



NVIDIA DOCA Rivermax

Sample Guide

Table of Contents

Chapter 1. Introduction.....	1
Chapter 2. Dependencies.....	2
Chapter 3. Prerequisites.....	3
Chapter 4. Running the Sample.....	4
Chapter 5. Samples.....	5
5.1. List Devices.....	5
5.2. Set CPU Affinity.....	5
5.3. Set Clock.....	6
5.4. Create Stream.....	6
5.5. Create Stream – Header-data Split Mode.....	7

Chapter 1. Introduction

DOCA Rivermax (RMAX) is a DOCA API for NVIDIA Rivermax, an optimized networking SDK for media and data streaming applications. This allows accelerating different networking operations by leveraging the DPU's hardware streaming acceleration technology.

This document describes DOCA Rivermax samples based on the DOCA Rivermax library. These samples illustrate how to use the DOCA Rivermax API to create and configure the stream to receive data. In addition, the samples show how to query network device capabilities (PTP), set a preferred device for querying PTP time, and set CPU affinity for the internal Rivermax thread, using DOCA Rivermax API, to achieve better performance.

For more information about the DOCA Rivermax library, refer to [NVIDIA DOCA Rivermax Programming Guide](#).

Chapter 2. Dependencies

As mentioned in the [NVIDIA DOCA Rivermax Programming Guide](#), the samples require the DOCA Rivermax (RMAX) library to compile properly. To run samples, you also must obtain a Rivermax license.

Chapter 3. Prerequisites

N/A

Chapter 4. Running the Sample

1. Refer to the following documents:

- ▶ [NVIDIA DOCA Installation Guide for Linux](#) for details on how to install BlueField-related software.
- ▶ [NVIDIA DOCA Troubleshooting Guide](#) for any issue you may encounter with the installation, compilation, or execution of DOCA samples.

2. To build a given sample:

```
cd /opt/mellanox/doca/samples/doca_rmax/<sample_name>
meson build
ninja -C build
```



Note: The binary `doca_<sample_name>` will be created under `./build/`.

3. Sample (e.g., `doca_rivermax_create_stream`) usage:

```
Usage: doca_rivermax_create_stream [DOCA Flags] [Program Flags]
DOCA Flags:
  -h, --help                Print a help synopsis
  -v, --version             Print program version information
  -l, --log-level           Set the log level for the program <CRITICAL=20,
ERROR=30, WARNING=40, INFO=50, DEBUG=60>
Program Flags:
  -p, --pci_addr <PCI-ADDRESS> PCI device address
For additional information per sample, use the -h option:
./build/doca_<sample_name> -h
```

Chapter 5. Samples

5.1. List Devices

This sample illustrates how to list all available devices, dump their IPv4 addresses, and tell whether or not the PTP clock is supported.

The sample logic includes:

1. Initializing DOCA Rivermax (RMAX) library.
2. Iterating over the available devices.
3. Dumping their IPv4 addresses
4. Dumping whether a PTP clock is supported for each device.
5. Releasing DOCA Rivermax library.

References:

- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_list_devices/rivermax_list_devices_sample.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_list_devices/rivermax_list_devices_main.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_list_devices/meson.build`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_common.h; /opt/mellanox/doca/samples/doca_rivermax/rivermax_common.c`

5.2. Set CPU Affinity

This sample illustrates how to set the CPU affinity mask for Rivermax internal thread to achieve better performance. This parameter must be set before library initialization; otherwise, it is not applied.

The sample logic includes:

1. Setting CPU affinity using the DOCA Rivermax (RMAX) API.
2. Initializing DOCA Rivermax library.
3. Releasing DOCA Rivermax library.

References:

- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_affinity/rivermax_set_affinity_sample.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_affinity/rivermax_set_affinity_main.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_affinity/meson.build`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_common.h; /opt/mellanox/doca/samples/doca_rivermax/rivermax_common.c`

5.3. Set Clock

This sample illustrates how to set the PTP clock device to be used internally in DOCA Rivermax (RMAX).

The sample logic includes:

1. Opening a DOCA device with a given PCIe address.
2. Initializing the DOCA Rivermax library.
3. Setting the device to use for obtaining PTP time.
4. Releasing the DOCA Rivermax library.

References:

- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_clock/rivermax_set_clock_sample.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_clock/rivermax_set_clock_main.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_set_clock/meson.build`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_common.h; /opt/mellanox/doca/samples/doca_rivermax/rivermax_common.c`

5.4. Create Stream

This sample illustrates how to create a stream, create a flow and attach it to the created stream, and finally to start receiving data buffers (based on the attached flow).

The sample logic includes:

1. Opening a DOCA device with a given PCIe address.
2. Initializing the DOCA RMAX library.
3. Creating an input stream.
4. Creating the context from the created stream.
5. Initializing DOCA core related objects.
6. Setting the attributes of the created stream.

7. Creating a flow and attaching it to the created stream.
8. Starting to receive data buffers.
9. Clean up—detaches flow and destroys it, destroys created stream and DOCA core related objects.

References:

- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream/rivermax_create_stream_sample.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream/rivermax_create_stream_main.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream/meson.build`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_common.h; /opt/mellanox/doca/samples/doca_rivermax/rivermax_common.c`

5.5. Create Stream – Header-data Split Mode

This sample illustrates how to create a stream in header-data split mode when packet headers and payload are split to different RX buffers.

The sample logic includes:

1. Opening a DOCA device with a given PCIe address.
2. Initialize the DOCA RMAX library.
3. Creating an input stream.
4. Creating a context from the created stream.
5. Initializing DOCA core related objects.
6. Setting attributes of the created stream. Chaining buffers and setting header size to non-zero is essential to create a stream with header-data split mode.
7. Creating a flow and attaching it to the created stream.
8. Starting to receive data to split buffers.
9. Clean up—detaches flow and destroys it, destroys created stream and DOCA core related objects.



Note: When running "Create Stream" samples, with or without header-split mode, an IPv4 address of value 192.168.105.2 must be configured to the required PCIe device prior to the run so the samples could run as expected.

If a different IPv4 address is configured, the user can either change the IPv4 address to 192.168.105.2 or change the IPv4 addresses in `init_config()` function found in the samples source code to the configured one before building and compiling the samples.

References:

- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream_hds/rivermax_create_stream_hds_sample.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream_hds/rivermax_create_stream_hds_main.c`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_create_stream_hds/meson.build`
- ▶ `/opt/mellanox/doca/samples/doca_rivermax/rivermax_common.h; /opt/mellanox/doca/samples/doca_rivermax/rivermax_common.c`

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.