

1C. [DATADA] Instalação e configuração do Whitebox Gen 1.0 (Ubuntu 16.10)

Instalação

Instalar OS: Ubuntu 16.10

Configuração

DPDK:

instalar dpdk:

```
sudo apt update  
sudo apt install -y dpdk
```

Edite o arquivo /etc/dpdk/dpdk.conf e adicione a variável:

```
NR_2M_PAGES=2048
```

Depois aloque a memória via kernel e isole os CPUs que serão alocados pelo DPDK, edite o arquivo /etc/default/grub e adicione:

```
GRUB_CMDLINE_LINUX_DEFAULT="isolcpus=2-7 hugepages=2048"
```

O parâmetro anterior (isolcpus) afeta apenas processos do userspace, para evitar que processos do kernel utilizem os CPUs dedicados ao DPDK, instale o pacote sysfsutils:

```
sudo apt-get install -y sysfsutils
```

Para restringir processos do kernel aos cores 6 e 7, adicione a linha abaixo no arquivo /etc/sysfs.conf

```
devices/virtual/workqueue/cpumask = 3  
devices/virtual/workqueue/writeback/cpumask = 3
```



Limitando os núcleos utilizados por processos do kernel

Core:

```
7 6 5 4 3 2 1 0  
0 0 0 0 0 0 1 1 = 0x3 (apenas core0 e core1 serão utilizados)
```

1 = core habilitado
0 = core desabilitado

Para restringir que as interrupções de hardware sejam tratadas pelos cores que serão alocados para o DPDK, adicione a linha abaixo no arquivo /etc/default/irqbalance

```
IRQBALANCE_BANNED_CPUS="FC"
```



Limitando os núcleos utilizados para interrupções de hardware

Core:

```
7 6 5 4 3 2 1 0
1 1 1 1 1 1 0 0 = 0xfc (core2 a core7 não serão utilizados)
```

1 = core habilitado
0 = core desabilitado

Defina as interfaces que usarão o driver dpdk, editando o arquivo /etc/dpdk/interfaces, insira as interface compatíveis com dpdk:

```
<bus>      <id>          <driver> (uio_pci_generic ou vfio_pci)

pci        0000:00:14.0    uio_pci_generic
pci        0000:00:14.1    uio_pci_generic
pci        0000:00:14.2    uio_pci_generic
pci        0000:00:14.3    uio_pci_generic
pci        0000:05:00.0    uio_pci_generic
pci        0000:05:00.1    uio_pci_generic
```

Habilite o inicialização das interfaces dpdk:

```
systemctl enable dpdk
```

Atualize o grub:

```
update-grub
```

Reboot o whitebox:

```
reboot
```

OpenvSwitch:

Instalar pacotes

```
sudo apt install -y openvswitch-switch-dpdk
```

Alterar o daemon do openvswitch padrão para o daemon do openvswitch com suporte a dpdk:

```
sudo update-alternatives --set ovs-vswitchd /usr/lib/openvswitch-switch-dpdk/ovs-vswitchd-dpdk
```

Reinicie o daemon do openvswitch:

```
sudo systemctl restart openvswitch-switch.service
```

Inicializando modulo dpdk do openvswitch:

```
sudo ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-init=true

sudo ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-socket-mem="4096,0"

sudo ovs-vsctl set Open_vSwitch . other_config:pmd-cpu-mask=FC

sudo systemctl restart openvswitch-switch
```



Habilitando os núcleos para dpdk

Core:

```
7 6 5 4 3 2 1 0
1 1 1 1 1 0 0 = 0xfc (core2 a core7 reservados)
```

1 = core habilitado
0 = core desabilitado



Deprecated

Outros:

Foi detectado um conflito entre a alocação das interfaces pelo DPDK e pelo OVS durante a inicialização. Os passos abaixo devem ser utilizados para mitigar o problema até que uma solução definitiva seja encontrada:

```
cat << EOF > /etc/rc.local
#!/bin/bash

sleep 3
systemctl restart openvswitch-switch
exit 0
EOF

chmod 755 /etc/rc.local
systemctl enable rc-local
```

Post-Config:

Foi detectado um conflito entre a alocação das interfaces pelo DPDK e pelo OVS durante a inicialização. Os passos abaixo devem ser utilizados para mitigar o problema até que uma solução definitiva seja encontrada:

vim /lib/systemd/system/openvswitch-nonetwork.service

```
[Unit]
Description=Open vSwitch Internal Unit
PartOf=openvswitch-switch.service
Wants=openvswitch-switch.service
After=dpdk.service

[Service]
Type=oneshot
RemainAfterExit=yes
EnvironmentFile=-/etc/default/openvswitch-switch
ExecStart=/usr/share/openvswitch/scripts/ovs-ctl start \
--system-id=random $OVS_CTL_OPTS
ExecStop=/usr/share/openvswitch/scripts/ovs-ctl stop
```

vim /lib/systemd/system/openvswitch-switch.service

```
[Unit]
Description=Open vSwitch
After=network.target openvswitch-nonetwork.service
Requires=openvswitch-nonetwork.service

[Service]
Type=oneshot
ExecStart=/bin/true
ExecStop=/bin/true
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Configurando o openvswitch para inserir controladores, bridge, filas e interface:

Segue abaixo dois modelos de configuração. Um para criar um switch virtual de 6 portas e outro para criar dois switches virtuais com 4 portas (3 portas físicas e 1 lógica)

Switch Virtual de 6 portas

Bridge

```
sudo ovs-vsctl add-br br-dpdk0 -- set bridge br-dpdk0 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
```

Ports

```
sudo ovs-vsctl add-port br-dpdk0 dpdk0 -- set Interface dpdk0 type=dpdk mtu_request=9600 ofport=1
sudo ovs-vsctl add-port br-dpdk0 dpdk1 -- set Interface dpdk1 type=dpdk mtu_request=9600 ofport=2
sudo ovs-vsctl add-port br-dpdk0 dpdk2 -- set Interface dpdk2 type=dpdk mtu_request=9600 ofport=3
sudo ovs-vsctl add-port br-dpdk0 dpdk3 -- set Interface dpdk3 type=dpdk mtu_request=9600 ofport=4
sudo ovs-vsctl add-port br-dpdk0 dpdk4 -- set Interface dpdk4 type=dpdk mtu_request=9600 ofport=5
sudo ovs-vsctl add-port br-dpdk0 dpdk5 -- set Interface dpdk5 type=dpdk mtu_request=9600 ofport=6
```

Dois Switches Virtuais de 4 portas

Bridge

```
sudo ovs-vsctl add-br br-dpdk0 -- set bridge br-dpdk0 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
sudo ovs-vsctl add-br br-dpdk1 -- set bridge br-dpdk1 datapath_type=netdev protocols=OpenFlow10 fail-mode=secure
```

Portas da Bridge br-dpdk0:

```
sudo ovs-vsctl add-port br-dpdk0 dpdk0 -- set Interface dpdk0 type=dpdk mtu_request=9600 ofport=1
sudo ovs-vsctl add-port br-dpdk0 dpdk2 -- set Interface dpdk2 type=dpdk mtu_request=9600 ofport=2
sudo ovs-vsctl add-port br-dpdk0 dpdk4 -- set Interface dpdk4 type=dpdk mtu_request=9600 ofport=3
sudo ovs-vsctl add-port br-dpdk0 patch0 -- set Interface patch0 type=patch options:peer=patch1 mtu_request=9600 ofport=4
```

Portas da Bridge br-dpdk1:

```
sudo ovs-vsctl add-port br-dpdk1 dpdk1 -- set Interface dpdk1 type=dpdk mtu_request=9600 ofport=1
sudo ovs-vsctl add-port br-dpdk1 dpdk3 -- set Interface dpdk3 type=dpdk mtu_request=9600 ofport=2
sudo ovs-vsctl add-port br-dpdk1 dpdk5 -- set Interface dpdk5 type=dpdk mtu_request=9600 ofport=3
sudo ovs-vsctl add-port br-dpdk1 patch1 -- set Interface patch1 type=patch options:peer=patch0 mtu_request=9600 ofport=4
```

Configuração do Controlador

```
sudo ovs-vsctl set-controller br-dpdk0 tcp:<ip controller>:6653
```

ToDo

- Adicionar passos para configuração persistente das interfaces do DPDK
- Revisar passos para versão 2.6.0
- Verificar configuração usando multiplas filas (n-dpdk-rxqs):
- Verificar driver vfio-pci (Não suportado pelo CPU)
- Verificar suporte a Jumbo Frame na próxima release (OVS 2.6.0 e DPDK 16.10): <https://github.com/openvswitch/ovs/blob/master/INSTALL.DPDK-ADVANCED.md#jumbo>

Referências

- [Ubuntu Server LTS: Data Plane Development Kit](#)
- <https://software.intel.com/en-us/articles/using-open-vswitch-with-dpdk-on-ubuntu>
- <https://insights.ubuntu.com/2016/05/05/the-new-simplicity-to-consume-dpdk/>
- <https://github.com/openvswitch/ovs/blob/branch-2.5/INSTALL.DPDK.md>
- [https://github.com/openvswitch/ovs/blob/branch-2.6/INSTALL.DPDK-ADVANCED.m](https://github.com/openvswitch/ovs/blob/branch-2.6/INSTALL.DPDK-ADVANCED.md)
- [Jumbo Frames in Open vSwitch* with DPDK](#)