

# 性能测试

## 简介

服务器概况及测试程序性能的方法及注意事项。

## 网卡

服务器两台，配置完全一致。

`lspci | grep -i eth` 命令查看网卡详情，如下：

```
02:00.0 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.1 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.2 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
02:00.3 Ethernet controller: Broadcom Inc. and subsidiaries NetXtreme BCM5719 Gigabit Ethernet PCIe (rev 01)
12:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
12:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
13:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
13:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
```

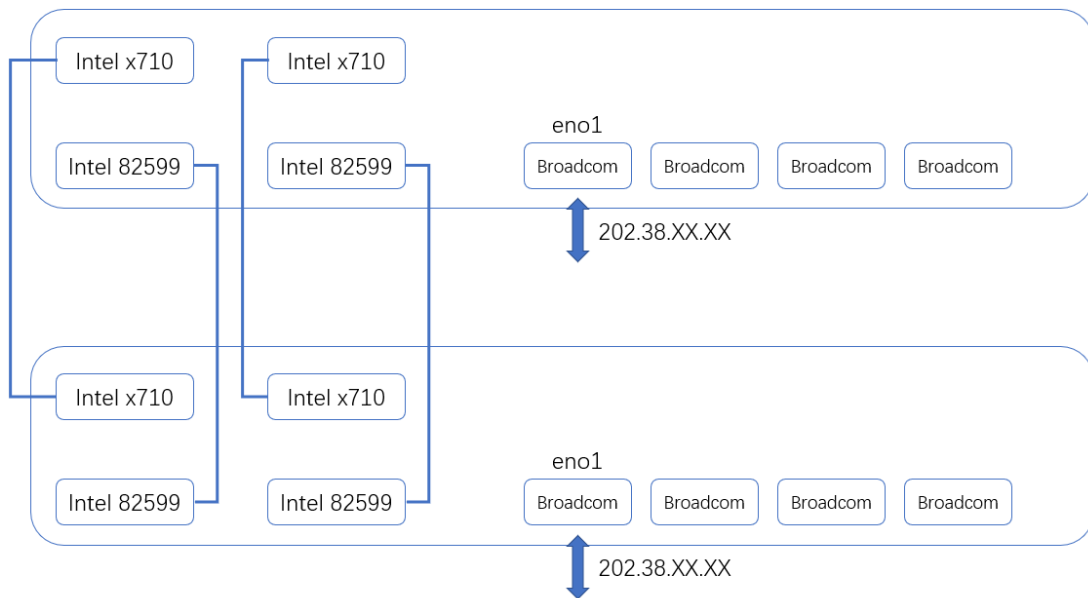
Broadcom千兆网卡×4，其中一个用于通信，另3个闲置。

Intel x710万兆网卡×2 (备用)

Intel 82599万兆网卡×2 (用于DPDK实验)

## 网卡连接

每台服务器有一个网卡(eno1)用于通信，用于实验的网卡通过光纤直连。网卡连接示意图如下：



**注意：eno1仅用于通信，请避免误操作将其断开。**

两台服务器的配置是完全一致的。可以以其中任意一台模拟DNS服务器，另一台模拟客户端。

在模拟DNS服务器的服务器上编译DPDK，并将上传你的程序。在模拟客户端的服务器上编译pktgen(用于发包)。

## 编译DPDK

主目录下有一个 `dpkg-18.11.2.tar`

```
tar xf dpkg-18.11.2.tar
```

```
cd dpkg-stable-18.11.2
```

```
make config T=x86_64-native-linuxapp-gcc
```

```
sed -ri 's,(PMD_PCAP=).*,\1y,' build/.config
```

```
make
```

## 配置DPDK

配置的过程与实验一中给出的过程类似，要注意的是，由于多用户共享了服务器资源，并且服务器是长期运行的，其中的一些步骤可以省去。一般情况下，仅需指定RTE\_SDK和RTE\_TARGET即可。

## 编译pktgen

[pktgen](#)是基于DPDK开发的一款数据包发送工具。主目录下有一个 `pktgen-3.7.2.tar.gz`

```
tar xf pktgen-3.7.2.tar.gz
```

```
cd pktgen-3.7.2
```

```
make
```

```
mv ../data.pcap . //移动查询数据到pktgen-3.7.2下
```

(注：data.pcap中包含了10000个packet，相当于 `dig @xx.xx.xx.xx -p 9000 foo.bar.com A` 命令执行10000次)

## 修改代码

有多种方法可以用于统计SimpleDNS成功完成的解析数目。直接修改其源码是较为简单的一种，对于原生SimpleDNS，在sendto()函数执行后，更新并打印统计量即可。作为对比的SimpleDNS(using DPDK)也应该有相应的功能。

## misc

- 远程操作，退格键不能正常使用时，使用CTRL+退格。
- 运行pktgen时。通过同一根光纤连接的另一个服务器上网卡，如果也处在DPDK驱动下，则只有当另一台服务器上有DPDK程序运行并使用该网卡时，pktgen控制页面上的link state才会是UP状态。
- 如何将DPDK驱动的网卡切换为内核驱动？

运行DPDK的脚本 `usertools/dpkg-setup.sh`，输入30，然后输入要切换的网卡的总线号(如0000:13:00.0)，最后输入网卡对应的内核驱动(x710对应i40e，82599对应ixgbe)。

## 测试具体步骤

- \* 测试过程中用到了新的工具——[pktgen][1]，如果以下步骤中关于pktgen的命令的使用有表述不清楚不够详细的地方，请参考相关文档。
- \* 以下将模拟客户端的服务器称为C，将模拟DNS服务器的服务器称为S。

完成上述配置后，接下来在同一环境下进行对比测试。首先测试原生SimpleDNS的性能：

在S上运行原生SimpleDNS。

在C上，当前位置在 `pktgen-3.7.2` 文件夹下时，运行下述命令启动pktgen

```
sudo -E ./app/build/pktgen -l 0-1 -n 4 -- -m "1.0" -s 0:data.pcap
```

//其中，`-l 0-1` 表示使用了2个逻辑核，`-m "1.0"` 表示1号核对应0号port

//`-s 0:data.pcap` 表示0号port以data.pcap为发包数据源（当不指定该参数时，数据源为任意内容）

//另外，0号核被规定不用于数据发送

在pktgen命令行交互界面下，

```
set 0 count 10000 //设置0号port单次发包数为10000
```

```
start 0 //开始发包
```

```
stop 0 //结束发包
```

记录下，原生SimpleDNS成功完成的解析数目。

```
quit //退出pktgen
```

然后测试SimpleDNS(using DPDK)的性能：

替换在S上运行的程序，重复上述过程即可，记录下，SimpleDNS(using DPDK)成功完成的解析数目。

## 结果

SimpleDNS(using DPDK)成功完成的解析数目是否多于原生SimpleDNS？