NVIDIA DOCA PCC Counter Tool
Table of contents

<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Prerequisites</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Execution</td>
</tr>
</tbody>
</table>
This document provides instruction on the usage of the PCC Counter tool.

Introduction

The PCC Counter tool is used to print PCC-related hardware counters. The output counters help debug the PCC user algorithm embedded in the DOCA PCC application.

Prerequisites

DOCA 2.2.0 and higher.

Description

If NVIDIA® BlueField®-3 is operating in DPU mode, the script must be executed on the Arm side. If BlueField-3 is operating in NIC mode, the script must be executed on the host side.

The following performance counters are supported for PCC:

- **MAD_RTT_PERF_CONT_REQ** – the number of RTT requests received in total
- **MAD_RTT_PERF_CONT_RES** – the number of RTT responses received in total
- **SX_EVENT_WRED_DROP** – the number of TX events dropped due to the CC event queue being full
- **SX_RTT_EVENT_WRED_DROP** – the number of "TX event with RTT request sent indication" dropped due to the CC event queue being full
- **ACK_EVENT_WRED_DROP** – the number of Ack events dropped due to the CC event queue being full

Info

Refer to NVIDIA BlueField Modes of Operation for more information on the DPU's modes of operation.
• NACK_EVENT_WRED_DROP – the number of Nack events dropped due to the CC event queue being full

• CNP_EVENT_WRED_DROP – the number of CNP events dropped due to the CC event queue being full

• RTT_EVENT_WRED_DROP – the number of RTT events dropped due to the CC event queue being full

• HANDLED_SXW_EVENTS – the number of handled CC events related to SXW

• HANDLED_RXT_EVENTS – the number of handled CC events related to RXT

• DROP_RTT_PORT0_REQ – the number of RTT requests dropped in total from port 0

• DROP_RTT_PORT1_REQ – the number of RTT requests dropped in total from port 1

• DROP_RTT_PORT0_RES – the number of RTT responses dropped in total from port 0

• DROP_RTT_PORT1_RES – the number of RTT responses dropped in total from port 1

• RTT_GEN_PORT0_REQ – the number of RTT requests sent in total from port 0

• RTT_GEN_PORT1_REQ – the number of RTT requests sent in total from port 1

• RTT_GEN_PORT0_RES – the number of RTT responses sent in total from port 0

• RTT_GEN_PORT1_RES – the number of RTT responses sent in total from port 1

• PCC_CNP_COUNT – the number of CNP received in total, regardless of whether it is handled or ignored

**Execution**

To use the PCC Counter:

1. Initialize all supported hardware counters. Run:

```bash
sudo ./pcc_counters.sh set /dev/mst/mt41692_pciconf0
```
2. Query all supported hardware counters. Run:

```
sudo ./pcc_counters.sh query /dev/mst/mt41692_pciconf0
```

Example output:

```
sudo ./pcc_counters.sh query /dev/mst/mt41692_pciconf0
-------------PCC Counters-------------
Counter: MAD_RTT_PERF_CONT_REQ Value: 000000000028b85b
Counter: MAD_RTT_PERF_CONT_RES Value: 000000000028b85a
Counter: SX_EVENT_WRED_DROP     Value: 0000000000000000
Counter: SX_RTT_EVENT_WRED_DROP Value: 0000000000000000
Counter: ACK_EVENT_WRED_DROP    Value: 0000000000000000
Counter: NACK_EVENT_WRED_DROP   Value: 0000000000000000
Counter: CNP_EVENT_WRED_DROP    Value: 0000000000000000
Counter: RTT_EVENT_WRED_DROP    Value: 0000000000000000
Counter: HANDLED_SXW_EVENTS     Value: 000000000932543a
Counter: HANDLED_RXT_EVENTS     Value: 000000000028b85c
Counter: DROP_RTT_PORT0_REQ     Value: 0000000000000000
Counter: DROP_RTT_PORT1_REQ     Value: 0000000000000000
Counter: DROP_RTT_PORT0_RES     Value: 0000000000000000
Counter: DROP_RTT_PORT1_RES     Value: 0000000000000000
```
| Counter: RTT_GEN_PORT0_REQ | Value: 0000000000000000 |
| Counter: RTT_GEN_PORT1_REQ | Value: 000000000028b85c |
| Counter: RTT_GEN_PORT0_RES | Value: 0000000000000000 |
| Counter: RTT_GEN_PORT1_RES | Value: 000000000028b85d |
| Counter: PCC_CNP_COUNT     | Value: 0000000000000000 |