NVIDIA DOCA Applications Overview

Guide

MLNX-15-060539 _v1.4 | November 2022

Table of Contents

Chapter 1. Introduction

DOCA applications are an educational resource provided as a guide on how to program on the NVIDIA® BlueField® DPU using DOCA API.

For instructions regarding the development environment and installation, refer to the NVIDIA DOCA Developer Guide and the NVIDIA DOCA Installation Guide for Linux respectively.

1.1. Installation

DOCA applications are installed under `/opt/mellanox/doca/applications` with each application having its own dedicated folder. Each application installs two directories:

- `src` – contains the source code and compilation files
- `bin` – contains pre-built binaries alongside runtime configuration files

The application's binary is located under `/opt/mellanox/doca/applications/<application_name>/bin/doca_<application_name>`.

1.2. Compilation

As applications are shipped alongside their sources, developers may want to modify some of the code during their development process and then recompile the applications. The files required for the compilation are the following:

- `/opt/mellanox/doca/applications/meson.build` – main compilation file for a project that contains all the applications
- `/opt/mellanox/doca/applications/meson_options.txt` – configuration file for the compilation process
- `/opt/mellanox/doca/applications/<application_name>/src/meson.build` – application-specific compilation definitions

To recompile the application:

1. Move to the application's directory:
   ```
   cd /opt/mellanox/doca/applications
   ```
2. Prepare the compilation definitions:
   ```
   meson build
   ```

3. Compile the applications:
```
ninja -C build
```

The generated applications will be located under the `./build/` directory.

1.3. Developer Configurations

When recompiling the applications, meson compiles them by default in "debug" mode. Therefore, the binaries would not be optimized for performance as they would include the debug symbol. For comparison, the application binaries shipped as part of DOCA's installation are compiled in "release" mode. To compile the applications in something other than debug, please consult Meson's configuration guide.

The applications also offer developers the ability to use the DOCA Developer Log (`DOCA_DLOG`) on top of the existing DOCA Runtime Log (`DOCA_LOG`). Enabling the developer log during compilation activates various developer log messages that were left out of the release compilation. Activating the developer log may be done through `enable_developer_log` in the `meson_options.txt` file, or directly from the command line:

1. Prepare the compilation definitions to use the developer log:
```
meson build -Denable_developer_log=true
```
2. Compile the application(s):
```
ninja -C build
```

Chapter 2. Applications

2.1. Allreduce

This application is a collective operation that allows data from many processing units to be collected and merged into a global result before being delivered to all processing units using an operator. The application is implemented using the UCX communication framework, which leverages the DPU's low-latency and high-bandwidth utilization of its network engine.

2.2. Application Recognition

This application identifies applications that are in use on a monitored networking node. The application is based on the deep packet inspection (DPI) library, which leverages DPU capabilities such as regular expression (RXP) acceleration engine, hardware-based connection tracking, and more.

2.3. App Shield Agent

This application describes how to build secure process monitoring and is based on the DOCA APSH library, which leverages DPU capabilities such as regular expression (RXP) acceleration engine, hardware-based DMA, and more.

2.4. DMA Copy

This application describes how to transfer files between the DPU and the host. The application is based on the direct memory access (DMA) library, which leverages hardware acceleration for data copy for both local and remote memory.

2.5. DNS Filter

This application offloads DNS requests from the host to the DPU's Arm cores which allows reducing CPU overhead as they allow further DNS processing (e.g., allow/deny list) to be done. The application is based on the DOCA Flow and RegEx libraries which leverage DPU capabilities such as regular expression (RXP) acceleration engine, building generic execution pipes in HW, and more.

2.6. East-West Overlay Encryption

This application, IPsec, sets up encrypted connections between different devices and works by encrypting IP packets and authenticating the packets' originator. It is based on a strongSwan solution which is an open-source IPsec-based VPN solution.

2.7. File Compression

This application shows how to compress and decompress data using hardware acceleration and to send and receive it. The application is based on the DOCA Compress and DOCA Comm-Channel libraries.

2.8. File Integrity

This application shows how to send and receive files in a secure way using the hardware Crypto engine. It is based on the DOCA SHA and DOCA Comm-Channel libraries.

2.9. File Scan

This application describes how to scan a file using the hardware RegEx engine to find whether there are matches according to the compiled regular expressions. It is based on the DOCA RegEx library which leverages the DPU's regular expression (RXP) acceleration engine.

2.10. Firewall

This application applies network security based on DOCA Flow gRPC and is used for remote programming of the DPU hardware. It leverages DPU capabilities such as building generic execution pipes in hardware, monitoring incoming and outgoing network traffic, and more.

2.11. IPS

This application monitors a network for malicious activities or policy violations and is based on a deep packet inspection (DPI) library, which leverages DPU capabilities such as the regular expression (RXP) acceleration engine, hardware-based connection tracking, and more.

2.12. L2 Reflector

This application uses the data path accelerator (DPA) engine to intercept network traffic and swap the source and destination MAC addresses of each packet. It is based on the FlexIO API which leverages DPU capabilities such as high-speed DPA.

2.13. L4 OVS Firewall

This application performs basic access control list (ACLs) operations. It allows the identification of different flows based on L3/L4 headers and executes different actions using Open vSwitch (OVS) commands.

2.14. NAT

This application, network address translation, switches packets with local IP addresses to global ones and vise versa. It is based on the DOCA Flow library which leverages DPU hardware capabilities such as building generic execution pipes in the hardware, executing specific actions on the traffic, and more.

2.15. Secure Channel

This application is used to establish a secure, network-independent communication channel between the host and the DPU based on the DOCA Comm Channel library.

2.16. Simple Forward VNF

This application is a forwarding application that takes VXLAN traffic from a single RX port and transmits it on a single TX port. It is based on the DOCA Flow library which leverages DPU capabilities such as building generic execution pipes in the hardware, and more.

2.17. Switch

This application is used to establish internal switching between representor ports on the DPU. It is based on the the DOCA Flow library which leverages DPU capabilities such as building generic execution pipes in the hardware, and more.

2.18. URL Filter

This application limits access by comparing web traffic against a database to prevent users from different threats (e.g., malware, harmful sites, phishing). It is based on a deep packet inspection (DPI) library, which leverages DPU capabilities such as the regular expression (RXP) acceleration engine, hardware-based connection tracking, and more.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051

http://www.nvidia.com