NVIDIA GPUDirect with PCI Pass-Through Virtualization

TABLE OF CONTENTS

CHANGELOG

Date	Author	Notes
2017-01-19	Will Davis	Initial draft of PCI Express Virtual Peer-to-Peer Approval Capability Structure specification.
2017-02-03	Will Davis	Updated specification to clarify: ● When the capability is read by NVIDIA system software ● Recommendations about where the hypervisor should embed the capability structure
2017-05-18	Will Davis	Added note clarifying that the existence of the capability in a GPU’s configuration space will not affect the PCI topology presented to virtual machine, or the peer-to-peer path reported by NVIDIA tools within a virtual machine.
2017-07-31	Will Davis	Updated the “4+4 GPU” example layout to include the correct value for GPUs under the second socket.
2019-06-12	Will Davis	Updated the recommended PCI configuration space offset for NVIDIA Turing architecture GPUs.
2023-05-30	Will Davis	Updated the PCI configuration space offset table row for NVIDIA Turing architecture GPUs to "Turing and later"

OVERVIEW

NVIDIA GPUs can be driven by the NVIDIA driver stack running in a virtual machine (VM) when full control over the device is granted to the VM using PCI pass-through virtualization. Nearly all of the NVIDIA driver functionality is available to software running within the VM, but applications using peer-to-peer technology to communicate with other GPUs within the same VM do not work out-of-the-box.

This is because NVIDIA qualifies specific chipsets and PCI Express switches for use with GPUDirect, and the NVIDIA driver stack will use the PCI Express topology of the system it's running on to determine whether the hardware is capable of supporting the peer-to-peer communication required by GPUDirect. In a virtual environment, the PCI Express topology is flattened and obfuscated by the hypervisor to create a uniform environment to present to software inside the VM, so the NVIDIA driver stack is unable to validate the hardware for GPUDirect use.

Unfortunately, a robust solution to this problem is not as simple as always allowing GPUDirect when the GPUs are in such a virtual environment. There are certain physical device topologies, such as GPUs attached to multiple CPU sockets, that are incapable of supporting peer-to-peer communication between subsets of the GPUs when they are passed-through to a VM. In a bare-metal configuration, the NVIDIA driver stack is able to detect such topologies and organize GPUs in mutually exclusive “cliques”, or groups of GPUs which are capable of peer-to-peer communication. Without a small amount of additional topology information from the hypervisor, the NVIDIA driver stack must either reject all peer mappings between GPUs, or inadvertently allow peer mappings between GPUs that the underlying physical topology cannot support.

PROPOSED SOLUTION

This section outlines a mechanism for hypervisors to provide a small amount of additional information to the NVIDIA driver stack inside the VM, by trapping reads to an address range in each pass-through device's PCI configuration space and returning data for a small “emulated” PCI capability structure.

PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL CAPABILITY DEFINITION

The PCI Express Virtual Peer-to-Peer Approval Capability is entirely emulated by the hypervisor in the PCI configuration space of pass-through NVIDIA GPU devices. The capability structure is designed to provide the minimum information needed for the NVIDIA driver stack to make informed decisions about peer-to-peer capabilities of the underlying GPU while being flexible for various physical topologies and future expansion, if necessary.

The values encoded in the capability are static, and should not change while the device is under the control of the NVIDIA driver stack in the virtual machine. The NVIDIA driver stack will read the values from this capability during driver initialization and, if the NVIDIA driver is loaded, during system resume.

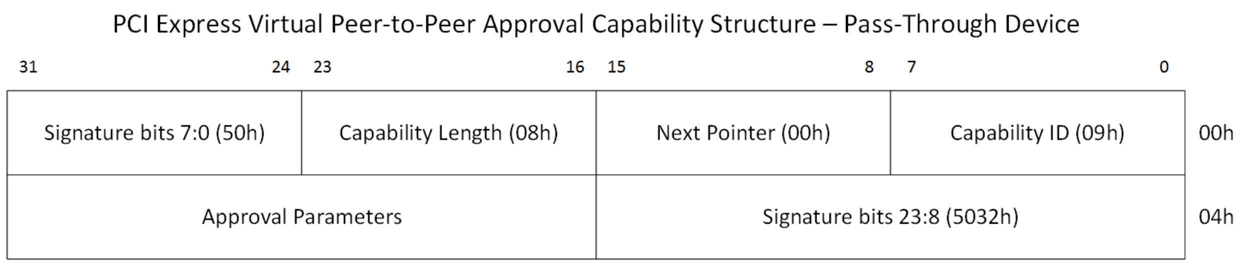
The hypervisor may locate the capability structure at any unused offset within PCI configuration space, but should not mask any other structure that is already present in the physical GPU's PCI configuration space. NVIDIA has reserved space within PCI configuration space specifically for this structure, and recommends that hypervisors place the capability structure at that offset.

RECOMMENDED VIRTUAL PEER-TO-PEER APPROVAL CAPABILITY OFFSET

NVIDIA GPU Architectures	Recommended PCI Configuration Space Offset
Kepler, Maxwell, Pascal, Volta	C8h
Turing and later	D4h

Regardless of the specific offset of the capability structure in PCI configuration space, the structure must be linked in the PCI capability list of the NVIDIA GPU, to make the emulated capability reachable by capability list traversal.

The existence of the capability structure cannot and will not alter the PCI topology presented to the system and NVIDIA tools within the virtual machine, nor will it change the peer-to-peer path (the common upstream PCI device) reported by NVIDIA tools. This is because in pass-through configurations where PCI Express traffic is routed through an IOMMU, peer-to-peer traffic will also be routed to the host bridge, instead of directly between the peer devices.



PCI Express Virtual Peer-to-Peer Approval Capability Structure – Pass-Through Device

31 … 24	23 … 16	15 … 8	7 … 0	
Signature bits 7:0 (50h)	Capability Length (08h)	Next Pointer (00h)	Capability ID (09h)	00h
Approval Parameters		Signature bits 23:8 (5032h)		04h

CAPABILITY ID FOR PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL

Bits	Field	Description
7::0	CAP_ID	The value of 09h in this field identifies this capability as a vendor-specific capability.

NEXT POINTER FOR PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL

Bits	Field	Description
7::0	NXT_PTR	This should always be NULL (00h) for this capability, because the hypervisor should always emulate it at the end of the capability list in legacy PCI configuration space.

CAPABILITY LENGTH FOR PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL

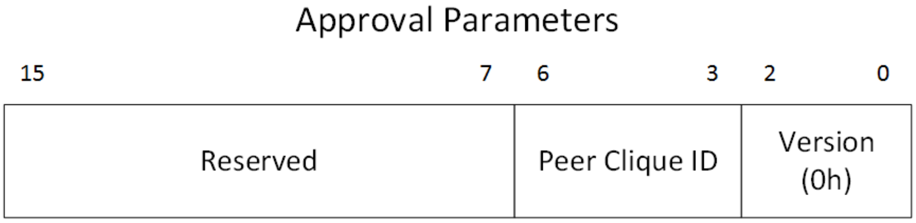
Bits	Field	Description
7::0	CAP_LEN	This field identifies the length of the vendor-specific capability, and should have a value of 08h.

SIGNATURE FOR PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL

The signature field distinguishes this vendor-specific capability from other vendor-specific capabilities that may exist in the device's legacy PCI configuration space.

Bits	Field	Description
23::0	SIG	This should always be "P2P" (503250h) for this capability.

APPROVAL PARAMETERS FOR PCI EXPRESS VIRTUAL PEER-TO-PEER APPROVAL



Bits	Field	Description
15::7	RSVD	These bits are reserved for future use and should return a value of 00h when read.
6::3	PEER_CLIQUE_ID	This is a small unique integer that specifies which peer “clique” the pass-through device belongs. All pass-through devices that have the same value for this field are assumed to be physically capable of exchanging PCI Express peer transactions.
2::0	VERSION	This field describes the version of the capability structure. The current version for this structure is 0.

REQUIREMENTS

To support this new model of peer-to-peer approval, changes will be required in both the hypervisor and NVIDIA driver stack.

HYPERVISOR

The hypervisor will require at least the following changes:

1. When passing a PCI device through to the VM, identify which pass-through devices should be capable of peer-to-peer with each other. Peer-to-peer configurations should be qualified by the system integrator on bare-metal prior to enabling peer-to-peer between pass-through devices, so this information may be loaded from a static configuration provided by the system integrator.
2. When setting up an NVIDIA GPU device for pass-through, emulate the above defined PCI capability at the specified offset in PCI configuration space of the pass-through device, using data from step 1 to set the PEER_CLIQUE_ID value accordingly.

By setting up this capability on pass-through NVIDIA GPUs, the system integrator warrants that the underlying physical topology supports PCI Express peer-to-peer communication between all devices with the same PEER_CLIQUE_ID value.

NVIDIA DRIVER

The NVIDIA driver in the virtual environment will require the following changes:

1. Locate the PCI Express Virtual Peer-to-Peer Approval Capability at the specified offset in each pass-through device's PCI configuration space, if it exists, and extract the PEER_CLIQUE_ID value.
2. Update peer-to-peer approval algorithms to allow GPUs with the same PEER_CLIQUE_ID, if found in step 1, to set up peer-to-peer mappings.

NVIDIA commits to locating and using the PEER_CLIQUE_ID for peer-to-peer approval as described above, when present, for all pass-through NVIDIA Tesla GPU products, across the operating systems on which the NVIDIA GPU driver is supported for pass-through devices.

EXAMPLE CONFIGURATIONS

These examples are provided as a quick summary of how a hypervisor might use the PCI Express Virtual Peer-to-Peer Approval Capability to enable GPUDirect between pass-through NVIDIA GPUs.

2 GPU

In this configuration, an NVIDIA Tesla K80 (dual-GPU) are passed-through to a virtual machine by the hypervisor. The K80 is physically connected to an Intel Xeon E5-2690 in a LGA 2011-v3 socket over PCI Express.

There is only one peer clique in this scenario, so both GPUs would have the below capability (for peer clique 0) emulated in its PCI configuration space.

PCI Express Virtual Peer-to-Peer Approval Capability – 2 GPU

31 24	23 16	15 8	7 0	
Signature bits 7:0 (50h)	Capability Length (08h)	Next Pointer (00h)	Capability ID (09h)	00h
Approval Parameters (0000h)		Signature bits 23:8 (5032h)		04h

4+4 GPU

In this configuration, 4 NVIDIA Tesla K80s (each dual-GPU) are passed-through to a virtual machine by the hypervisor. The K80s are physically connected in pairs to two Intel Xeon E5-2698s in LGA 2011-v3 sockets over PCI Express. The two CPUs are connected via QPI.
In this case, there are two peer cliques, each with four GPUs, corresponding to the two physical CPU sockets.
Each GPU would have the below capability emulated in its PCI configuration space - four GPUs would belong to peer clique 0, and four GPUs would belong to peer clique 1.

PCI Express Virtual Peer-to-Peer Approval Capability – 4+4 GPU

31 24	23 16	15 8	7 0	
Signature bits 7:0 (50h)	Capability Length (08h)	Next Pointer (00h)	Capability ID (09h)	00h
Approval Parameters (0000h or 0008h, depending on socket)		Signature bits 23:8 (5032h)		04h