kvm虚拟机中配置RDMA（以太网连接）

一、配置SR-IOV

1. 物理机配置

1）服务器通过以太网交换机连接

2）安装kvm

```
sudo apt-get install kvm
sudo apt-get install virt-manager libvirt libvirt-python python-virtinst
```

3）BIOS上开启SR-IOV

实验室服务器N2上开启SR-IOV可参考此链接，在这里还可以设置网卡端口对应的VF数量，后续实验可能会用到

HowTo Set Dell PowerEdge R730 BIOS parameters to support SR-IOV (nvidia.com)

其中会遇到devices中看不到mlx网卡的情况，我参考这个链接解决的

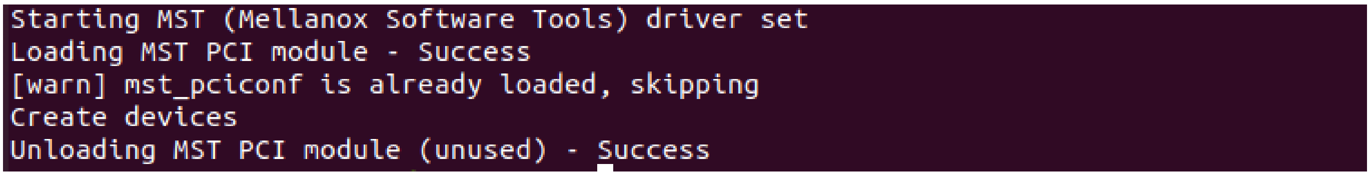
工程师笔记 | 服务器OS升级找不到网卡怎么办？ - 腾讯云开发者社区-腾讯云 (tencent.com)

4）在grub中开启**intel_iommu=on**和**iommu=pt**

5）安装MLNX_OFED驱动

6）运行MFT

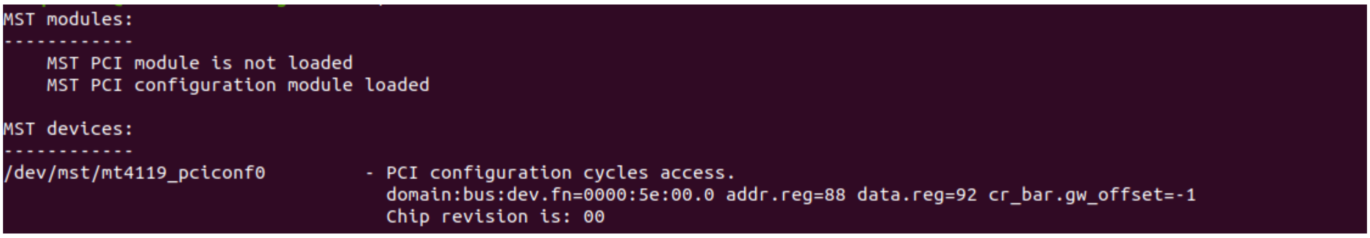
```
sudo mst start
```



```
Starting MST (Mellanox Software Tools) driver set
Loading MST PCI module - Success
[warn] mst_pciconf is already loaded, skipping
Create devices
Unloading MST PCI module (unused) - Success
```

7）找到网卡设备在哪个PCI插槽

```
sudo mst status
```



```
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded

MST devices:
------------
/dev/mst/mt4119_pciconf0         - PCI configuration cycles access.
                                   domain:bus:dev.fn=0000:5e:00.0 addr.reg=88 data.reg=92 cr_bar.gw_offset=-1
                                   Chip revision is: 00
```

此处是`/dev/mst/mt4119_pciconf0`

8）设置网卡开启SR-IOV，并设定需要的VF数量

```
sudo mlxconfig -d /dev/mst/mt4119_pciconf0 q # 查询参数设置
sudo mlxconfig -d /dev/mst/mt4119_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=4 #设置参数
```

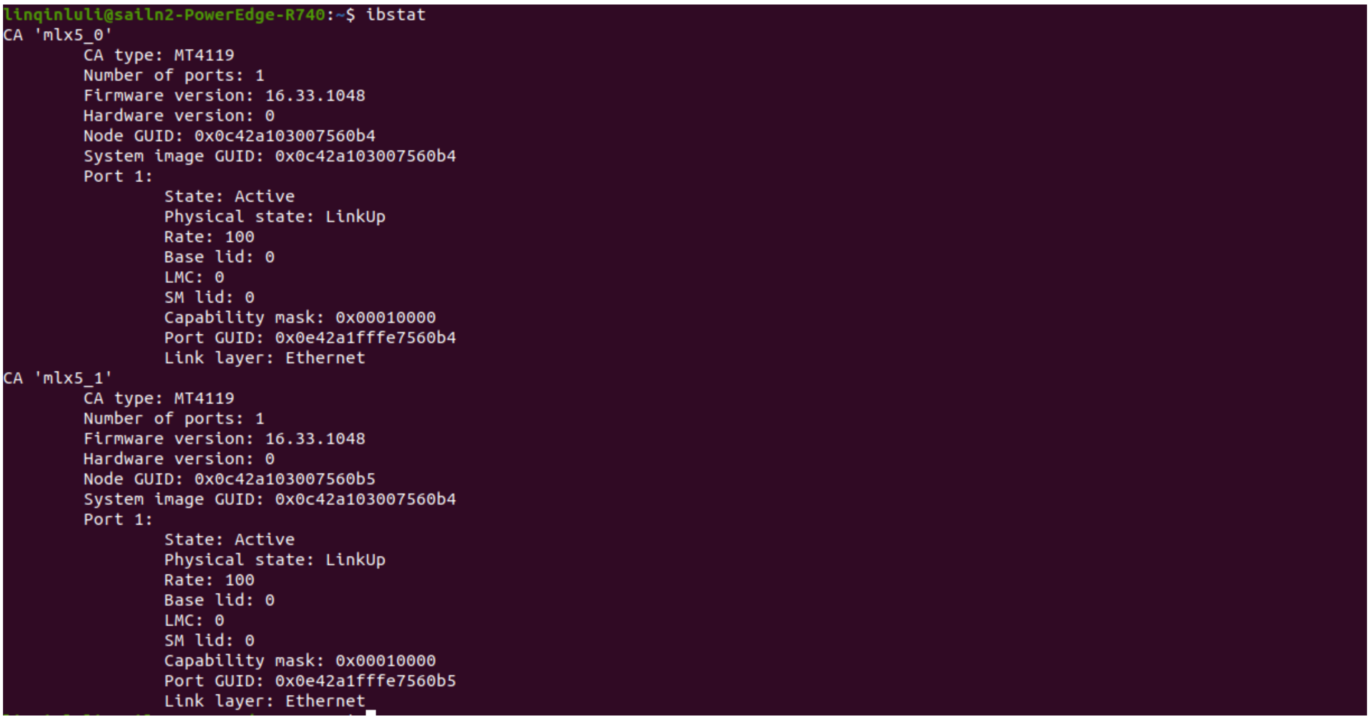
- SRIOV_EN=1
- NUM_OF_VFS=4

保证这两个参数设置成功，设置完成需要重启物理机

2. MLNX_OFED驱动配置SR-IOV

1）找到mlx网卡设备对应网卡号

```
ibstat # 查询端口状态
```



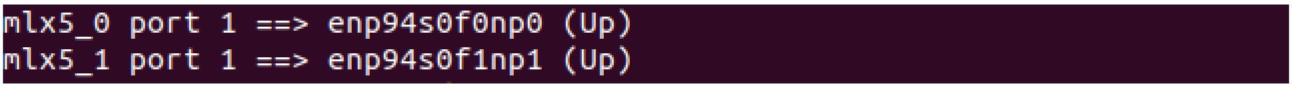
```
linqinluli@sailn2-PowerEdge-R740:~$ ibstat
CA 'mlx5_0'
        CA type: MT4119
        Number of ports: 1
        Firmware version: 16.33.1048
        Hardware version: 0
        Node GUID: 0x0c42a103007560b4
        System image GUID: 0x0c42a103007560b4
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 100
                Base lid: 0
                LMC: 0
                SM lid: 0
                Capability mask: 0x00010000
                Port GUID: 0x0e42a1fffe7560b4
                Link layer: Ethernet
CA 'mlx5_1'
        CA type: MT4119
        Number of ports: 1
        Firmware version: 16.33.1048
        Hardware version: 0
        Node GUID: 0x0c42a103007560b5
        System image GUID: 0x0c42a103007560b4
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 100
                Base lid: 0
                LMC: 0
                SM lid: 0
                Capability mask: 0x00010000
                Port GUID: 0x0e42a1fffe7560b5
                Link layer: Ethernet
```

这里两个端口mlx5_0和mlx_1，需要使用哪个端口需要保证那个设备参数

State: Active

Physical state: LinkUp

```
ibdev2netdev # 查询端口和网卡绑定状态
```



```
mlx5_0 port 1 ==> enp94s0f0np0 (Up)
mlx5_1 port 1 ==> enp94s0f1np1 (Up)
```

mlx5_0 port 1 ==> enp94s0f0np0 (Up) mlx5_1 port 1 ==> enp94s0f1np1 (Up)

2）获取固件所允许的VFs总数

```
 cat /sys/class/net/enp94s0f0np0/device/sriov_totalvfs
```

结果为4，即之前配置的**NUM_OF_VFS=4**

如果没有看见这个参数，则表示之前**intel_iommu=on**没有配置成功

3）配置VF数量

有三种方式配置

```
sudo sh -c "echo 4 > /sys/class/infiniband/mlx5_0/device/mlx5_num_vfs""
sudo cat /sys/class/infiniband/mlx5_0/device/mlx5_num_vfs

sudo sh -c "echo 4 > /sys/class/net/enp94s0f0np0/device/sriov_numvfs"
sudo cat /sys/class/net/enp94s0f0np0/device/sriov_numvfs

sudo sh -c "echo 4 > /sys/class/net/enp94s0f0np0/device/mlx5_num_vfs"
sudo cat /sys/class/net/enp94s0f0np0/device/mlx5_num_vfs
```

任意方式配置成功即可，配置一个参数，三个参数的查询结果都是配置结果，如果**sriov_numvfs**参数不在，需要检查**intel_iommu**是否加入到grub文件中

！！！这一步由于需要先配置自动探测VF，所以建议依次执行以下命令！！！

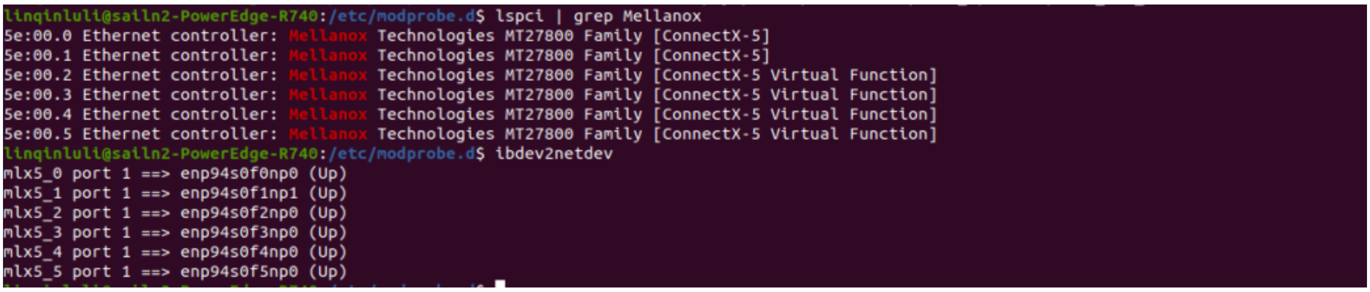
```
sudo sh -c "echo 0 > /sys/class/infiniband/mlx5_0/device/mlx5_num_vfs"
# 关掉sr-iov
sudo sh -c "echo 1 > /sys/module/mlx5_core/parameters/probe_vf"
# 开启驱动自动探测VF
sudo sh -c "echo 4 > /sys/class/infiniband/mlx5_0/device/mlx5_num_vfs"
# 开启sr-iov
```

注意！！！

1. VFs数量的参数配置不是永久存在，服务器重启之后需要重新配置
2. ~~由于实验室使用的是mlx5的网卡，**配置VF之前需要配置驱动自动探测VF，这里有个tudo，一直没弄好！！！！！！**，可以参考HowTo Configure and Probe VFs on mlx5 Drivers (nvidia.com)。我照着这篇文章配置了好几遍，还是没成功PCI里已经可以看到VF了，但是驱动还是没有找到VF，下一步尝试在网卡配置之前配置自动探测VF，重新走一遍流程~~

4）检查配置情况

```
lspci -D | grep Mellanox # PCI状态
sudo ibdev2netdev -v     # 驱动绑定状态
```



```
linqinluli@sailn2-PowerEdge-R740:/etc/modprobe.d$ lspci | grep Mellanox
5e:00.0 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
5e:00.1 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5]
5e:00.2 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
5e:00.3 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
5e:00.4 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
5e:00.5 Ethernet controller: Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
linqinluli@sailn2-PowerEdge-R740:/etc/modprobe.d$ ibdev2netdev
mlx5_0 port 1 ==> enp94s0f0np0 (Up)
mlx5_1 port 1 ==> enp94s0f1np1 (Up)
mlx5_2 port 1 ==> enp94s0f2np0 (Up)
mlx5_3 port 1 ==> enp94s0f3np0 (Up)
mlx5_4 port 1 ==> enp94s0f4np0 (Up)
mlx5_5 port 1 ==> enp94s0f5np0 (Up)
```

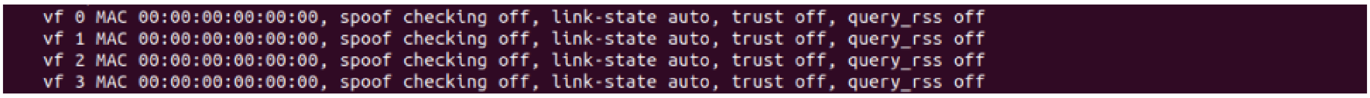
这里几个VF的基本信息如下：

PCI Function	VF num			
0000:5e:00.2	0	enp94s0f2np0		
0000:5e:00.3	1	enp94s0f3np0		
0000:5e:00.4	2	enp94s0f4np0		
0000:5e:00.5	3	enp94s0f5np0		

5）为每个VF设置MAC地址

运行

```
ip link show
```



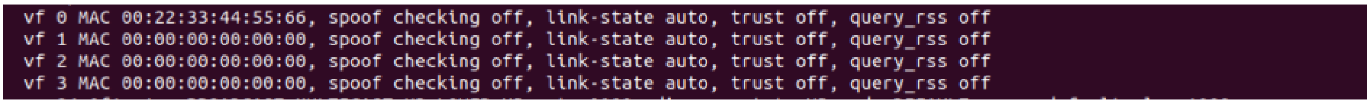
```
    vf 0 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
    vf 1 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
    vf 2 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
    vf 3 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
```

看到几个vf都没有分配MAC地址

运行以下命令分配MAC地址

```
sudo sh -c "echo 0000:5e:00.2 > /sys/bus/pci/drivers/mlx5_core/unbind"

sudo ip link set enp94s0f0np0 vf 0 mac 00:22:33:44:55:66

sudo sh -c "echo 0000:5e:00.2 > /sys/bus/pci/drivers/mlx5_core/bind"
```

之后运行 `ip link show`，结果如下，可以看到MAC地址已经配置完成，之后使用此VF0进行实验

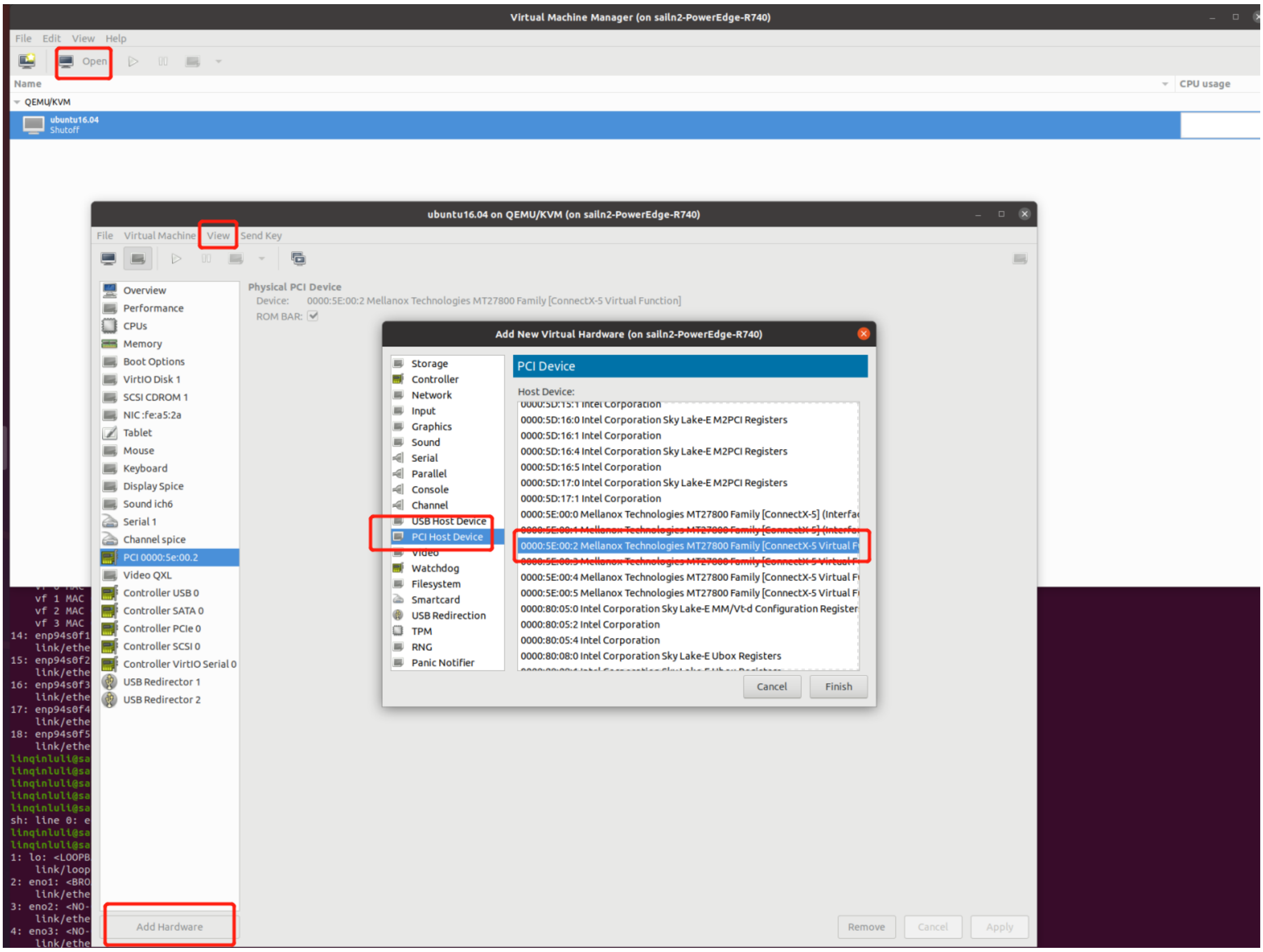


```
    vf 0 MAC 00:22:33:44:55:66, spoof checking off, link-state auto, trust off, query_rss off
    vf 1 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
    vf 2 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
    vf 3 MAC 00:00:00:00:00:00, spoof checking off, link-state auto, trust off, query_rss off
```

3.虚拟机配置

1）为虚拟机添加PCI设备

在添加之前关闭虚拟机



Physical PCI Device
Device: 0000:5E:00:2 Mellanox Technologies MT27800 Family [ConnectX-5 Virtual Function]
ROM BAR: ☑

Host Device:

```
0000:5D:15:1 Intel Corporation
0000:5D:16:0 Intel Corporation Sky Lake-E M2PCI Registers
0000:5D:16:1 Intel Corporation
0000:5D:16:4 Intel Corporation Sky Lake-E M2PCI Registers
0000:5D:16:5 Intel Corporation
0000:5D:17:0 Intel Corporation Sky Lake-E M2PCI Registers
0000:5D:17:1 Intel Corporation
0000:5E:00:0 Mellanox Technologies MT27800 Family [ConnectX-5] (Interfa
0000:5E:00:1 Mellanox Technologies MT27800 Family [ConnectX-5] (Interfa
0000:5E:00:2 Mellanox Technologies MT27800 Family [ConnectX-5 Virtual F
0000:5E:00:3 Mellanox Technologies MT27800 Family [ConnectX-5 Virtual F
0000:5E:00:4 Mellanox Technologies MT27800 Family [ConnectX-5 Virtual F
0000:5E:00:5 Mellanox Technologies MT27800 Family [ConnectX-5 Virtual F
0000:80:05:0 Intel Corporation Sky Lake-E MM/Vt-d Configuration Register
0000:80:05:2 Intel Corporation
0000:80:05:4 Intel Corporation
0000:80:08:0 Intel Corporation Sky Lake-E Ubox Registers
```

2）为虚拟机安装MLNX_OFED，可参考

Mellanox网卡OFED驱动安装 - 简书 (jianshu.com)

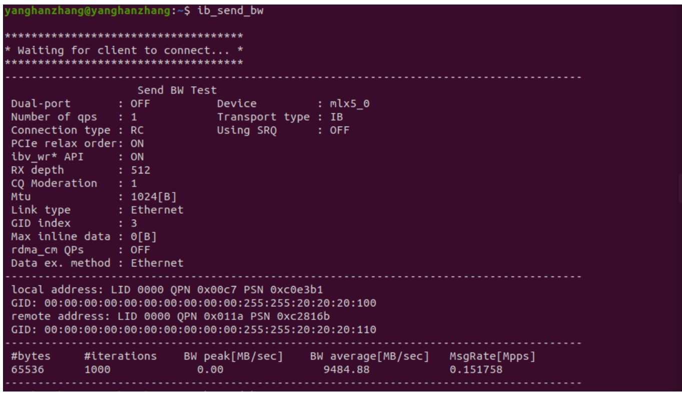
常用指令

```
sudo su #进入root权限用户
sudo mount -o loop /root/MLNX_OFED_LINUX-5.4-3.5.8.0-ubuntu16.04-x86_64.iso /mnt/iso/
#挂载镜像
sudo ./mlnxofedinstall #运行安装程序
/etc/init.d/openibd restart #重启驱动
/usr/sbin/ofed_uninstall.sh #卸载驱动
```

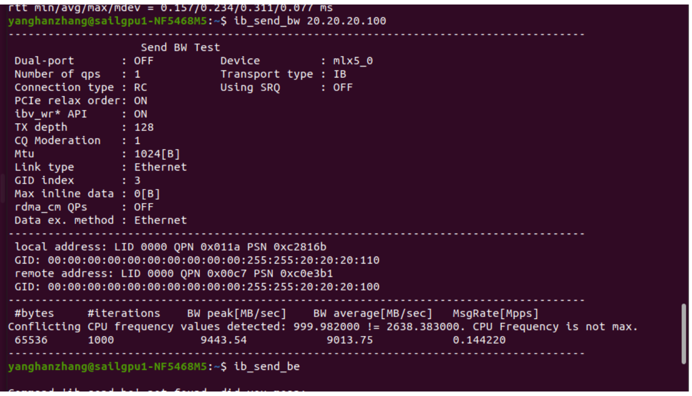
3）为虚拟机配置IP地址

```
ifconfig [网卡名] [ip] up
```

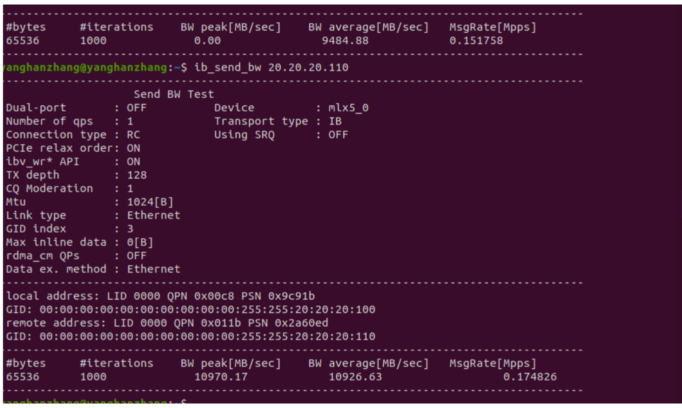
4）测试RDMA通信情况，至此可以看到kvm虚拟机中RDMA通信成功，可以进行后续实验



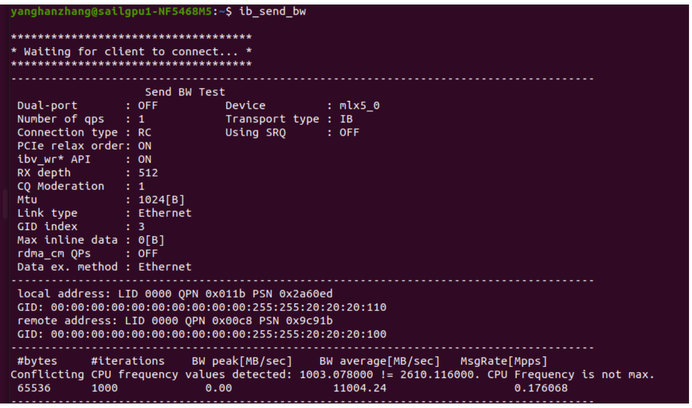
```
yanghanzhang@yanghanzhang:~$ ib_send_bw

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    Send BW Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 RX depth        : 512
 CQ Moderation   : 1
 Mtu             : 1024[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x00c7 PSN 0xc0e3b1
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:100
 remote address: LID 0000 QPN 0x011a PSN 0xc2816b
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:110
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
 65536      1000             0.00               9484.88            0.151758
---------------------------------------------------------------------------------------
```



```
rtt min/avg/max/mdev = 0.157/0.234/0.311/0.077 ms
yanghanzhang@sailgpu1-NF5468M5:~$ ib_send_bw 20.20.20.100
---------------------------------------------------------------------------------------
                    Send BW Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 TX depth        : 128
 CQ Moderation   : 1
 Mtu             : 1024[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x011a PSN 0xc2816b
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:110
 remote address: LID 0000 QPN 0x00c7 PSN 0xc0e3b1
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:100
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
Conflicting CPU frequency values detected: 999.982000 != 2638.383000. CPU Frequency is not max.
 65536      1000             9443.54            9013.75            0.144220
---------------------------------------------------------------------------------------
yanghanzhang@sailgpu1-NF5468M5:~$ ib_send_be
```



```
---------------------------------------------------------------------------------------
#bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
65536      1000             0.00               9484.88            0.151758
---------------------------------------------------------------------------------------
yanghanzhang@yanghanzhang:~$ ib_send_bw 20.20.20.110
---------------------------------------------------------------------------------------
                    Send BW Test
Dual-port       : OFF          Device         : mlx5_0
Number of qps   : 1            Transport type : IB
Connection type : RC           Using SRQ      : OFF
PCIe relax order: ON
ibv_wr* API     : ON
TX depth        : 128
CQ Moderation   : 1
Mtu             : 1024[B]
Link type       : Ethernet
GID index       : 3
Max inline data : 0[B]
rdma_cm QPs     : OFF
Data ex. method : Ethernet
---------------------------------------------------------------------------------------
local address: LID 0000 QPN 0x00c8 PSN 0x9c91b
GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:100
remote address: LID 0000 QPN 0x011b PSN 0x2a60ed
GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:110
---------------------------------------------------------------------------------------
#bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
65536      1000             10970.17           10926.63                   0.174826
---------------------------------------------------------------------------------------
```



```
yanghanzhang@sailgpu1-NF5468M5:~$ ib_send_bw

************************************
* Waiting for client to connect... *
************************************
---------------------------------------------------------------------------------------
                    Send BW Test
 Dual-port       : OFF          Device         : mlx5_0
 Number of qps   : 1            Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON
 ibv_wr* API     : ON
 RX depth        : 512
 CQ Moderation   : 1
 Mtu             : 1024[B]
 Link type       : Ethernet
 GID index       : 3
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x011b PSN 0x2a60ed
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:110
 remote address: LID 0000 QPN 0x00c8 PSN 0x9c91b
 GID: 00:00:00:00:00:00:00:00:00:00:255:255:20:20:20:100
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
Conflicting CPU frequency values detected: 1003.078000 != 2610.116000. CPU Frequency is not max.
 65536      1000             0.00               11004.24                   0.176068
---------------------------------------------------------------------------------------
```