如何配置BroadCOM网卡的SR-IOV功能

Linux系统安装 Linux系统调优 BIOS BIOS参数调优 **田思文** 2018-02-22 发表

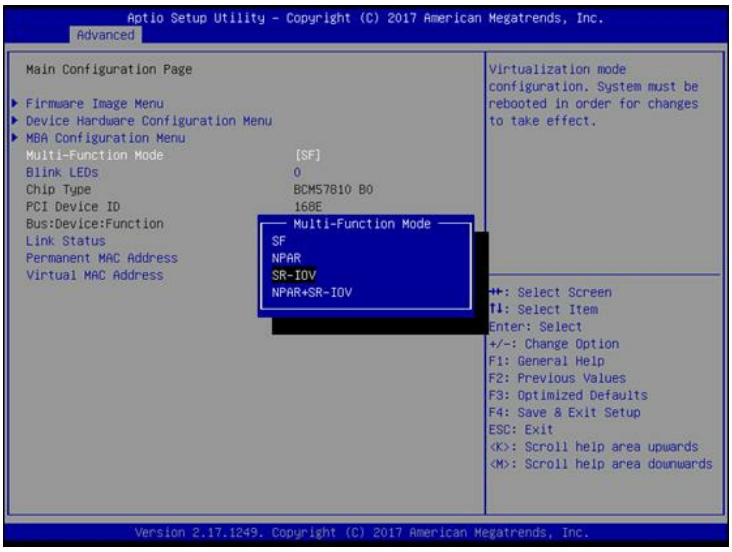
用户咨询如何配置Broadcom网卡的SR-IOV的功能，关于该配置网上相关手册很有限，解释的也不详细，对此笔者对现有设备进行实验和测试，为开启SR-IOV功能的操作做一个介绍。

配置过程中会有告警进行，需要适当的进行配置的调整，如
1. 在dmesg中可能会有如下的报错信息
bnx2x 0000:03:00.0: not enough MMIO resources for SR-IOV
2. 查看到VF网卡的MAC地址为00:00:00:00:00:00

关于bnx2x 0000:03:00.0: not enough MMIO resources for SR-IOV的报错， 主要是BIOS的问题，BIOS没有为VF提供足够的MMIO space，可以在系统的kernel中增加参数解决

本案例中使用的服务器测试环境如下：
服务器：H3C R390X G2
操作系统：RHEL7.3
SR-IOV网卡型号：Brocadcom 530FLB （BCM57810芯片）

1.　首先在BIOS中开启网卡的SR-IOV的支持
服务器开机自检按ESC或DEL进入BIOS Setup，点击Advanced -> 选中530FLR网卡。 默认Multi-Function Mode为SF，这里改成SR-IOV



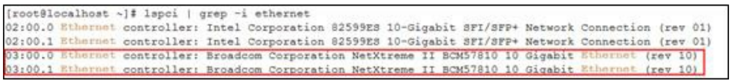
```
Aptio Setup Utility - Copyright (C) 2017 American Megatrends, Inc.
Advanced

Main Configuration Page                               Virtualization mode
                                                      configuration. System must be
▶ Firmware Image Menu                                 rebooted in order for changes
▶ Device Hardware Configuration Menu                  to take effect.
▶ MBA Configuration Menu
Multi-Function Mode                 [SF]
Blink LEDs                          0
Chip Type                           BCM57810 B0
PCI Device ID                       168E
Bus:Device:Function        ┌─ Multi-Function Mode ─┐
Link Status                │ SF                    │
Permanent MAC Address      │ NPAR                  │
Virtual MAC Address        │ SR-IOV                │
                           │ NPAR+SR-IOV           │
                           └───────────────────────┘
                                                      ++: Select Screen
                                                      ↑↓: Select Item
                                                      Enter: Select
                                                      +/-: Change Option
                                                      F1: General Help
                                                      F2: Previous Values
                                                      F3: Optimized Defaults
                                                      F4: Save & Exit Setup
                                                      ESC: Exit
                                                      <K>: Scroll help area upwards
                                                      <M>: Scroll help area downwards

Version 2.17.1249. Copyright (C) 2017 American Megatrends, Inc.
```

2.　操作系统中开启IOMMU支持
执行dmesg | grep -i iommu看操作系统是否开启了IOMMU支持，如果没开启，则编辑如下
# vi /etc/default/grub
...
GRUB_CMDLINE_LINUX="nofb splash=quiet cOnsole=tty0 intel_iommu=on
...
重新生成grub配置文件
#grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
#reboot
重启后查看iommu启动情况
#dmesg | grep -i iommu

```
[    0.000000] DMAR: IOMMU enabled
```

则表示开启成功

3.　系统中查看网卡，我们使用的是BCM57810芯片的网卡（Broadcom）
查看网卡信息



```
[root@localhost ~]# lspci | grep -i ethernet
02:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
02:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
03:00.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet (rev 10)
03:00.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet (rev 10)
```

4.　开启网卡的VF端口

注意：首先要确保端口是up状态

#ifup ens9f0

查看sriov的端口数量

# cat /sys/class/net/ens9f0/device/sriov_numvfs

0

如果返回结果是0，表示没有VF接口

5.　开启VF端口

# echo 8 > /sys/class/net/ens9f0/device/sriov_numvfs

备注：enable VF时，可能会报错如下：

[641.704649] bnx2x 0000:03:00.0: not enough MMIO resources for SR-IOV

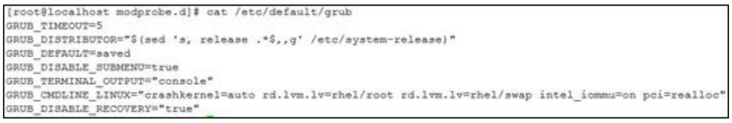
[641.704656] [bnx2x_enable_sriov:2514(ens9f0)]pci_enable_sriov failed with -12

上面的报错通常是BIOS issue，可能是BIOS不支持 (The BIOS is not providing enough MMIO space for VFs)

参考文档：https://access.redhat.com/solutions/37376

解决办法: 在kernel中再加入一个参数pci=realloc

修改/etc/default/grub，在之前的iommu选项后，加入该参数



```
[root@localhost modprobe.d]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap intel_iommu=on pci=realloc"
GRUB_DISABLE_RECOVERY="true"
```

重新生成grub

#grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg

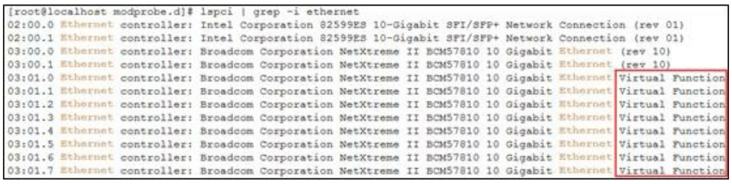
#reboot

6.　检查VF开启情况

经过刚刚的设置之后，再次执行下面的命令后，即可查看到VF port

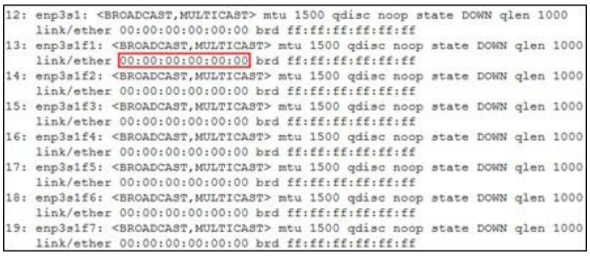
# echo 8 > /sys/class/net/ens9f0/device/sriov_numvfs

lspci | grep –i ethernet



```
[root@localhost modprobe.d]# lspci | grep -i ethernet
02:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
02:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
03:00.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet (rev 10)
03:00.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet (rev 10)
03:01.0 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.1 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.2 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.3 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.4 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.5 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.6 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
03:01.7 Ethernet controller: Broadcom Corporation NetXtreme II BCM57810 10 Gigabit Ethernet Virtual Function
```

# ip addr show



```
12: enp3s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
13: enp3s1f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
14: enp3s1f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
15: enp3s1f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
16: enp3s1f4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
17: enp3s1f5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
18: enp3s1f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
19: enp3s1f7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

但是所有的mac地址都是00:00:00:00:00:00

根据Broadcom bnx2x driver的readme描述，这属于正常情况

https://downloads.hpe.com/pub/softlib2/software1/pubsw-linux/p1050551721/v140545/README

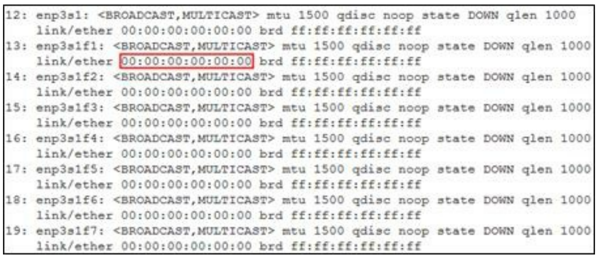
Known issues/Limitations/Caveats

-The bnx2x driver now assigns all zeroes as the MAC address for SR-IOV virtual functions. Users need to manually configure valid MAC addresses for virtual functions using iproute2 or ifconfig methods

7.　手动设置VF的MAC地址

# ip link show

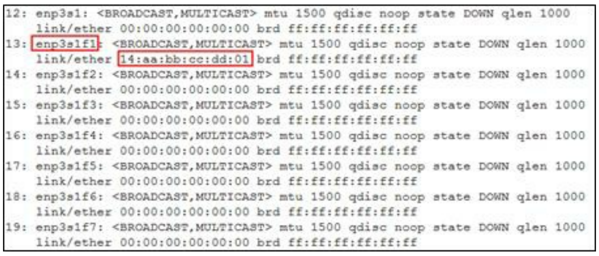
先查看MAC地址



```
12: enp3s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
13: enp3s1f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
14: enp3s1f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
15: enp3s1f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
16: enp3s1f4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
17: enp3s1f5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
18: enp3s1f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
19: enp3s1f7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

手动设置MAC地址

# ip link set enp3s1f1 addr 14:aa:bb:cc:dd:01

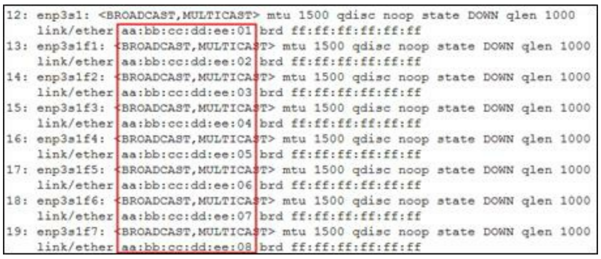


```
12: enp3s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
13: enp3s1f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 14:aa:bb:cc:dd:01 brd ff:ff:ff:ff:ff:ff
14: enp3s1f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
15: enp3s1f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
16: enp3s1f4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
17: enp3s1f5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
18: enp3s1f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
19: enp3s1f7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

下面的脚本是为了实现自动化配置所有VF端口的命令

# counter=1; for i in $(ip a | grep enp3s1 | awk & # 39;{print $2;}& # 39; | tr -d ":"); do ip link set $i addr aa:bb:cc:dd:ee:$counter; ((counter++)); done

注意：需要适当修改enp3s1端口名称为实际端口的名称



```
12: enp3s1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:01 brd ff:ff:ff:ff:ff:ff
13: enp3s1f1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:02 brd ff:ff:ff:ff:ff:ff
14: enp3s1f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:03 brd ff:ff:ff:ff:ff:ff
15: enp3s1f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:04 brd ff:ff:ff:ff:ff:ff
16: enp3s1f4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:05 brd ff:ff:ff:ff:ff:ff
17: enp3s1f5: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:06 brd ff:ff:ff:ff:ff:ff
18: enp3s1f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:07 brd ff:ff:ff:ff:ff:ff
19: enp3s1f7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether aa:bb:cc:dd:ee:08 brd ff:ff:ff:ff:ff:ff
```

本文对Broadcom网卡启用SR-IOV功能做了详细的介绍，需要注意，手动配置网卡的MAC地址，根据不同型号的网卡可能会有差异。

上面是在设置Broadcom网卡，芯片型号为BCM57810时所执行的命令，我们可以看到这款卡的特点是每个VF都有自己的端口名称

相比其他型号网卡，比如intel，可能会有些差异，所有的VF是挂在某个PF下，且没有单独的网卡名称。这时需要执行如下的命令

#ip link set eth2 vf 1 mac 00:52:44:11:22:33

所以要根据实际情况进行修改