

鲲鹏 BoostKit

# 板载网卡 DPDK 用户指南

文档版本 04  
发布日期 2024-01-10



**版权所有 © 华为技术有限公司 2025。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## **商标声明**



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## **注意**

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 目 录

1 介绍.....	1
2 环境要求.....	2
3 物理机下安装运行 DPDK.....	4
3.1 安装依赖包.....	4
3.2 安装 DPDK 19.11.....	5
3.3 安装 DPDK 21.11.5.....	7
3.4 运行 DPDK.....	7
A 修改记录.....	11

# 1 介绍

---

DPDK ( Data Plane Development Kit ) 数据平面开发工具包，是Intel开发的用于网络数据包加速的开发套件。它将传统的网卡收到数据包通过内核态处理的流程转移到了用户态处理，减少了CPU处理中断和上下文切换的额外开销，能够使CPU得到更高效的利用。

## 建议的版本

DPDK版本以实际情况为准，本文以DPDK 19.11和21.11.5为例，19.11版本支持板载网卡DPDK。

# 2 环境要求

## 硬件要求

项目	说明
处理器	鲲鹏920处理器
核数	48/64
主频	2600MHz
内存大小	32GB*16
内存频率	2666MHz (Micron 2R*8)
网卡	板载TM280(4*25G)
RAID卡	LSI SAS3508

## 操作系统和软件要求

- CentOS

项目	版本
OS	CentOS Linux release 7.6.1810
Kernel	4.14.0-115.el7a.0.1.aarch64
DPDK	19.11
GCC	7.3.0

### 说明

DPDK 18.11及以上的版本需要GCC 5.0+以上版本，否则编译会报错。

- openEuler

项目	版本
OS	openEuler release 22.03 (LTS-SP2)
Kernel	5.10.0-153.12.0.92.oe2203sp2.aarch64
DPDK	21.11.5
GCC	10.3.1

# 3 物理机下安装运行 DPDK

## 3.1 安装依赖包

## 3.2 安装DPDK 19.11

## 3.3 安装DPDK 21.11.5

## 3.4 运行DPDK

## 3.1 安装依赖包

如果系统已经安装本节提到的依赖包和网卡驱动，请跳过本节。否则，请按以下步骤进行安装。

### 配置本地镜像

**步骤1** 请根据使用的操作系统获取对应的OS镜像文件并将其拷贝到每台服务器的“/root”目录下。

系统镜像下载地址：

- CentOS 7.6镜像：[CentOS-7-aarch64-Everything-1810.iso](#)
- openEuler 22.03 LTS SP2镜像：[openEuler-22.03-LTS-SP2-everything-aarch64-dvd.iso](#)

**步骤2** 创建镜像挂载目录。

```
mkdir -p /mirror
```

**步骤3** 将“/root”目录下的操作系统对应ISO文件挂载到“/mirror”目录下。

```
mount /root/Your.iso /mirror
```

#### 📖 说明

此种配置，重启后会失效。如果需要重启仍有效，请参见[步骤4](#)设置。

**步骤4** （可选）创建开启自动挂载镜像。

1. 编辑“/etc/fstab”文件。

```
vi /etc/fstab
```

2. 按“i”进入编辑模式，在文件末尾添加如下内容。

```
/root/Your.iso /mirror iso9660 loop 0 0
```



3. 按“Esc”键，输入:**wq!**，按“Enter”保存并退出编辑。

#### 步骤5 创建本地镜像源文件。

1. 进入“/etc/yum.repos.d/”目录，并备份Yum源至“bak”目录。

```
cd /etc/yum.repos.d/  
mkdir bak  
mv *.repo bak
```

2. 创建local.repo文件。

```
vim local.repo
```

3. 按“i”进入编辑模式，编辑local.repo文件。

```
[local]  
name=local.repo  
baseurl=file:///mirror  
enabled=1  
gpgcheck=0
```

4. 按“Esc”键，输入:**wq!**，按“Enter”保存并退出编辑。

5. 生效本地源。

```
yum clean all  
yum makecache  
yum list
```

----结束

## 安装 DPDK 运行依赖包

执行如下命令安装DPDK运行依赖包。

- CentOS  

```
yum -y install kernel-devel numactl-devel
```
- openEuler  

```
yum -y install kernel-devel numactl-devel  
yum groupinstall "Development Tools"  
pip3 install pyelftools  
pip3 install meson ninja
```

## 3.2 安装 DPDK 19.11

### 获取 DPDK 源码包

步骤1 获取[dpdk-19.11.tar.gz](https://dpdk.org/doc/installation_guide.html)。

步骤2 将获取的源码包上传至服务器“/home”目录下。

步骤3 解压源码包。

```
tar -xvf dpdk-19.11.tar.xz
```

----结束

### 编译 DPDK，生成依赖包

步骤1 打开板载网卡DPDK PMD选项。

1. 打开“/home/dpdk-19.11/config/common\_base”文件。

```
vim /home/dpdk-19.11/config/common_base
```

2. 按“i”进入编辑模式，将第288行的“CONFIG\_RTE\_LIBRTE\_HNS3\_PMD”参数设置为“y”。

```
#  
# Compile burst-oriented HNS3 PMD driver  
#  
CONFIG_RTE_LIBRTE_HNS3_PMD=y
```

- 按“Esc”键，输入:wq!，按“Enter”保存并退出编辑。

#### 说明

配置文件路径，根据当前DPDK实际目录为准。

### 步骤2 编译DPDK源码。

- 配置源码路径环境变量。

```
export RTE_SDK=/home/dpdk-19.11
```

- 配置环境体系结构变量。

- 获取当前环境所支持的体系结构选项。

```
cd /home/dpdk-19.11  
./usertools/dpdk-setup.sh
```

- 在回显的选项中输入“4”。

本文以鲲鹏服务器为参考，因此选择“arm64-armv8a-linuxapp-gcc”体系结构。

```
[root@localhost dpdk-next-net]# ./usertools/dpdk-setup.sh  
-----  
RTE_SDK exported as /home/dpdk-next-net  
-----  
Step 1: Select the DPDK environment to build  
-----  
[1] arm64-armada-linuxapp-gcc  
[2] arm64-armada-linux-gcc  
[3] arm64-armv8a-linuxapp-clang  
[4] arm64-armv8a-linuxapp-gcc  
[5] arm64-armv8a-linux-clang  
[6] arm64-armv8a-linux-gcc  
[7] arm64-bluefield-linuxapp-gcc  
[8] arm64-bluefield-linux-gcc  
[9] arm64-dpaa2-linuxapp-gcc  
[10] arm64-dpaa2-linux-gcc  
[11] arm64-dpaa-linuxapp-gcc  
[12] arm64-dpaa-linux-gcc
```

- 退出脚本。

- 执行如下命令进行环境变量配置。

```
export RTE_TARGET=arm64-armv8a-linuxapp-gcc
```

#### 说明

RTE\_SDK以实际版本DPDK源码目录为准，RTE\_TARGET以实际CPU体系结构为准。

#### 须知

DPDK 18.11.1及以上版本，需要保证GCC版本在5.0以上，否则编译会报错。

- 在DPDK源码目录“/home/dpdk-next-net”下，编译DPDK。  
该命令可以充分发挥多核特性，提高编译的速度。

```
make install T=$RTE_TARGET -j 16
```

----结束

## 3.3 安装 DPDK 21.11.5

步骤1 获取[dpdk-21.11.5.tar.xz](#)。

步骤2 将获取的源码包上传至服务器“/home”目录下。

步骤3 解压源码包。

```
tar -xvf dpdk-21.11.5.tar.xz  
cd dpdk-stable-21.11.5/
```

步骤4 执行以下命令编译并安装DPDK。

```
meson setup build  
cd build  
ninja  
ninja install  
ldconfig
```

----结束

## 3.4 运行 DPDK

### 配置大页内存

使用大页可以提升DPDK转发性能。最大支持1G大页，但是4.14内核不支持1G大页，若需要使用1G大页则需要升级内核。

内核为4.14时默认支持2M和512M大页。本文以内核配置512M大页为例，2M大页的配置方法类似，本文不再赘述。

步骤1 查看当前环境支持的NUMA Node个数，根据NUMA Node数为每个Node预留大页内存。

```
numactl -H
```

```
[root@localhost dpdk-next-net]# numactl -H  
available: 4 nodes (0-3)  
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
node 0 size: 130047 MB  
node 0 free: 81091 MB  
node 1 cpus: 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  
node 1 size: 130939 MB  
node 1 free: 96255 MB  
node 2 cpus: 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47  
node 2 size: 130939 MB  
node 2 free: 103199 MB  
node 3 cpus: 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63  
node 3 size: 130936 MB  
node 3 free: 103246 MB  
node distances:  
node    0    1    2    3  
  0:   10   16   32   33  
  1:   16   10   25   32  
  2:   32   25   10   16  
  3:   33   32   16   10
```

**步骤2** 查看系统支持的大页类型。

```
ll /sys/kernel/mm/hugepages/
```

若回显信息中显示有**hugepages-524288kB**字样则表示支持512M大页。

若回显信息中显示有**hugepages-1048576kB**字样则表示支持1G大页。

可以根据实际情况进行选择使用，默认优先选择最大的大页类型。

**步骤3** 预留大页内存。

- 配置512M大页。
  - a. 配置大页个数。

```
echo 50 > /sys/devices/system/node/node0/hugepages/hugepages-524288kB/nr_hugepages
echo 50 > /sys/devices/system/node/node1/hugepages/hugepages-524288kB/nr_hugepages
echo 50 > /sys/devices/system/node/node2/hugepages/hugepages-524288kB/nr_hugepages
echo 50 > /sys/devices/system/node/node3/hugepages/hugepages-524288kB/nr_hugepages
```
  - b. 挂载大页。

```
mkdir -p /mnt/huge_512mb
mount -t hugetlbfs none /mnt/huge_512mb -o pagesize=512MB
```
- （可选，需系统支持）配置1G大页。
  - a. 打开“/etc/default/grub”文件。

```
vi /etc/default/grub
```
  - b. 修改系统启动项，预留大页，预留16个1G大页，总共16G。在GRUB\_CMDLINE\_LINUX行最后加上：

```
default_hugepagesz=1G hugepagesz=1G hugepages=16
```
  - c. 生成系统grub并重启系统使配置生效。

```
grub2-mkconfig -o /boot/efi/EFI/openeuler/grub.cfg
```
  - d. 确认是否配置成功。

```
cat /proc/meminfo | grep -i huge
```

若回显信息中显示已配置的大页数量为**16**，则预留大页配置成功。
  - e. 挂载大页。

```
mkdir -p /mnt/huge_1gb
mount -t hugetlbfs none /mnt/huge_1gb -o pagesize=1GB
```

----结束

## 接管内核态网口

**步骤1** 加载内核驱动模块（使用vfio需要BIOS开启SMMU）。

```
modprobe vfio
modprobe vfio-pci
```

**步骤2** 绑定被测网口用户态，用于DPDK转发。首先获取网口PCI地址，可以使用DPDK源码“usertools/dpdk-devbind.py”脚本进行查看，以及进行接口绑定工作。

1. 查看待绑定接口信息。

```
dpdk-devbind.py -s
```

```
[root@localhost dpdk-next-net]# ./usertools/dpdk-devbind.py -s
Network devices using DPDK-compatible driver
=====
0000:05:00.0 'Hi1822 Family (4*25GE) 1822' drv=igb_uio unused=hinic,vfio-pci
0000:7d:00.1 'HNS GE/10GE/25GE Network Controller a221' drv=igb_uio unused=hns3,vfio-pci
Network devices using kernel driver
=====
0000:06:00.0 'Hi1822 Family (4*25GE) 1822' if=enp6s0 drv=hinic unused=igb_uio,vfio-pci
0000:07:00.0 'Hi1822 Family (4*25GE) 1822' if=enp7s0 drv=hinic unused=igb_uio,vfio-pci
0000:08:00.0 'Hi1822 Family (4*25GE) 1822' if=enp8s0 drv=hinic unused=igb_uio,vfio-pci
0000:09:00.0 'Hi1822 Family (4*25GE) 1822' if=enp9s0 drv=hinic unused=igb_uio,vfio-pci
0000:7d:00.0 'HNS GE/10GE/25GE RDMA Network Controller a222' if=enp125s0f0 drv=hns3 unused=hns3
0000:7d:00.2 'HNS GE/10GE/25GE RDMA Network Controller a222' if=enp125s0f2 drv=hns3 unused=hns3
0000:bd:00.0 'HNS GE/10GE/25GE RDMA Network Controller a222' if=enp189s0f0 drv=hns3 unused=hns3
0000:bd:00.1 'HNS GE/10GE/25GE Network Controller a221' if=enp189s0f1 drv=hns3 unused=igb_uio,vfio-pci
0000:bd:00.2 'HNS GE/10GE/25GE RDMA Network Controller a222' if=enp189s0f2 drv=hns3 unused=hns3
0000:bd:00.3 'HNS GE/10GE/25GE Network Controller a221' if=enp189s0f3 drv=hns3 unused=igb_uio,vfio-pci
```

2. 查看端口状态，接管链路状态up的端口。  
ethtool enp125s0f1

```
[root@localhost usertools]# ethtool enp125s0f1
Settings for enp125s0f1:
    Supported ports: [ FIBRE ]
    Supported link modes:   25000baseSR/Full
                           1000baseX/Full
                           10000baseSR/Full
    Supported pause frame use: Symmetric
    Supports auto-negotiation: No
    Supported FEC modes: Not reported
    Advertised link modes:  Not reported
    Advertised pause frame use: No
    Advertised auto-negotiation: No
    Advertised FEC modes: Not reported
    Speed: 10000Mb/s
    Duplex: Full
    Port: FIBRE
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: off
    Link detected: yes
```

3. 绑定用于DPDK转发的网口，此处以绑定0000:7d:00.1网口为例，使用时以查询到的待绑定的网口为准。

```
dpdk-devbind.py --bind=vfio-pci 0000:7d:00.1
```

查看端口是否成功绑定：

```
dpdk-devbind.py -s
```

```
[root@localhost dpdk-next-net]# ./usertools/dpdk-devbind.py -s
Network devices using DPDK-compatible driver
=====
0000:7d:00.1 'HNS GE/10GE/25GE Network Controller a221' drv=vfio-pci unused=hns3
```

#### 说明

端口处于up状态时进行绑定会报错，此时应执行ifconfig xxx down命令，再进行绑定。

----结束

## 运行 DPDK

完成上述配置，即可开始运行DPDK相关测试程序。本文以编译DPDK生成的标准测试程序testpmd为例，根据环境实际信息运行以下命令。

```
testpmd -c 0xf-n 4 -w $PCI -- --rxd=$depth --txd=$depth --rxq=$qnum --txq=$qnum --nb-cores=$cores -i
```

说明

testpmd为DPDK 19.11版本中的测试命令，DPDK 21.11.5版本中testpmd命令为dpdk-testpmd。

参数	说明
-c	指定参与转发的CPU核，后跟参加转发核的掩码。
-n	指定内存通道个数。
-w	指定参与转发的网口PCI，即3.2 安装DPDK 19.11章节中绑定的网口PCI地址。
--rxd/--txd	指定参与转发的网口队列深度。
--rxq/--txq	指定参与转发的网口队列个数。
--nb-cores	指定实际参与转发的CPU核数。
-i	以交互模式启动程序。

# A 修改记录

发布日期	修订记录
2024-01-10	第四次正式发布。 新增DPDK 21.11.5适配openEuler 22.03 LTS SP2操作系统。
2023-09-27	第三次正式发布。 优化文档内容和结构。
2023-09-12	第二次正式发布。 <ul style="list-style-type: none"><li>修改配置1G大页的文件名称。</li><li>增加确认是否支持配置1G大页的内容。</li></ul>
2023-07-07	第一次正式发布。