关于DPDK绑定的一些操作（PCIe地址以03:00.0和03:00.1为例，网卡型号为82599ES）：

1. 进入/opt目录（公司服务器DPDK基本都是放在这的）下的dpdk-16.11（项目主要用这个）或者dpdk-17.11。
2. 设置环境变量export DESTDIR=/opt/dpdk-16.11、export RTE\_TARGET=x86\_64-native-linuxapp-icc。
3. cd dpdk-16.11/x86\_64-native-linuxapp-icc/kmod ，如果没有kmod目录，需要重新编译DPDK，在x86\_64-native-linuxapp-icc目录下输入make，kmod目录下包含两个驱动文件igb\_uio.ko 、rte\_kni.ko，输入指令：1. sudo modprobe uio 2. sudo insmod igb\_uio.ko 3. sudo insmod igb\_uio.ko，使用lsmod指令查看驱动加载情况，驱动名称为igb\_uio和igb\_uio。
4. cd dpdk-16.11/tool，输入指令./dpdk-devbind.py –status查看网卡绑定情况，Network devices using DPDK-compatible driver下的为DPDK网卡，如果没有绑定网卡 ，执行指令./dpdk-devbind.py --bind=igb\_uio 03:00.0、./dpdk-devbind.py --bind=igb\_uio 03:00.1。
5. 如果想要卸载某个网卡，在tool目录下执行指令/dpdk-devbind.py –u 03:00.0。
6. 完成以上步骤应该就OK了。
7. 如果操作人员嫌弃以上步骤过于繁琐，直接cd dpdk-16.11/tool，执行./bind.sh脚本（脚本是我们自己人写的），注意进入脚本查看PCIe地址是否正确即可，然后世界清静了。以下附赠bind.sh脚本一份，预祝您玩的愉快。

#! /bin/sh

export DESTDIR=/opt/dpdk-16.11

export export RTE\_SDK=/opt/dpdk-16.11/

export RTE\_TARGET=x86\_64-native-linuxapp-icc/

cd /opt/dpdk-16.11/x86\_64-native-linuxapp-icc/

sudo modprobe uio

sudo insmod kmod/igb\_uio.ko

sudo insmod kmod/rte\_kni.ko

cd ../tools/

./dpdk-devbind.py --bind=igb\_uio 03:00.0

./dpdk-devbind.py --bind=igb\_uio 03:00.1

cd ~

exit 0