



VF Dynamic MSIX

Table of contents

Firmware Configuration

MSIX

MSIX Capability

Reallocating VF MSIX

MSIX Limitations

Queue Pairs

QP Capability

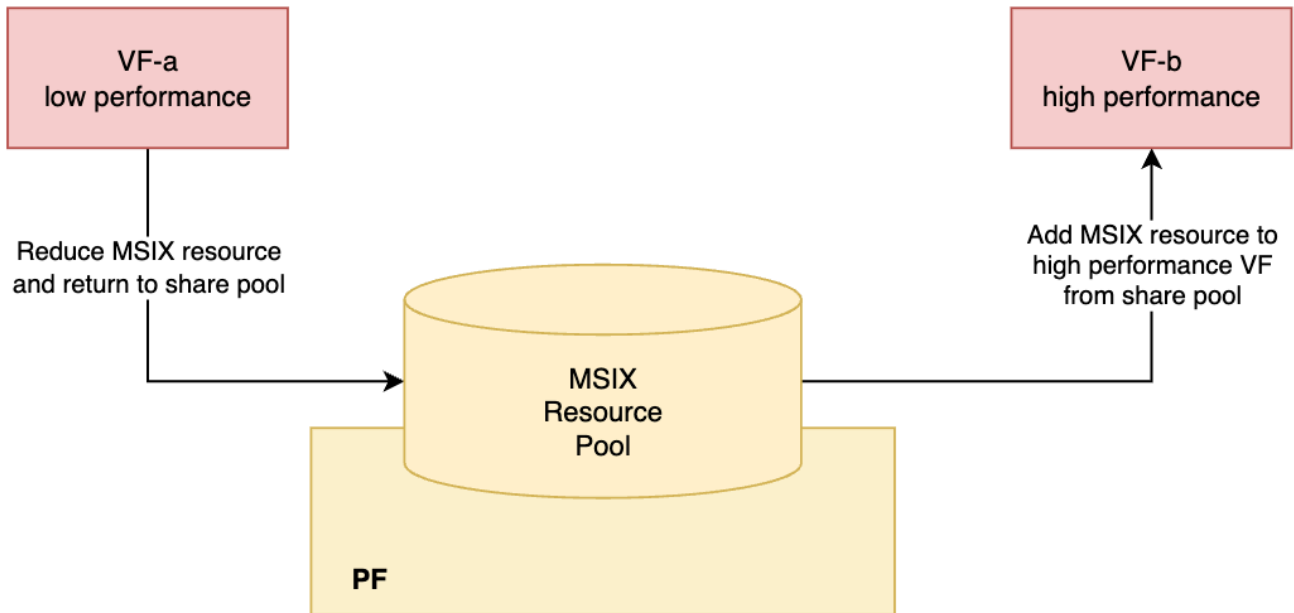
Reallocating VF QPs

QP Limitations

In virtio-net controller, each VF gets the same number of MSIX and virtqueues (VQs) so that each data VQ has a MSIX assigned. This means that changing the number of MSIX updates the number of VQs.

By default, each VF is assigned with the same number of MSIX, the default number is determined by the minimum of `NUM_VF_MSIX` and `VIRTIO_NET_EMULATION_NUM_MSIX`.

Using dynamic VF MSIX, a VF can be assigned with more MSIX/queues than its default. MSIX hardware resources of all VF devices are managed by PF via a shared MSIX pool. The user can reduce the MSIX of one VF, thus releasing its MSIX resources to the shared pool. On the other hand, another VF can be assigned with more MSIX than its default to gain more performance.



Firmware Configuration

The emulation VF device uses `VIRTIO_NET_EMULATION_NUM_VF_MSIX` to set the MSIX number.

`VIRTIO_NET_EMULATION_NUM_VF_MSIX` is available to set the MSIX number of the emulation VF device. For the emulation VF device, uses the new configuration `VIRTIO_NET_EMULATION_NUM_VF_MSIX` instead of the old configuration `NUM_VF_MSIX`.

- If `VIRTIO_NET_EMULATION_NUM_VF_MSIX` $\neq 0$, `VIRTIO_NET_EMULATION_NUM_MSIX` is used for the PF only, and VF uses `VIRTIO_NET_EMULATION_NUM_VF_MSIX`.

For example, to configure the default MSIX number for a VF to 32:

```
[dpu]# mlxconfig -y -d 03:00.0 s VIRTIO_NET_EMULATION_NUM_MSIX=32 VIRTIO_NET_EMULATION_NUM_VF_MSIX=32
```

- If `VIRTIO_NET_EMULATION_NUM_VF_MSIX` $= 0$, `VIRTIO_NET_EMULATION_NUM_MSIX` is used for the PF and VF.

The default number of MSIX for each VF is determined by `minimum(NUM_VF_MSIX, VIRTIO_NET_EMULATION_NUM_MSIX)`. For example, to configure the default MSIX number for a VF to 32:

```
[dpu]# mlxconfig -y -d 03:00.0 s VIRTIO_NET_EMULATION_NUM_MSIX=32 NUM_VF_MSIX=32
```

Power cycle the BlueField and host to have the `mlxconfig` taking effect.

MSIX

MSIX Capability

The MSIX pool for VFs is managed by their PF. To check the share pool size, run the following command (using PF 0 as example):

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 8 | grep -i msix_num_pool_size
```

By default, the share pool size is empty (0), since all MSIX resources have already been allocated to VFs evenly. Upon reducing the MSIX of one or more VFs, the reduced MSIX is released back to the pool.

However, the number of MSIX can be assigned to a given VF is also bound by capability. To check those caps, run the following command:

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 10 | grep -i  
max_msix_num  
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 10 | grep -i  
min_msix_num
```

To check the currently assigned number of MSIX, run the following command:

```
[dpu]# virtnet query -p 0 -v 0 | grep num_msix
```

If `num_msix` is less than `max_msix_num` cap, more MSIX can be assigned to the VF.

Reallocating VF MSIX

To allocate more MSIX to one VF, there should be MSIX available from the pool. This is done by reducing the MSIX from another VF(s).

The following example shows the steps to reallocate MSIX from VF1 to VF0, assuming that each VF has `32` MSIX available as default:

1. Unbind both VF devices from host driver.

```
[host]# echo <vf0_bdf> > /sys/bus/pci/drivers/virtio-  
pci/unbind
```

```
[host]# echo <vf1_bdf> > /sys/bus/pci/drivers/virtio-  
pci/unbind
```

2. Reduce the MSIX of VF1.

```
[dpu]# virtnet modify -p 0 -v 1 device -n 4
```

3. Check pool size of PF0.

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 8 | grep -i  
msix_num_pool_size
```

Confirm the reduced MSIX are added to the share pool.

4. Increase the MSIX of VF0.

```
[dpu]# virtnet modify -p 0 -v 0 device -n 48
```

5. Check the MSIX of VF0.

```
[dpu]# virtnet query -p 0 -v 0 | grep -i num_msix
```

6. Bind both VF devices to host driver.

```
[host]# echo <vf0_bdf> > /sys/bus/pci/drivers/virtio-pci/bind  
[host]# echo <vf1_bdf> > /sys/bus/pci/drivers/virtio-pci/bind
```

Note

The number of MSIX must be an even number greater than 4.

MSIX Limitations

- MSIX and QP configuration is mutually exclusive (i.e., only one of them can be configured at a time). For example, the following `modify` command should result in failure:

```
[dpu]# virtnet modify -p 0 -v 1 device -qp 2 -n 6
```

- To use a VF, make sure to assign a valid MSIX number:

```
[dpu]# virtnet modify -p 0 -v 1 device -n 10
```

The minimum number of MSIX resources required for the VF to load the host driver is 4 if `VIRTIO_NET_F_CTRL_VQ` is negotiated, or 2 if it is not.

- The MSIX resources of a VF can be reduced to 0, but doing so prevents the VF from functioning.

```
[dpu]# virtnet modify -p 0 -v 1 device -n 0
```

Queue Pairs

Queue pairs (QPs) are the number of data virtio queue (VQ) pairs. Each VQ pair has one transmit (TX) queue and one receive (RX) queue. These pairs are dedicated to handling data traffic and do not include control or admin VQs.

QP Capability

The QP pool for VFs is managed by their PF.

To check the shared pool size, run the following command (using PF 0 as example):

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 13 | grep -i qp_pool_size
```

By default, the shared pool size is empty (0), since all QP resources have already been allocated to VFs evenly. Upon reducing the QP of one or more VFs, the reduced QP is released back into the pool.

However, the number of QPs assignable to a VF depends on its supported capabilities. To verify these capabilities, run the following command:

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 12 | grep -i max_num_of_qp
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 12 | grep -i min_num_of_qp
```

To check the currently assigned number of QPs, run the following command:

```
[dpu]# virtnet query -p 0 -v 0 | grep max_queue_pairs
```

If `max_queue_pairs` is less than `max_num_of_qp` cap, then more QPs can be assigned to the VF.

Reallocating VF QPs

To allocate more QPs to one VF, there should be QPs available from the pool as explained in the previous section.

The following example illustrates the process of reallocating a QP from VF1 to VF0, assuming that each VF initially has 32 QPs available by default:

1. Unbind both VF devices from the host driver:

```
[host]# echo <vf0_bdf> > /sys/bus/pci/drivers/virtio-  
pci/unbind  
[host]# echo <vf1_bdf> > /sys/bus/pci/drivers/virtio-  
pci/unbind
```

2. Reduce the number of QPs VF1 has:

```
[dpu]# virtnet modify -p 0 -v 1 device -qp 1
```

3. Check the pool size of PF0 and confirm that the reduced number of QPs are added to the shared pool:

```
[dpu]# virtnet list | grep -i '"pf_id": 0' -A 13 | grep -i  
qp_pool_size
```

4. Increase the number of QPs VF0 has:

```
[dpu]# virtnet modify -p 0 -v 0 device -qp 23
```

5. Check the number of QPs VF0 has:

```
[dpu]# virtnet query -p 0 -v 0 | grep -i max_queue_pairs
```

6. Bind both VF devices to the host driver:

```
[host]# echo <vf0_bdf> > /sys/bus/pci/drivers/virtio-pci/bind  
[host]# echo <vf1_bdf> > /sys/bus/pci/drivers/virtio-pci/bind
```

Note

The number of QPs must be greater than 0.

QP Limitations

- QP and MSIX configuration is mutually exclusive (i.e., only one of them can be configured at a time). For example, the following `modify` command should result in failure:

```
[dpu]# virtnet modify -p 0 -v 1 device -qp 2 -n 6
```

- To use a VF, assign it with a valid QP number:

```
[dpu]# virtnet modify -p 0 -v 1 device -n 4
```

The minimum number of QP resources which allows the VF to load the host driver is 1.

- The QP resources of a VF can be reduced to 0. However, the VF would not be functional in this case.

```
[dpu]# virtnet modify -p 0 -v 1 device -qp 0
```

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 ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes 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 and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Copyright 2024, NVIDIA. PDF Generated on 11/12/2024