



NVIDIA DOCA Troubleshooting

Guide

Table of Contents

Chapter 1. DOCA Infrastructure.....	1
1.1. RShim Troubleshooting and How-Tos.....	1
1.1.1. Another Backend Already Attached.....	1
1.1.2. RShim Driver Not Loading.....	1
1.1.2.1. RShim Driver Not Loading on DPU with Integrated BMC.....	2
1.1.2.2. Change Ownership of RShim from NIC BMC to Host.....	3
1.2. Connectivity Troubleshooting.....	3
1.2.1. Connection (ssh, screen console) to DPU is Lost.....	3
1.2.2. Driver Not Loading in Host Server.....	4
1.2.3. No Connectivity Between Network Interfaces of Source Host to Destination Device.....	5
1.2.4. Uplink in Arm Down While Uplink in Host Server Up.....	5
1.3. Performance Degradation.....	6
1.4. SR-IOV Troubleshooting.....	6
1.4.1. Unable to Create VFs.....	6
1.4.2. No Traffic Between VF to External Host.....	6
1.5. eSwitch Troubleshooting.....	7
1.5.1. Unable to Configure Legacy Mode.....	7
1.5.2. DPU Appears as two Interfaces.....	8
Chapter 2. DOCA Applications.....	9
2.1. SFT Error – SFs.....	9
2.1.1. Error.....	9
2.1.2. Solution.....	9
2.2. SFT Error – VFs.....	9
2.2.1. Error.....	9
2.2.2. Solution.....	10
2.3. Mlx-regex Error.....	10
2.3.1. Error.....	10
2.3.2. Solution.....	10
2.4. EAL Initialization Failure.....	11
2.4.1. Error.....	11
2.4.2. Solution.....	11
2.5. Ring Memory Issue.....	11
2.5.1. Error.....	11
2.5.2. Solution.....	11

2.6. DOCA Apps Using DPDK in Parallel Issue.....	12
2.6.1. Error.....	12
2.6.2. Solution.....	12
2.7. Compilation of DOCA Apps on CentOS.....	13
2.7.1. Error.....	13
2.7.2. Solution.....	13
2.8. Failure to Set Huge Pages.....	13
2.8.1. Error.....	13
2.8.2. Solution.....	14
Chapter 3. DOCA Libraries.....	15
3.1. DOCA Flow Error.....	15
3.1.1. Error.....	15
3.1.2. Solution.....	15
Chapter 4. DOCA SDK Compilation.....	16
4.1. Meson Complains About Missing Dependencies.....	16
4.1.1. Error.....	16
4.1.2. Solution.....	17
4.2. Static Compilation on CentOS: Undefined References to C++.....	18
4.2.1. Error.....	18
4.2.2. Solution.....	18
4.3. Static Compilation on CentOS: Unresolved Symbols.....	18
4.3.1. Error.....	19
4.3.2. Solution.....	19
Chapter 5. Cross-compiling DOCA and CUDA.....	20
5.1. Application Build Error.....	20
5.1.1. Error.....	20
5.1.2. Solution.....	20
Chapter 6. DOCA Services (Containers).....	21
6.1. YAML Syntax Error.....	21
6.1.1. Error.....	21
6.1.2. Solution.....	21

Chapter 1. DOCA Infrastructure

1.1. RShim Troubleshooting and How-Tos

1.1.1. Another Backend Already Attached

Several generations of BlueField DPUs are equipped with a USB interface in which RShim can be routed, via USB cable, to an external host running Linux and the RShim driver.

In this case, typically following a system reboot, the RShim over USB prevails and the DPU host reports RShim status as `another backend already attached`. This is correct behavior, since there can only be one RShim backend active at any given time. However, this means that the DPU host does not own RShim access.

To reclaim RShim ownership safely:

1. Stop the RShim driver on the remote Linux. Run:

```
systemctl stop rshim
systemctl disable rshim
```

2. Restart RShim on the DPU host. Run:

```
systemctl enable rshim
systemctl start rshim
```

The `another backend already attached` scenario can also be attributed to the RShim backend being owned by the BMC in DPUs with integrated BMC. This is elaborated on further down on this page.

1.1.2. RShim Driver Not Loading

Verify whether your DPU features an integrated BMC or not. Run:

```
# sudo sudo lspci -s $(sudo lspci -d 15b3: | head -1 | awk '{print $1}') -vvv |
grep "Product Name"
```

Example output for DPU with integrated BMC:

```
Product Name: BlueField-2 DPU 25GbE Dual-Port SFP56, integrated BMC, Crypto and
Secure Boot Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
```

If your DPU has an integrated BMC, refer to [RShim Driver Not Loading on DPU with Integrated BMC](#).

If your DPU does not have an integrated BMC, refer to [Change Ownership of RShim from NIC BMC to Host](#).

1.1.2.1. RShim Driver Not Loading on DPU with Integrated BMC

1.1.2.1.1. RShim Driver Not Loading on Host

1. Access the BMC via the RJ45 management port of the DPU.
2. Delete RShim on the BMC:

```
systemctl stop rshim
systemctl disable rshim
```

3. Enable RShim on the host:

```
systemctl enable rshim
systemctl start rshim
```

4. Restart RShim service. Run:

```
sudo systemctl restart rshim
```

If RShim service does not launch automatically, run:

```
sudo systemctl status rshim
```

This command is expected to display `active (running)`.

5. Display the current setting. Run:

```
# cat /dev/rshim<N>/misc | grep DEV_NAME
DEV_NAME          pcie-0000:04:00.2
```

This output indicates that the RShim service is ready to use.

1.1.2.1.2. RShim Driver Not Loading on BMC

1. Download the suitable DEB/RPM for RShim (management interface for DPU from the host) driver.
2. Reinstall RShim package on the host.

- ▶ For Ubuntu/Debian, run:

```
sudo dpkg --force-all -i rshim-<version>.deb
```

- ▶ For RHEL/CentOS, run:

```
sudo rpm -Uhv rshim-<version>.rpm
```

3. Restart RShim service. Run:

```
sudo systemctl restart rshim
```

If RShim service does not launch automatically, run:

```
sudo systemctl status rshim
```

This command is expected to display `active (running)`.

4. Display the current setting. Run:

```
# cat /dev/rshim<N>/misc | grep DEV_NAME
DEV_NAME          pcie-0000:04:00.2
```

This output indicates that the RShim service is ready to use.

1.1.2.2. Change Ownership of RShim from NIC BMC to Host

1. Verify that your card has BMC. Run the following on the host:

```
# sudo sudo lspci -s $(sudo lspci -d 15b3: | head -1 | awk '{print $1}') -vvv |
grep "Product Name"
Product Name: BlueField-2 DPU 25GbE Dual-Port SFP56, integrated BMC, Crypto and
Secure Boot Enabled, 16GB on-board DDR, 1GbE OOB management, Tall Bracket, FHHL
```

The product name is supposed to show "integrated BMC".

2. Access the BMC via the RJ45 management port of the DPU.
3. Delete RShim on the BMC:

```
systemctl stop rshim
systemctl disable rshim
```

4. Enable RShim on the host:

```
systemctl enable rshim
systemctl start rshim
```

5. Restart RShim service. Run:

```
sudo systemctl restart rshim
```

If RShim service does not launch automatically, run:

```
sudo systemctl status rshim
```

This command is expected to display `active (running)`.

6. Display the current setting. Run:

```
# cat /dev/rshim<N>/misc | grep DEV_NAME
DEV_NAME          0000:04:00.2
```

This output indicates that the RShim service is ready to use.

1.2. Connectivity Troubleshooting

1.2.1. Connection (ssh, screen console) to DPU is Lost

The UART cable in the Accessories Kit (OPN: MBF20-DKIT) can be used to connect to the DPU console and identify the stage at which BlueField is hanging.

Follow this procedure:

1. Connect the UART cable to a USB socket, and find it in your USB devices.

```
sudo lsusb
```

Bus 002 Device 003: ID 0403:6001 Future Technology Devices International, Ltd
FT232 Serial (UART) IC



Note: For more information on the UART connectivity, please refer to the [DPU's hardware user guide](#) under Supported Interfaces > Interfaces Detailed Description > NC-SI Management Interface.



Note: It is good practice to connect the other end of the NC-SI cable to a different host than the one on which the BlueField DPU is installed.

2. Install the minicom application.

- ▶ For CentOS/RHEL:

```
sudo yum install minicom -y
```

- ▶ For Ubuntu/Debian:

```
sudo apt-get install minicom
```

3. Open the minicom application.

```
sudo minicom -s -c on
```

4. Go to "Serial port setup".

5. Enter "F" to change "Hardware Flow control" to NO.

6. Enter "A" and change to /dev/ttyUSB0 and press Enter.

7. Press ESC.

8. Type "Save setup as dfl".

9. Exit minicom by pressing Ctrl + a + z.

1.2.2. Driver Not Loading in Host Server

What this looks like in dmsg:

```
[275604.216789] mlx5_core 0000:af:00.1: 63.008 Gb/s available PCIe bandwidth,
limited by 8 GT/s x8 link at 0000:ae:00.0 (capable of 126.024 Gb/s with 16 GT/s x8
link)
[275624.187596] mlx5_core 0000:af:00.1: wait_fw_init:316:(pid 943): Waiting for FW
initialization, timeout abort in 100s
[275644.152994] mlx5_core 0000:af:00.1: wait_fw_init:316:(pid 943): Waiting for FW
initialization, timeout abort in 79s
[275664.118404] mlx5_core 0000:af:00.1: wait_fw_init:316:(pid 943): Waiting for FW
initialization, timeout abort in 59s
[275684.083806] mlx5_core 0000:af:00.1: wait_fw_init:316:(pid 943): Waiting for FW
initialization, timeout abort in 39s
[275704.049211] mlx5_core 0000:af:00.1: wait_fw_init:316:(pid 943): Waiting for FW
initialization, timeout abort in 19s
[275723.954752] mlx5_core 0000:af:00.1: mlx5_function_setup:1237:(pid 943): Firmware
over 120000 MS in pre-initializing state, aborting
[275723.968261] mlx5_core 0000:af:00.1: init_one:1813:(pid 943): mlx5_load_one
failed with error code -16
[275723.978578] mlx5_core: probe of 0000:af:00.1 failed with error -16
```

The driver on the host server is dependent on the Arm side. If the driver on Arm is up, then the driver on the host server will also be up.

Please verify that:

- ▶ The driver is loaded in the BlueField DPU
- ▶ The Arm is booted into OS

- ▶ The Arm is not in UEFI Boot Menu
- ▶ The Arm is not hanged

Then:

1. Power cycle on the host server.
2. If the problem persists, please reset nvconfig (`sudo mlxconfig -d /dev/mst/<device> -y reset`), and then power cycle the host.
3. If this problem still persists, please make sure to install the latest bfb image and then restart the driver in host server. Please refer to the [NVIDIA DOCA Installation Guide for Linux](#) for more information.

1.2.3. No Connectivity Between Network Interfaces of Source Host to Destination Device

Verify that the bridge is configured properly on the Arm side.

The following is an example for default configuration:

```
$ sudo ovs-vsctl show
f6740bfb-0312-4cd8-88c0-a9680430924f
    Bridge ovsbr1
        Port pf0sf0
            Interface pf0sf0
        Port p0
            Interface p0
        Port pf0hpf
            Interface pf0hpf
        Port ovsbr1
            Interface ovsbr1
                type: internal
    Bridge ovsbr2
        Port p1
            Interface p1
        Port pf1sf0
            Interface pf1sf0
        Port pf1hpf
            Interface pf1hpf
        Port ovsbr2
            Interface ovsbr2
                type: internal
    ovs_version: "2.14.1"
```

If no bridge configuration exists, refer to section "OpenvSwitch Offload" under [NVIDIA DOCA Switching Support](#).

1.2.4. Uplink in Arm Down While Uplink in Host Server Up

Please check that the cables are connected properly into the network ports of the DPU and the peer device.

1.3. Performance Degradation

Degradation in performance indicates that openvswitch may not be offloaded.

Verify offload state. Run:

```
# ovs-vsctl get Open_vSwitch . other_config:hw-offload
```

- ▶ If `hw-offload = true` – Fast Pass is configured (desired result)
- ▶ If `hw-offload = false` – Slow Pass is configured

If `hw-offload = false`:

- ▶ For RHEL/CentOS, run:

```
# ovs-vsctl set Open_vSwitch . other_config:hw-offload=true;
# systemctl restart openvswitch;
# systemctl enable openvswitch;
```

- ▶ For Ubuntu/Debian, run:

```
# ovs-vsctl set Open_vSwitch . other_config:hw-offload=true;
# /etc/init.d/openvswitch-switch restart
```

1.4. SR-IOV Troubleshooting

1.4.1. Unable to Create VFs

1. Please make sure that SR-IOV is enabled in BIOS.
2. Verify `SRIOV_EN` is true and `NUM_OF_VFS` bigger than 1. Run:


```
# mlxconfig -d /dev/mst/mt41686_pciconf0 -e q |grep -i "SRIOV_EN\|num_of_vf"
Configurations:      Default      Current      Next Boot
*      NUM_OF_VFS      16           16           16
*      SRIOV_EN        True(1)      True(1)      True(1)
```
3. Verify that `GRUB_CMDLINE_LINUX="iommu=pt intel_iommu=on pci=assign-busses"`.

1.4.2. No Traffic Between VF to External Host

1. Please verify creation of representors for VFs inside the Bluefield DPU. Run:

```
# /opt/mellanox/iproute2/sbin/rdma link |grep -i up
...
link mlx5_0/2 state ACTIVE physical_state LINK_UP netdev pf0vf0
...
```

2. Make sure the representors of the VFs are added to the bridge. Run:

```
# ovs-vsctl add-port <bridge_name> pf0vf0
```

3. Verify VF configuration. Run:

```
$ ovs-vsctl show
bb993992-7930-4dd2-bc14-73514854b024
  Bridge ovsbr1
    Port pf0vf0
      Interface pf0vf0
        type: internal
    Port pf0hpf
```

```

    Interface pf0hpf
  Port pf0sf0
    Interface pf0sf0
  Port p0
    Interface p0
Bridge ovsbr2
  Port ovsbr2
    Interface ovsbr2
      type: internal
  Port pf1sf0
    Interface pf1sf0
  Port p1
    Interface p1
  Port pflhpf
    Interface pflhpf
ovs_version: "2.14.1"

```

1.5. eSwitch Troubleshooting

1.5.1. Unable to Configure Legacy Mode

To set devlink to "Legacy" mode in BlueField, run:

```
# devlink dev eswitch set pci/0000:03:00.0 mode legacy
# devlink dev eswitch set pci/0000:03:00.1 mode legacy
```

Please verify that:

- ▶ No virtual functions are open. To verify if VFs are configured, run:

```
# /opt/mellanox/iproute2/sbin/rdma link | grep -i up
link mlx5_0/2 state ACTIVE physical_state LINK_UP netdev pf0vf0
link mlx5_1/2 state ACTIVE physical_state LINK_UP netdev pflvf0
```

If any VFs are configured, destroy them by running:

```
# echo 0 > /sys/class/infiniband/mlx5_0/device/mlx5_num_vfs
# echo 0 > /sys/class/infiniband/mlx5_1/device/mlx5_num_vfs
```

- ▶ If any SFs are configured, delete them by running:

```
/sbin/mlnx-sf -a delete --sfindex <SF Index>
```



Note: You may retrieve the <SF-Index> of the currently installed SFs by running.

```
# mlnx-sf -a show

SF Index: pci/0000:03:00.0/229408
Parent PCI dev: 0000:03:00.0
Representor netdev: en3f0pf0sf0
Function HWADDR: 02:61:f6:21:32:8c
Auxiliary device: mlx5_core.sf.2
netdev: enp3s0f0s0
RDMA dev: mlx5_2

SF Index: pci/0000:03:00.1/294944
Parent PCI dev: 0000:03:00.1
Representor netdev: en3f1pf1sf0
Function HWADDR: 02:30:13:6a:2d:2c
Auxiliary device: mlx5_core.sf.3
netdev: enp3s0f1s0
RDMA dev: mlx5_3
```

Pay attention to the SF Index values. For example:

```
/sbin/mlnx-sf -a delete --sfindex pci/0000:03:00.0/229408
```

```
/sbin/mlnx-sf -a delete --sfindex pci/0000:03:00.1/294944
```

If the error "Error: mlx5_core: Can't change mode when flows are configured" is encountered while trying to configure legacy mode, make sure that:

1. Any configured SFs are deleted (see above for commands).
2. Shut down the links of all interfaces, delete any `ip xfrm` rules, delete any configured OVS flows, and stop openvswitch service. Run:

```
ip link set dev p0 down
ip link set dev p1 down
ip link set dev pf0hpf down
ip link set dev pf1hpf down
ip link set dev vxlan_sys_4789 down

ip x s f ;
ip x p f ;

tc filter del dev p0 ingress
tc filter del dev p1 ingress
tc qdisc show dev p0
tc qdisc show dev p1
tc qdisc del dev p0 ingress
tc qdisc del dev p1 ingress
tc qdisc show dev p0
tc qdisc show dev p1

systemctl stop openvswitch-switch
```

1.5.2. DPU Appears as two Interfaces

What this looks like:

```
# sudo /opt/mellanox/iproute2/sbin/rdma link
link mlx5_0/1 state ACTIVE physical_state LINK_UP netdev p0
link mlx5_1/1 state ACTIVE physical_state LINK_UP netdev p1
```

- ▶ Check if you are working in legacy mode.

```
# devlink dev eswitch show pci/0000:03:00.<0|1>
```

If the following line is printed, this means that you are working in legacy mode:

```
pci/0000:03:00.<0|1>: mode legacy inline-mode none encap enable
```

Please configure the DPU to work in switchdev mode. Run:

```
devlink dev eswitch set pci/0000:03:00.<0|1> mode switchdev
```

- ▶ Check if you are working in separated mode:

```
# mlxconfig -d /dev/mst/mt41686_pciconf0 q | grep -i cpu
* INTERNAL_CPU_MODEL SEPERATED_HOST(0)
```

Please configure the DPU to work in embedded mode. Run:

```
devlink dev eswitch set pci/0000:03:00.<0|1> mode switchdev
```

Chapter 2. DOCA Applications

This chapter deals with troubleshooting issues related to DOCA applications.

2.1. SFT Error – SFs

An SFT error appears when running an SFT-based application on top of SFs.

2.1.1. Error

This error may appear in many applications. For example, when running URL Filter, the error you get is as follows:

```
Forward to SFT IPV4-UDP failed, error=SFT was not initialized
```

The error here is because the SFs you are using are not set as trusted.

2.1.2. Solution

Delete the SFs and create them again as trusted. See section "SF Configuration" in [Scalable Function Setup Guide](#).

2.2. SFT Error – VFs

An SFT error appears when running an SFT-based application on top of SFs.

2.2.1. Error

This error may appear in many applications. For example, when running URL Filter on the host, the error you get is as follows:

```
port-0: SFT init failed err=-22  
[12:56:51:326652][DOCA][ERR][NUTILS:188]: SFT init failed
```

The error here is because of an SFT-related configuration error. When running on the host, it is usually due to a using too many cores.

As stated in the pages of the SFT-based applications, there is a core limit to the SFT mechanism: The SFT supports a maximum of 64 queues. Therefore, the application cannot be run with more than 64 cores.

2.2.2. Solution

When running in setups with more than 64 cores, it is recommended to limit the number of cores used by the worker jobs. This could be achieved using one of the following EAL flags:

- ▶ `-c <core-mask>` – set the hexadecimal bitmask of the cores to run on.
- ▶ `-l <core-list>` – list of cores to run on

For example:

```
/opt/mellanox/doca/applications/url_filter/bin/doca_url_filter -a
0000:3b:00.0,class=regex -a 3b:00.3 -a 3b:00.4 -l 0-64 -- -p
```

2.3. Mlx-regex Error

When running an application that depends on a RegEx device, a RegEx device error may appear.

2.3.1. Error

This error may appear in many applications that use a RegEx device. The error is:

```
mlx5_regex: Rules program failed 22 mlx5_regex: Failed to program rxp rules.
```

The error here is `mlx-regex` is not running.

2.3.2. Solution

1. Make sure that `mlx-regex` is running. On the DPU, run:

```
dpu# systemctl status mlx-regex
```

2. You will probably see the `Active` line as `Failed` or `inactive`. To fix this, on the DPU, run:

```
dpu# systemctl restart mlx-regex
```

3. Make sure that the RegEx device is active. Run:

```
dpu# systemctl status mlx-regex
```

You should see the `Active` line as `active (running)`.

4. If the `Active` line is still `Failed`, you probably need to restart the InfiniBand (RDMA) driver. On the DPU, run:

```
dpu# /etc/init.d/openibd restart
```

5. Restart the RegEx device again. Run:

```
dpu# systemctl restart mlx-regex
```

6. This should fix the issue. Verify that the RegEx device is active again. Run:

```
dpu# systemctl status mlx-regex
```

2.4. EAL Initialization Failure

EAL initialization failure is a common error that may appear while running applications like URL Filter, Application Recognition, or others.

2.4.1. Error

The error looks like this:

```
[DOCA][ERR][NUTILS]: EAL initialization failed
```

There may be many causes for this error. Some of them are as follows:

- ▶ The application requires a `.cdo` file and you gave a wrong path to the file or you did not create the file
- ▶ The application requires huge pages, and you did not allocate huge pages
- ▶ The application requires root privileges to run, and you did not run it as root

2.4.2. Solution

The following solutions are respective to the possible causes listed above:

- ▶ Check that the `.cdo` file exists and that the path that you provided is correct. If the `.cdo` path does not exist, create one using `doxa-dpi-compiler`. Refer to [NVIDIA DOCA DPI Compiler](#) for more information.
- ▶ Allocate huge pages. For example, run (on the host or the DPU, depending on where you are running the application):

```
sudo echo 2048 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

- ▶ Run the application using `sudo` (or as root):

```
sudo <run_command>
```

2.5. Ring Memory Issue

This is a common memory issue when running application on the host.

2.5.1. Error

The error looks as follows:

```
RING: Cannot reserve memory
[13:00:57:290147][DOCA][ERR][UFLTR::Core:156]: DPI init failed
```

The most common cause for this error is lack of memory (i.e., not enough huge pages per worker threads).

2.5.2. Solution

Possible solutions:

- ▶ Recommended: Increase the amount of allocated huge pages. Instructions about allocating huge pages can be found in the second bullet of section [Solution](#).



Note: For an SFT application with 64 cores, it is recommended to increase the allocation from 2048 to 8192.

- ▶ Alternatively, one can also limit the number of cores used by the application, as is explained in section [Solution](#).

2.6. DOCA Apps Using DPDK in Parallel Issue

When running two DOCA apps in parallel that use DPDK, the first app runs but the second one fails.

2.6.1. Error

In this example, the first application is Application Recognition, and the second is URL Filter. The following error is received:

```
Failed to start URL Filter with output: EAL: Detected 16 lcore(s)
EAL: Detected 1 NUMA nodes
EAL: RTE Version: 'MLNX_DPDK 20.11.4.0.3'
EAL: Detected shared linkage of DPDK
EAL: Cannot create lock on '/var/run/dpdk/rte/config'. Is another primary process
running?
EAL: FATAL: Cannot init config
EAL: Cannot init config
[15:01:57:246339][DOCA][ERR][NUTILS]: EAL initialization failed
```

The cause of the error is that the second application is using `/var/run/dpdk/rte/config` when the first application is already using it.

2.6.2. Solution

To run two applications in parallel, the second application needs to be run with DPDK EAL option `--file-prefix <name>`.

In this example, after running Application Recognition (without adding the `eal` option), to run URL Filter, the EAL option must be added. Run:

```
/opt/mellanox/doca/applications/url_filter/bin/doca_url_filter --file-prefix second
-a 0000:01:00.0,class=regex -a 0000:01:00.6,sft_en=1 -a 0000:01:00.7,sft_en=1 -v -c
0xff -- -p
```


2.7. Compilation of DOCA Apps on CentOS

When compiling gRPC-enabled applications on old (7.6) CentOS machines, there is a conflict between the libstdc++ version available out-of-the-box and the one used by DOCA's SDK when building the gRPC packages.

2.7.1. Error

Compiling the gRPC-enabled application results in the following errors:

```
$ meson /tmp/build -Denable_grpc_support=true ; ninja -C /tmp/build
...
l_log_severity.a -Wl,--end-group
/opt/mellanox/grpc/lib/libgrpc++.a(server_cc.cc.o): In function
`grpc::Server::RegisterService(std::string const*, grpc::Service)':
(.text+0x2467): undefined reference to `std::basic_ios<char, std::char_traits<char>
>::operator bool() const'
/opt/mellanox/grpc/lib/libgrpc++.a(server_cc.cc.o): In function
`grpc::Server::RegisterService(std::string const*, grpc::Service)':
(.text+0x249e): undefined reference to `std::basic_ios<char, std::char_traits<char>
>::operator bool() const'
collect2: error: ld returned 1 exit status
```

2.7.2. Solution

Upgrading the devtoolset on the machine to the one used when building the gRPC package resolves the version conflict:

```
$ sudo yum install epel-release
$ sudo yum install centos-release-scl-rh
$ sudo yum install devtoolset-8
$ sudo scl enable devtoolset-8
# This will enable the use of devtoolset-8 to the *current* bash session
$ source /opt/rh/devtoolset-8/enable
```

2.8. Failure to Set Huge Pages

When trying to configure the huge pages from an unprivileged user account, a permission error is raised.

2.8.1. Error

Compiling the gRPC-enabled application results in the following errors:

```
$ sudo echo 600 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
-bash: /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages: Permission denied
```

2.8.2. Solution

Using `sudo` with `echo` works differently than users usually expect. Instead, the command should be as follows:

```
$ echo '600' | sudo tee -a /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
```

Chapter 3. DOCA Libraries

This chapter deals with troubleshooting issues related to DOCA libraries.

3.1. DOCA Flow Error

When trying to add new entry to the pipe, an error is received.

3.1.1. Error

The error happens after trying to add new entry function. The error message would look similar to the following:

```
mlx5_common: Failed to create TIR using DevX
mlx5_net: Port 0 cannot create DevX TIR.
[10:26:39:622581][DOCA][ERR][dpdk_engine]: create pipe entry fail on index:1,
error=Port 0 create flow fail, type 1 message: cannot get hash queue, type=8
```

The issue here seems to be caused by SF/ports configuration.

3.1.2. Solution

To fix the issue, apply the following commands on the DPU:

```
dpu# /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.0 mode
legacy
dpu# /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.1 mode
legacy
dpu# echo none > /sys/class/net/p0/compat/devlink/encap
dpu# echo none > /sys/class/net/p1/compat/devlink/encap
dpu# /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.0 mode
switchdev
dpu# /opt/mellanox/iproute2/sbin/devlink dev eswitch set pci/0000:03:00.1 mode
switchdev
```

Chapter 4. DOCA SDK Compilation

This chapter deals with troubleshooting issues related to compiling DOCA-based programs to use the DOCA SDK (e.g., missing dependencies).

4.1. Meson Complains About Missing Dependencies

As part of DOCA's installation, a basic set of environment variables are defined so that projects (such as DOCA applications) could easily compile against the DOCA SDK, and to allow users easy access to the various DOCA tools. In addition, the set of DOCA applications sometimes rely on various 3rd party dependencies, some of which require specific environment variables so to be correctly found by the compilation environment (meson).

4.1.1. Error

There are multiple forms this error may appear in, such as:

- ▶ DOCA libraries are missing:

```
Dependency doca found: NO (tried pkgconfig and cmake)
meson.build:13:1: ERROR: Dependency "doca" not found, tried pkgconfig and cmake
```

- ▶ DPDK definitions are missing:

```
Dependency libdpdk found: NO (tried pkgconfig and cmake)
meson.build:41:1: ERROR: Dependency "libdpdk" not found, tried pkgconfig and
cmake
```

- ▶ mpicc is missing for the DPA all-to-all application:

```
Program mpicc found: NO
dpa_all_to_all/src/meson.build:23:0: ERROR: Program 'mpicc' not found or not
executable
```

- ▶ gRPC definitions are missing (when gRPC support is activated):

```
Dependency protobuf found: NO (tried pkgconfig and cmake)
meson.build:47:1 ERROR: Dependency "protobuf" not found, tried pkgconfig and
cmake
```

- ▶ gRPC compiler definitions are missing (when gRPC support is activated):

```
Dependency protobuf found: YES 3.15.8.0
Dependency grpc++ found: YES 1.39.0
Program protoc found: NO
meson.build:50:1: ERROR: Program(s) ['protoc'] not found or not executable
```

4.1.2. Solution

All the dependencies mentioned above are installed as part of DOCA's installation, and yet it is recommended to check that the packages themselves were installed correctly. The packages that install each dependency define the environment variables needed by it, and apply these settings per user login session:

- ▶ If DOCA was just installed (on the host or DPU), user session restart is required to apply these definitions (i.e., log off and log in).
- ▶ It is important to compile DOCA using the same logged in user. Logging as `ubuntu` and using `sudo su`, or compiling using `sudo`, will not work.

If restarting the user session is not possible (e.g., automated non-interactive session), the following is a list of the needed environment variables:



Note: All the following examples use the required environment variables for the DPU. For the host, the values should be adjusted accordingly (`aarch64` is for the DPU and `x86` is for the host):

```
aarch64-linux-gnu → x86_64-linux-gnu
```



Tip: It is recommended to define all of the following settings so as to not have to remember which DOCA application requires which module (whether DPDK, gRPC, FlexIO, etc).

DOCA Libraries & Tools:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/doca/lib/aarch64-linux-  
gnu/pkgconfig  
export PATH=${PATH}:/opt/mellanox/doca/tools
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/doca/lib64/pkgconfig  
export PATH=${PATH}:/opt/mellanox/doca/tools
```

DOCA Applications:

- ▶ For Ubuntu:

```
export PATH=${PATH}:/usr/mpi/gcc/openmpi-4.1.5rc2/bin
```

DPDK:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/dpdk/lib/aarch64-linux-  
gnu/pkgconfig
```

- ▶ For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/dpdk/lib64/pkgconfig
```

gRPC:

- ▶ For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/grpc/lib/pkgconfig  
export PATH=${PATH}:/opt/mellanox/grpc/bin
```

► For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/grpc/lib/pkgconfig:/opt/mellanox/grpc/lib64/pkgconfig
export PATH=${PATH}:/opt/mellanox/grpc/bin
```

FlexIO:

► For Ubuntu:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/flexio/lib/pkgconfig
```

► For CentOS:

```
export PKG_CONFIG_PATH=${PKG_CONFIG_PATH}:/opt/mellanox/flexio/lib/pkgconfig
```

4.2. Static Compilation on CentOS: Undefined References to C++

When statically compiling against the DOCA SDK on RHEL 7.x machines, there could be a conflict between the `libstdc++` version available out-of-the-box and the one used when building DOCA's SDK libraries.

4.2.1. Error

There are multiple forms this error may appear in, such as:

```
$ cc test.o -o test_out `pkg-config --libs --static doca`
/opt/mellanox/doca/lib64/libdoca_common.a(doca_common_core_src_doca_dev.cpp.o): In
function `doca_devinfo_rep_list_create':
(.text.experimental+0x2193): undefined reference to
`__cxa_throw_bad_array_new_length'
/opt/mellanox/doca/lib64/libdoca_common.a(doca_common_core_src_doca_dev.cpp.o): In
function `doca_devinfo_rep_list_create':
(.text.experimental+0x2198): undefined reference to
`__cxa_throw_bad_array_new_length'
collect2: error: ld returned 1 exit status
```

4.2.2. Solution

Upgrading the `devtoolset` on the machine to the one used when building the DOCA SDK resolves the undefined references issue:

```
$ sudo yum install epel-release
$ sudo yum install centos-release-scl-rh
$ sudo yum install devtoolset-8
$ sudo scl enable devtoolset-8
# This will enable the use of devtoolset-8 to the *current* bash session
$ source /opt/rh/devtoolset-8/enable
```

4.3. Static Compilation on CentOS: Unresolved Symbols

When statically compiling against the DOCA SDK on RHEL 7.x machines, a [known issue](#) in the default `pkg-config` version (0.27) causes a linking error.

4.3.1. Error

There are multiple forms this error may appear in. For example:

```
$ cc test.o -o test_out 'pkg-config --libs --static doca'  
...  
/opt/mellanox/dpdk/lib64/librte_net_mlx5.a(net_mlx5_mlx5_sft.c.o): In function  
  'mlx5_sft_start':  
mlx5_sft.c:(.text+0x1827): undefined reference to 'mlx5_malloc'  
...
```

4.3.2. Solution

Use an updated version of `pkg-config` or `pkgconf` instead when building applications (as is recommended in DPDK's compilation instructions).

Chapter 5. Cross-compiling DOCA and CUDA

This chapter deals with troubleshooting issues related to DOCA-CUDA cross-compilation.

5.1. Application Build Error

When trying to build with meson, an architecture-related error is received.

5.1.1. Error

The error may happen when trying to build DOCA or DOCA-CUDA applications.

```
ccl: error: unknown value 'corei7' for -march
```

It indicates that some dependency (usually `libdpdk`) is not taken from the host machine (i.e., the machine the executable file should be running on). This dependency should be taken from the Arm dependencies directories (the path is specified in the cross file) but is skipped if the host's `PKG_CONFIG_PATH` environment variable is used instead.

5.1.2. Solution

Make sure that the cross file contains the following `PKG_CONFIG` related definitions:

```
[built-in options]
pkg_config_path = ''
[properties]
pkg_config_libdir = ... // Some content here
```

In addition, verify that `pkg_config_libdir` properly points to all `pkgconfig`-related directories under your cross-build root directory, and that the dependency reported in the error is not missing.

Chapter 6. DOCA Services (Containers)

This section deals with troubleshooting issues related to DOCA-based containers.

6.1. YAML Syntax Error

When deploying the container using the respective YAML file, the pod fails to start.

6.1.1. Error

The error may happen after modifying a service's YAML file, or after copying an example YAML file from one of the guides.



Note: This error can occur when there is a whitespace issue if the YAML file has been copied from one of the guides causing a formatting mistake. It is important to ensure that the space characters used in the files are indeed spaces (' ') and not some other whitespace character.

```
$ crictl pods
POD ID          CREATED          STATE          NAME
  NAMESPACE      ATTEMPT         RUNTIME
$ journalctl -u kubelet
...
Oct 06 12:10:08 dpu-name kubelet[3260]: E1006 12:10:08.552306      3260
file.go:108] "Unable to process watch event" err="can't process config file \"/
etc/kubelet.d/file_name.yaml\": invalid pod: [metadata.name: Invalid value: \"-
dpu-name\": a lowercase RFC 1123 subdomain must consist of lower case alphanumeric
characters, '-' or '.', and must start and end with an alphanumeric character (e.g.
'example.com', regex used for validation is '[a-z0-9]([-a-z0-9]*[a-z0-9])?(\\.[a-
z0-9]([-a-z0-9]*[a-z0-9])?)*') spec.containers: Required value]"
...
```

This indicates that some of the fields in the YAML file fail to comply with RFC 1123.

6.1.2. Solution

Go over the file and ensure that the following applies:

- ▶ Indentation – the file should use spaces (' ') for indentations (2 per indent). Using any other number of spaces causes undefined behavior.

- ▶ Naming conventions – both the pod name and container name have a strict alphabet (RFC 1123). This means that you can only use "-" and not "_", as the latter is an illegal character and cannot be used in the pod/container name. However, for the container's image name, you do use "_" instead of "-". This helps differentiate the two.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assume no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of Mellanox Technologies Ltd. and/or NVIDIA Corporation in the U.S. and in other countries. The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2023 NVIDIA Corporation & affiliates. All rights reserved.