnt/huge

2.2 编译环境设置

进入到安装目录：[root@localhost vm]# cd dpdk-stable-17.05.2/

export RTE_SDK=/opt/ dpdk-stable-17.05.2/ //dpdk 的安装路径

export RTE_TARGET= x86_64-native-linuxapp-gcc //编译的环境变量

2.3 编译安装 DPDK

在解路径下编译

make install T=x86_64-native-linuxapp-gcc

加载 uio 模块

cd x86_64-native-linuxapp-gcc/kmod

modprobe uio

insmod igb_uio.ko

insmod rte_kni.ko

2.4 绑定网卡到 DPDK

运行 dpdk-devbind.py 查看当前网卡状态是否已经绑定

./usertools/ dpdk-devbind.py –status

```
Network devices using DPDK-compatible driver
============================================
0000:03:00.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' drv=igb_uio unused=

Network devices using kernel driver
===================================
0000:02:00.0 'I350 Gigabit Network Connection 1521' if=eno1 drv=igb unused=igb_uio
0000:02:00.1 'I350 Gigabit Network Connection 1521' if=eno2 drv=igb unused=igb_uio *Active*
0000:09:00.0 'I350 Gigabit Network Connection 1521' if=ens4f0 drv=igb unused=igb_uio
0000:09:00.1 'I350 Gigabit Network Connection 1521' if=ens4f1 drv=igb unused=igb_uio
0000:09:00.2 'I350 Gigabit Network Connection 1521' if=ens4f2 drv=igb unused=igb_uio
0000:09:00.3 'I350 Gigabit Network Connection 1521' if=ens4f3 drv=igb unused=igb_uio

Other Network devices
=====================
0000:03:00.1 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' unused=igb_uio

Crypto devices using DPDK-compatible driver
===========================================
```

如果所需网卡在 linux 内核驱动上，先从系统内核卸载，例如

Ifconfig ens4f0 down

上图以绑定一个 10G 网卡，需要再绑定另一个

./usertools/ dpdk-devbind.py –b igb_uio 03:00.1

查看网卡状态

./usertools/ dpdk-devbind.py –status

```
Network devices using DPDK-compatible driver
============================================
0000:03:00.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' drv=igb_uio unused=
0000:03:00.1 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' drv=igb_uio unused=

Network devices using kernel driver
===================================
0000:02:00.0 'I350 Gigabit Network Connection 1521' if=eno1 drv=igb unused=igb_uio *Active*
0000:02:00.1 'I350 Gigabit Network Connection 1521' if=eno2 drv=igb unused=igb_uio
0000:09:00.0 'I350 Gigabit Network Connection 1521' if=ens4f0 drv=igb unused=igb_uio
0000:09:00.1 'I350 Gigabit Network Connection 1521' if=ens4f1 drv=igb unused=igb_uio
0000:09:00.2 'I350 Gigabit Network Connection 1521' if=ens4f2 drv=igb unused=igb_uio
0000:09:00.3 'I350 Gigabit Network Connection 1521' if=ens4f3 drv=igb unused=igb_uio

Other Network devices
=====================
<none>

Crypto devices using DPDK-compatible driver
===========================================
<none>
```

3 速率测试

3.1 测试工具

3.1.1 Testpmd

testpmd 是一个使用 DPDK 软件包分发的参考应用程序。其主要目的是在网络接口的以太网端口之间转发数据包。此外，用户还可以用 testpmd 尝试一些不同驱动程序的功能。

Testpmd 有以下几种发包模式

输入输出模式（INPUT/OUTPUT MODE）：也称为 IO 模式，是最常用的转发模式，也是 testpmd 启动时的默认模式。在 IO 模式下，CPU 内核从一个端口接收数据包，并将其发送到另一个端口，如果需要的话，一个端口可同时用于接收和发送。

收包模式（RXONLY MODE）：在收包模式下，应用程序会轮询 RX 端口的数据包，然后直接释放而不发送。他以这种方式充当数据包接收器。

发包模式（TXONLY MODE）：在发包模式下，应用程序生成 64 字节的 IP 数据包，并从 TX 端口发送出去。它不接收数据包，仅作为数据包源。

本次测试主要使用以上几种功能。

3.1.2 iperf3

iperf3 是用来测量网络最大带宽的工具。它支持测试调节各种参数比如发送持续时间，发送/接收缓存，通信协议。每次测试他都会报告网络带宽，丢包率和其他参数。本次测试用 iperf3 来测试 10G 网卡在 linux 内核 ixgbe 驱动下的最大带宽，作为 DPDK 的性能测试的对比参考。

3.1.3 pktgen-dpdk

pktgen-dpdk 是因特尔为 dpdk 开发的一个应用，它类似于 linux 原生的 pktgen，通过自己构造数据包，然后发送。Pktgen-dpdk 的功能要更强大一些，他可以通过 lua 脚本编辑自己的测试过程，同

时输出自己所关心的测试数据，比如发送、接受的数据报数量，流量带宽等等。本次测试使用 pktgen-dpdk 测试不同大小的数据包，分析数据包大小对带宽的影响。

3.2 testpmd 测试结果

3.2.1 在单服务器的两个网卡间测试

用光纤连接服务器的两个 10G 网口

运行 testpmd

./app/build/app/testpmd –l 12,13,14 –n 4 -- -i

-l选项用于指定逻辑核，核 12 用于处理命令行，核 13 和核 14 用于转发数据包。

```
[root@localhost dpdk-stable-17.05.2]# ./app/build/app/testpmd -l 12,13,14 -n 4 -- -i
EAL: Detected 40 lcore(s)
EAL: Probing VFIO support...
EAL: PCI device 0000:02:00.0 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
EAL: PCI device 0000:02:00.1 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
EAL: PCI device 0000:03:00.0 on NUMA socket 0
EAL:   probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:03:00.1 on NUMA socket 0
EAL:   probe driver: 8086:10fb net_ixgbe
EAL: PCI device 0000:09:00.0 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
EAL: PCI device 0000:09:00.1 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
EAL: PCI device 0000:09:00.2 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
EAL: PCI device 0000:09:00.3 on NUMA socket 0
EAL:   probe driver: 8086:1521 net_e1000_igb
Interactive-mode selected
USER1: create a new mbuf pool <mbuf_pool_socket_0>: n=163456, size=2176, socket=0
USER1: create a new mbuf pool <mbuf_pool_socket_1>: n=163456, size=2176, socket=1
Configuring Port 0 (socket 0)
PMD: ixgbe_dev_link_status_print():  Port 0: Link Down
Port 0: 90:E2:BA:EF:CC:90
Configuring Port 1 (socket 0)
PMD: ixgbe_dev_link_status_print():  Port 1: Link Down
Port 1: 90:E2:BA:EF:CC:91
Checking link statuses...
PMD: ixgbe_dev_link_status_print(): Port 1: Link Up - speed 10000 Mbps - full-duplex
PMD: ixgbe_dev_link_status_print(): Port 0: Link Up - speed 10000 Mbps - full-duplex
Done
testpmd> PMD: ixgbe_dev_link_status_print(): Port 1: Link Up - speed 10000 Mbps - full-duplex

Port 1: LSC event
PMD: ixgbe_dev_link_status_print(): Port 0: Link Up - speed 10000 Mbps - full-duplex

Port 0: LSC event

testpmd>
```

testpmd>提示符允许用户输入命令，这被称为实时命令行。我们可以输入 show config fwd 检查转发配置

```
testpmd> show config fwd
io packet forwarding - ports=2 - cores=1 - streams=2 - NUMA support enabled, MP over anonymous pages disabled
Logical Core 13 (socket 1) forwards packets on 2 streams:
  RX P=0/Q=0 (socket 0) -> TX P=1/Q=0 (socket 0) peer=02:00:00:00:00:01
  RX P=1/Q=0 (socket 0) -> TX P=0/Q=0 (socket 0) peer=02:00:00:00:00:00
```

显示转发模式为 IO 模式，核 13 将轮询端口 0 上的数据包，然后转发到端口 1，反之亦然。

第一步：将转发模式设置为 txonly，测试发包速率

输入命令 testpmd>set fwd txonly

testpmd>start

查看端口之间是否有数据转发，执行以下命令查看应用程序正在使用的所有端口信息

```
testpmd> show port stats all

  ######################## NIC statistics for port 0  ########################
  RX-packets: 127        RX-missed: 280637715  RX-bytes:  8128
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 280644384  TX-errors: 0          TX-bytes:  17961237312

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:     11032844
  ############################################################################

  ######################## NIC statistics for port 1  ########################
  RX-packets: 127        RX-missed: 280640203  RX-bytes:  8128
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 280646981  TX-errors: 0          TX-bytes:  17961403384

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:     11032624
  ############################################################################
```

在单核发包模式下（仅 13 核），数据传输速率约为 11.03Mpps（pps：包每秒），约合 7400Mbps。

第二步：设置发包模式为 IO 模式 set fwd io 并开始运行

```
testpmd> show port stats all

  ######################## NIC statistics for port 0  ########################
  RX-packets: 1889141140 RX-missed: 4756133379 RX-bytes:  106358631999
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 6645593835 TX-errors: 0          TX-bytes:  182212046834

  Throughput (since last show)
  Rx-pps:     11637845
  Tx-pps:     11638443
  ############################################################################

  ######################## NIC statistics for port 1  ########################
  RX-packets: 1889449252 RX-missed: 4756144096 RX-bytes:  106362360127
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 6645288393 TX-errors: 0          TX-bytes:  182209925518

  Throughput (since last show)
  Rx-pps:     11638437
  Tx-pps:     11637835
  ############################################################################
```

在单核 IO 模式下（仅 13 核），数据传输速率约为 11.64Mpps（pps：包每秒），约合 7820Mbps。

第三步：

设置双核模式：输入命令：set nbcore 2。重复进行以上测试

输入命令 testpmd>set fwd txonly

testpmd>start

查看端口之间是否有数据转发，执行以下命令查看应用程序正在使用的所有端口信息

```
testpmd> show port stats all

  ######################## NIC statistics for port 0  ########################
  RX-packets: 4026132895 RX-missed: 4849841001 RX-bytes:  175078340158
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 8876413051 TX-errors: 0          TX-bytes:  248380429465

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:     14193800
  ############################################################################

  ######################## NIC statistics for port 1  ########################
  RX-packets: 4026557343 RX-missed: 4849853984 RX-bytes:  175081861438
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 8875987719 TX-errors: 0          TX-bytes:  248379314233

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:     14193101
  ############################################################################
testpmd>
```

在双核发包模式下，数据传输速率约为 14.19Mpps（pps：包每秒），约合 9550Mbps。

设置发包模式为 IO 模式并开始运行，输入命令

set fwd io

start

```
testpmd> show port stats all

  ######################## NIC statistics for port 0  ########################
  RX-packets: 86725376   RX-missed: 214513614  RX-bytes:  5550425856
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 301249119  TX-errors: 0          TX-bytes:  19279939060

  Throughput (since last show)
  Rx-pps:     11767644
  Tx-pps:     11770087
  ############################################################################

  ######################## NIC statistics for port 1  ########################
  RX-packets: 86734132   RX-missed: 214511598  RX-bytes:  5550986240
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 301248640  TX-errors: 0          TX-bytes:  19279908472

  Throughput (since last show)
  Rx-pps:     11769813
  Tx-pps:     11767212
  ############################################################################
```

```
testpmd> stop
Telling cores to stop...
Waiting for lcores to finish...

  ---------------------- Forward statistics for port 0  ----------------------
  RX-packets: 2506195712     RX-dropped: 0             RX-total: 2506195712
  TX-packets: 2506336273     TX-dropped: 0             TX-total: 2506336273
  ----------------------------------------------------------------------------

  ---------------------- Forward statistics for port 1  ----------------------
  RX-packets: 2506336288     RX-dropped: 0             RX-total: 2506336288
  TX-packets: 2506195713     TX-dropped: 0             TX-total: 2506195713
  ----------------------------------------------------------------------------

  ++++++++++++++++ Accumulated forward statistics for all ports++++++++++++++++
  RX-packets: 5012532000     RX-dropped: 0             RX-total: 5012532000
  TX-packets: 5012531986     TX-dropped: 0             TX-total: 5012531986
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

在双核 IO 模式下，数据传输速率约为 11.77Mpps（pps：包每秒），约合 7910Mbps

3.2.2 在不同服务期间测试

用光纤连接两个服务器的 10G 网口，运行 testpmd 测试工具，将运行模式设置为双核模式，发包模式分别设置为收包模式（rxonly）和发包模式（txonly）测试，过程使用命令略。

发包方统计端口信息：

```
testpmd> show port stats all

  ######################## NIC statistics for port 0  ########################
  RX-packets: 0          RX-missed: 0          RX-bytes:  0
  RX-errors: 0
  RX-nombuf:  0
  TX-packets: 704593903  TX-errors: 0          TX-bytes:  45094005916

  Throughput (since last show)
  Rx-pps:            0
  Tx-pps:     14192633
  ############################################################################
```

```
testpmd> stop
Telling cores to stop...
Waiting for lcores to finish...

  ---------------------- Forward statistics for port 0  ----------------------
  RX-packets: 0              RX-dropped: 0             RX-total: 0
  TX-packets: 997777122      TX-dropped: 987946113     TX-total: 1985723235
  ----------------------------------------------------------------------------

  ---------------------- Forward statistics for port 1  ----------------------
  RX-packets: 0              RX-dropped: 0             RX-total: 0
  TX-packets: 0              TX-dropped: 3058946465    TX-total: 3058946465
  ----------------------------------------------------------------------------

  ++++++++++++++++ Accumulated forward statistics for all ports++++++++++++++
  RX-packets: 0              RX-dropped: 0             RX-total: 0
  TX-packets: 997777122      TX-dropped: 4046892578    TX-total: 5044669700
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

收包方统计端口信息：

```
######################## NIC statistics for port 1  ########################
RX-packets: 690968408  RX-missed: 1950       RX-bytes:  44221979648
RX-errors: 0
RX-nombuf:  0
TX-packets: 0          TX-errors: 0          TX-bytes:  0

Throughput (since last show)
Rx-pps:     14193434
Tx-pps:            0
############################################################################
testpmd> stop
```

```
testpmd> stop
Telling cores to stop...
Waiting for lcores to finish...

  ---------------------- Forward statistics for port 0  ----------------------
  RX-packets: 0              RX-dropped: 0             RX-total: 0
  RX-error: 352532
  RX-nombufs: 0
  TX-packets: 0              TX-dropped: 0             TX-total: 0
  ----------------------------------------------------------------------------

  ---------------------- Forward statistics for port 1  ----------------------
  RX-packets: 997774745      RX-dropped: 2726          RX-total: 997777471
  TX-packets: 0              TX-dropped: 0             TX-total: 0
  ----------------------------------------------------------------------------

  ++++++++++++++ Accumulated forward statistics for all ports++++++++++++++
  RX-packets: 997774745      RX-dropped: 2726          RX-total: 997777471
  TX-packets: 0              TX-dropped: 0             TX-total: 0
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

在两个服务间分别运行收包、发包模式，传输速率约为 14.19 Mpps（pps：包每秒），约合 9550Mbps。

3.3 iperf 3 对比测试

使用 dpdk 的 devbind.sh 工具，将之前绑定到 dpdk 驱动的 10G 网卡解绑，并加载到系统内核的 ixgbe 驱动下。

./devbind.sh –u 03:00.0

./devbind.sh –b ixgbe 03:00.0　(03:00.0 为本机对应网卡的 PCI 地址)

运行 iperf3

```
Accepted connection from 192.168.1.101, port 33067
[  5] local 192.168.1.100 port 9000 connected to 192.168.1.101 port 47887
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   307 MBytes  2.58 Gbits/sec  0.002 ms  45094/267643 (17%)
[  5]   1.00-2.00   sec   409 MBytes  3.43 Gbits/sec  0.006 ms  60366/356604 (17%)
[  5]   2.00-3.00   sec   427 MBytes  3.58 Gbits/sec  0.004 ms  62971/372353 (17%)
[  5]   3.00-4.00   sec   425 MBytes  3.56 Gbits/sec  0.003 ms  62260/369838 (17%)
[  5]   4.00-5.00   sec   396 MBytes  3.32 Gbits/sec  0.003 ms  57643/344279 (17%)
[  5]   5.00-6.00   sec   421 MBytes  3.54 Gbits/sec  0.007 ms  61718/366914 (17%)
[  5]   6.00-7.00   sec   422 MBytes  3.54 Gbits/sec  0.004 ms  63314/368859 (17%)
[  5]   7.00-8.00   sec   406 MBytes  3.40 Gbits/sec  0.007 ms  60562/354477 (17%)
[  5]   8.00-9.00   sec   423 MBytes  3.55 Gbits/sec  0.003 ms  63608/370015 (17%)
[  5]   9.00-10.00  sec   422 MBytes  3.54 Gbits/sec  0.003 ms  65423/370782 (18%)
[  5]  10.00-11.00  sec   421 MBytes  3.53 Gbits/sec  0.004 ms  66765/371917 (18%)
[  5]  11.00-12.00  sec   416 MBytes  3.49 Gbits/sec  0.002 ms  67204/368227 (18%)
[  5]  12.00-13.00  sec   417 MBytes  3.50 Gbits/sec  0.006 ms  67359/369540 (18%)
[  5]  13.00-14.00  sec   406 MBytes  3.40 Gbits/sec  0.004 ms  67233/361141 (19%)
[  5]  14.00-15.00  sec   385 MBytes  3.23 Gbits/sec  0.005 ms  64972/343527 (19%)
[  5]  15.00-16.00  sec   394 MBytes  3.31 Gbits/sec  0.006 ms  66840/352393 (19%)
[  5]  16.00-17.00  sec   410 MBytes  3.44 Gbits/sec  0.006 ms  71821/369053 (19%)
[  5]  17.00-18.00  sec   406 MBytes  3.41 Gbits/sec  0.005 ms  74016/368383 (20%)
[  5]  18.00-19.00  sec   404 MBytes  3.39 Gbits/sec  0.005 ms  76420/368738 (21%)
[  5]  19.00-20.00  sec   399 MBytes  3.34 Gbits/sec  0.003 ms  78177/366903 (21%)
[  5]  20.00-21.00  sec   395 MBytes  3.31 Gbits/sec  0.006 ms  79680/365761 (22%)
[  5]  21.00-22.00  sec   398 MBytes  3.34 Gbits/sec  0.006 ms  81305/369268 (22%)
[  5]  22.00-23.00  sec   369 MBytes  3.09 Gbits/sec  0.007 ms  76551/343665 (22%)
[  5]  23.00-24.00  sec   387 MBytes  3.25 Gbits/sec  0.006 ms  85628/366152 (23%)
[  5]  24.00-25.00  sec   388 MBytes  3.25 Gbits/sec  0.006 ms  87120/367975 (24%)
[  5]  25.00-26.00  sec   384 MBytes  3.22 Gbits/sec  0.004 ms  90542/368526 (25%)
[  5]  26.00-27.00  sec   351 MBytes  2.94 Gbits/sec  0.004 ms  84758/338655 (25%)
[  5]  27.00-28.00  sec   350 MBytes  2.94 Gbits/sec  0.006 ms  86543/340027 (25%)
[  5]  28.00-29.00  sec   370 MBytes  3.10 Gbits/sec  0.007 ms  97574/365348 (27%)
[  5]  29.00-30.00  sec   354 MBytes  2.97 Gbits/sec  0.008 ms  96175/352825 (27%)
[  5]  30.00-30.24  sec  82.9 MBytes  2.85 Gbits/sec  0.004 ms  22540/82592 (27%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-30.24  sec  11.7 GBytes  3.31 Gbits/sec  0.004 ms  2192182/10842380 (20%)  receiver
-----------------------------------------------------------
```

用 iperf3 测试，intel 82599ES 10G 网卡在 linux ixgbe 驱动下的最大带宽为 3.31Gbps，明显低于 dpdk 的带宽

3.4 pktgen-dpdk 测试结果

需提前安装 dpdk，使用 dpdk 的环境变量编译安装 pktgen-dpdk。安装好 pktgen-dpdk 后，环境变量、大页内存配置以及网卡绑定等流程与 testpmd 环境相同。

在两台服务器分别运行 pktgen-dpdk

./pktgen -c 1f -n 4 --proc-type auto --socket-mem 1024,1024 -- -T -P -m "[1:3].0,[2:4].1"

默认为收包模式，命令行命令如下

start 0　　0 端口开始发包

stop 0　　0 端口停止发包

set ip dst 0 192.168.1.0　设置 0 端口目的 ip 地址

set ip src 0 192.168.1.1　设置 0 端口源 ip 地址

set 0 size 1518　　设置包大小为 1518

set 0 rate 100　　设置发送速率为 100%

3.4.1 不同大小数据包传输速率

带宽（Gbps） 包长（bytes）	2 核单工	2 核双工
64	9.94	9.94/7.14
128	9.94	9.94/9.94
256	9.96	9.96/9.96
512	9.97	9.97/9.97
1024	9.98	9.98/9.98
1518	9.98	9.98/9.98

3.4.2 测试端口绑定不同数量 CPU 核的传输速率

带宽（Gbps） 包长（bytes）	1 核单工	1 核双工	2 核单工	2 核双工	4 核单工	4 核双工
64	9.93	9.63/7.23	9.94	9.94/7.14	10	9.93/7.12
128	9.93	9.81/9.81	9.94	9.94/9.94	10	10/9.65
256	9.96	9.89/9.89	9.96	9.96/9.96	10	10/10
512	9.97	9.95/9.95	9.97	9.97/9.97	10	10/10
1024	9.98	9.97/9.97	9.98	9.98/9.98	10	10/10
1518	9.98	9.98/9.98	9.98	9.98/9.98	10	10/10

4 结论

在 testpmd 测试过程中，单向收/发模式中，多核模式相对于单核模式速率有明显提升。在 IO 模式中，多核模式对速率提升幅度略小。用 testpmd 测试两个服务器间使用 DPDK 的传输速率约为 14.19Mpps，约 9550Mbps。

使用 iperf3 测试，连续 30 秒传输的最大平均带宽为 3.31Gbps

使用 pktgen-dpdk 测试，64 字节数据包的传输最大带宽约为 9.54Gbps，1518 字节数据包传输最大带宽约为 9.5Gbps。

4.1 对比结果

在 linux 驱动下利用 iperf3 测试最大平均带宽为 3.31Gbps，明显低于 dpdk 的 9.55Gbps。

4.2 绑定 CPU 核数量的影响

在 testpmd 测试过程中，分别绑定 2 个和 4 个 cpu 核，测试最大带宽不变。本例中 testpmd 测试绑定 cpu 核数为 2 个，pktgen-dpdk 绑定数量为 4 个，当数据包大小同为 64 字节时，最大带宽基本相同。所以在多核模式下，增加 cpu 核无法提升带宽。

4.3 数据包大小的影响

在使用 pktgen-dpdk 测试过程中，分别测试 64 字节数据包和 1518 字节数据包的传输速率，64 字节包传输最大带宽为 9.54Gbps，1518 字节包传输最大带宽为 9.5Gbps，差距不大。但在测试 1518 字节数据包带宽时，速率波动比 64 字节包略大，约为 200Mbps（64 字节包速率波动约为 50Mbps）。

模式 / 工具	64B 包速率	1518B 包速率	2 核收发速率	4 核收发速率
testpmd	9.55Gbps	\	9.55Gbps	9.55Gbps
iperf	3.31Gbps	\	\	\
pktgen-dpdk	9.54Gbps	9.5Gbps	9.5Gbps	9.5Gbps

5 问题总结

在使用 iperf 测试过程中出现 linux 驱动不识别网卡的现象，原因是 ixgbe 驱动只能识别 intel 网卡及光模块，本机网卡为 intel 82599ES，更换 intel 光模块后问题解决。之后出现速率达到 800Mbps 不再上升的问题，原因是 iperf2.x 版本软件限速，安装 iperf3 后问题解决。

在使用 pktgen-dpdk 测试过程中遇到安装和启动失败的问题，安装 pktgen-dpdk 需要安装 lua、libpcap、libpcap-devel、raedline、readline-devel 等几个依赖包（所依赖库以及版本视 dpdk 及 pktgen-dpdk 版本而定）。启动失败查看是否有足够的剩余大页内存页，如大页内存不足重新分配即可。