持续助力数据中心虚拟化：KVM里的虚拟GPU

AGENDA

NVIDIA vGPU on KVM

NVIDIA vGPU product overview

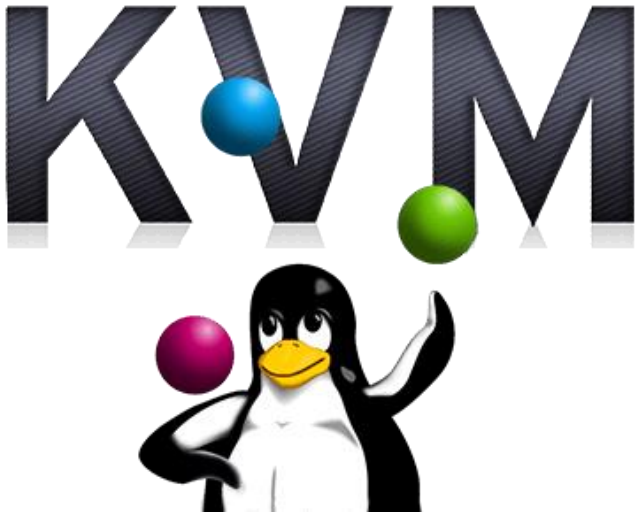
Summary

Q&A

NVIDIA VGPU ON KVM

KVM HYPERVISOR

Fast Growing and Widely Adopted

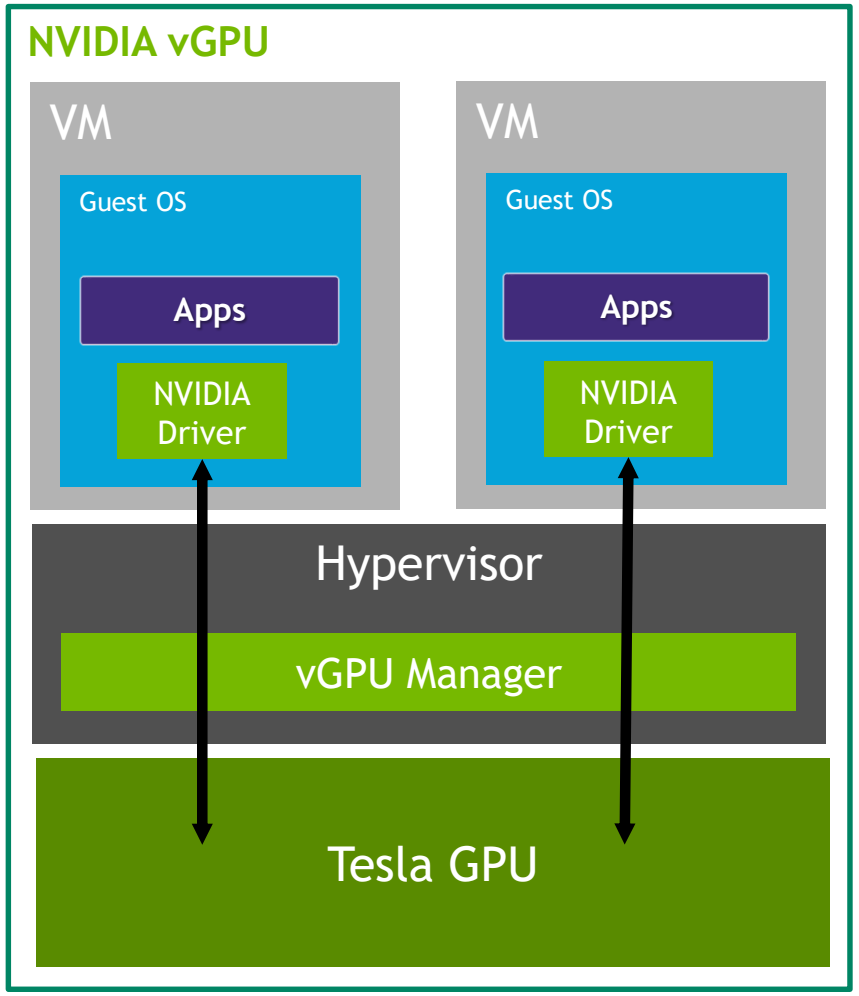


and more ...

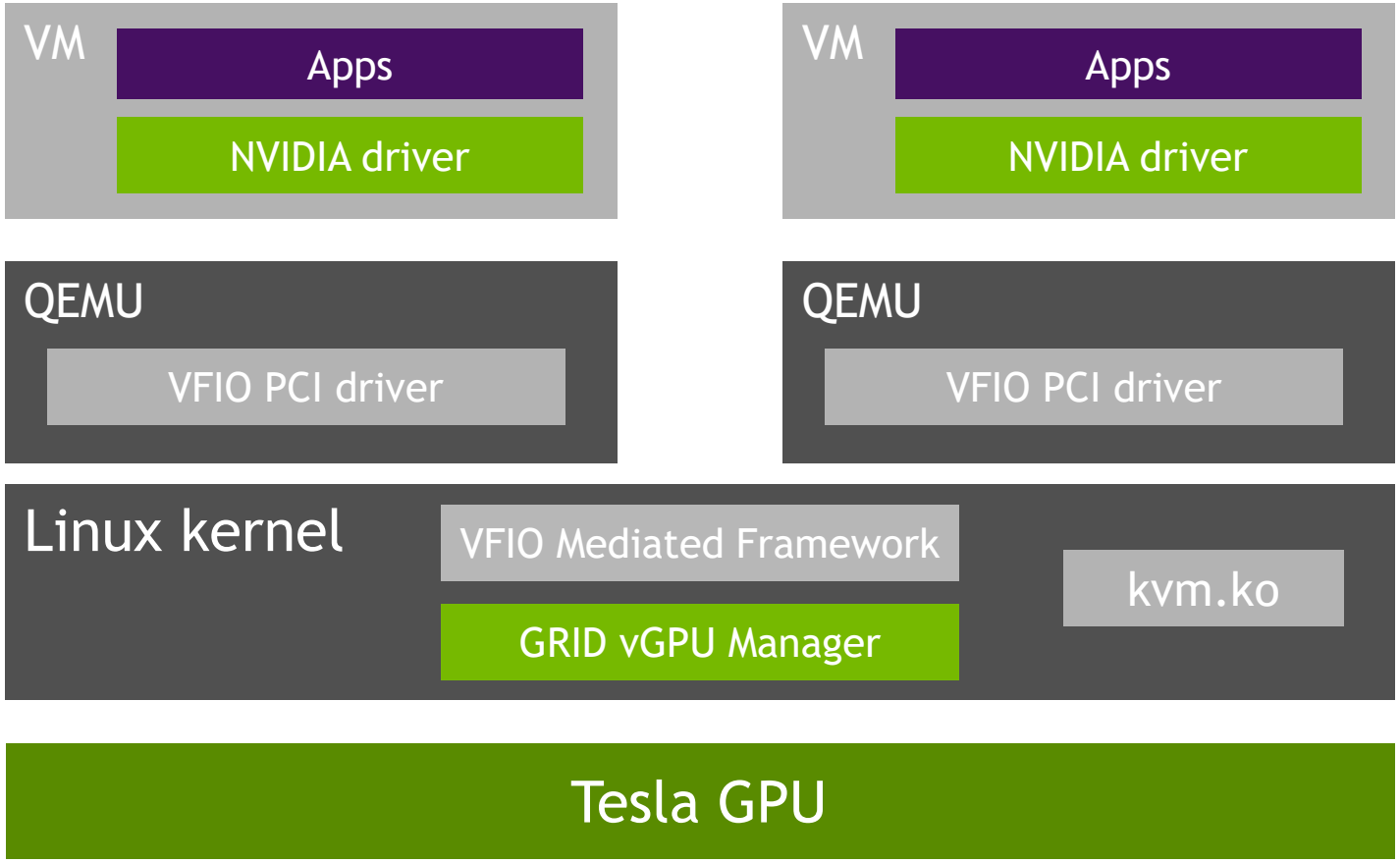
NVIDIA vGPU

Performance, Density, Manageability – for GPU

- Fully enables NVIDIA GPU on virtualized platforms
  - Wide availability - supported by all major hypervisors
  - Great app compatibility – NVIDIA driver inside VM
  - Great performance – VM direct access to GPU hardware
- Improved density
  - Multiple VMs can share one GPU
- Highly manageable
  - NVIDIA host driver, management tools retain full control of the GPU
  - vGPU suspend, resume, live migration enables workloads to be transparently moved between GPUs



NVIDIA vGPU KVM Architecture 101



Based on upstream VFIO-mediated architecture

No VFIO UAPI change

Mediated device managed by generic sysfs interface or libvirt

Mediated Device Framework – VFIO MDEV

A common framework for mediated I/O devices

Present in KVM Forum 2016, upstream since Linux 4.10, kernel maintainer - Kirti Wankhede @ NVIDIA

Mediated core module (new)

- Mediated bus driver, create mediated device
- Physical device interface for vendor driver callbacks
- Generic mediate device management user interface (sysfs)

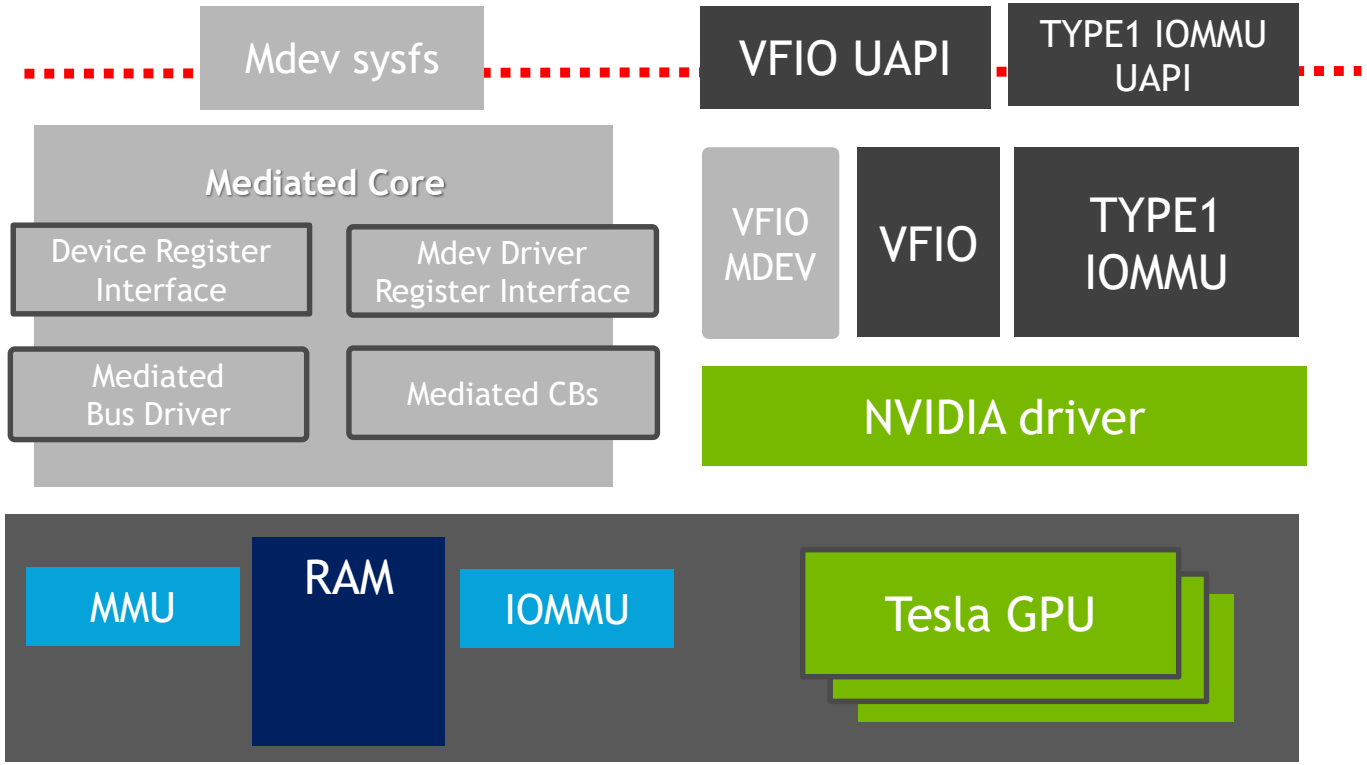
Mediated device module (new)

- Manage created mediated device, fully compatible with VFIO user API

VFIO IOMMU driver (enhancement)

- VFIO IOMMU API TYPE1 compatible, easy to extend to non-TYPE1

Mediated Device Framework - NVIDIA



Mediated Device Framework

Mediated Device sysfs

After NVIDIA driver device registration, under physical device sysfs:

create : create a virtual device (aka mdev device)

mdev_supported_types : supported mdev and configuration of this device

name: GRID M60-8Q, for example.

description: num_heads, frl_config=60, framebuffer size, max_resolution, max_instance

device_api: vfio-pci

Mdev node: /sys/bus/mdev/devices/$mdev_UUID/

https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-bus-vfio-mdev

Creation of vGPU

Generate vGPU mdev UUID via *uuid-gen*, for example "98d19132-f8f0-4d19-8743-9efa23e6c493"

Create vGPU device:

```
root@cjia-vgx-kvm:~

[root@cjia-vgx-kvm ~]# echo "98d19132-f8f0-4d19-8743-9efa23e6c493" > /sys/bus/pci/drivers/nvidia/0000:05:00.0/mdev_supported_types/nvidia-22/create
[root@cjia-vgx-kvm ~]#
```

dmesg output:

```
root@cjia-vgx-kvm:~

[root@cjia-vgx-kvm ~]# dmesg
[ 2345.877060] iommu: Adding device 98d19132-f8f0-4d19-8743-9efa23e6c493 to group 49
[ 2345.884587] vfio_mdev 98d19132-f8f0-4d19-8743-9efa23e6c493: MDEV: group_id = 49
[root@cjia-vgx-kvm ~]#
```

Start vGPU VM

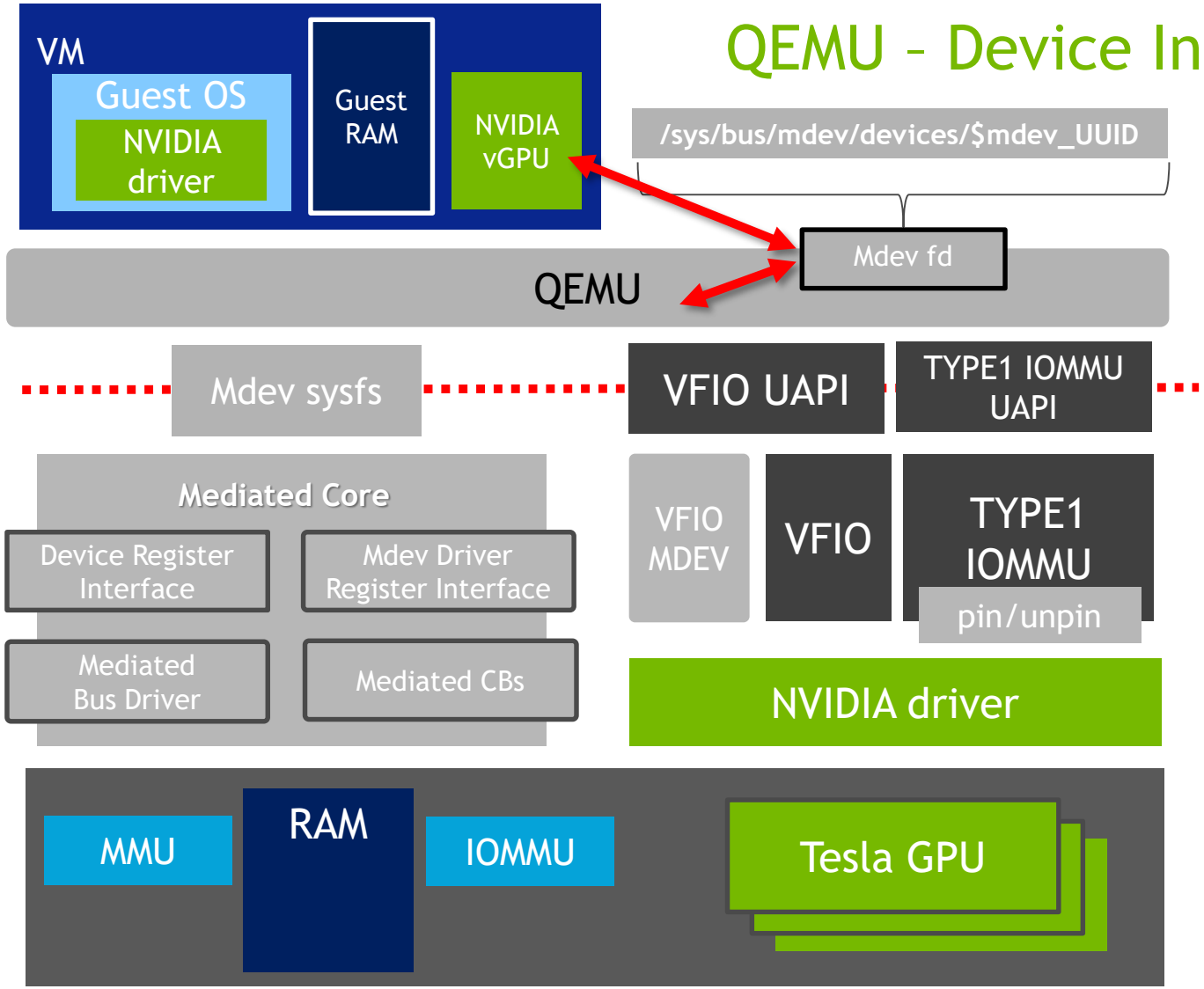
Directly via QEMU command line:

```
-sysfsdev /sys/bus/mdev/devices/98d19132-f8f0-4d19-8743-9efa23e6c493
```

libvirt:

```
<hostdev mode='subsystem' type='mdev' managed='no' model='vfio-pci'>
      <source>
        <address uuid='98d19132-f8f0-4d19-8743-9efa23e6c493'/>
      </source>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
 </hostdev>
```

Mediated Device Framework



QEMU – Device Initialization

Registers VFIO MDEV as driver

NVIDIA driver registers devices

NVIDIA driver registers Mediated CBs

User writes mdev sysfs to create mdev device

QEMU calls VFIO API to add VFIO dev to IOMMU container, group, get fd back

QEMU access device fd and present it into VM

Mediated Device Access

Emulated vs Passthrough

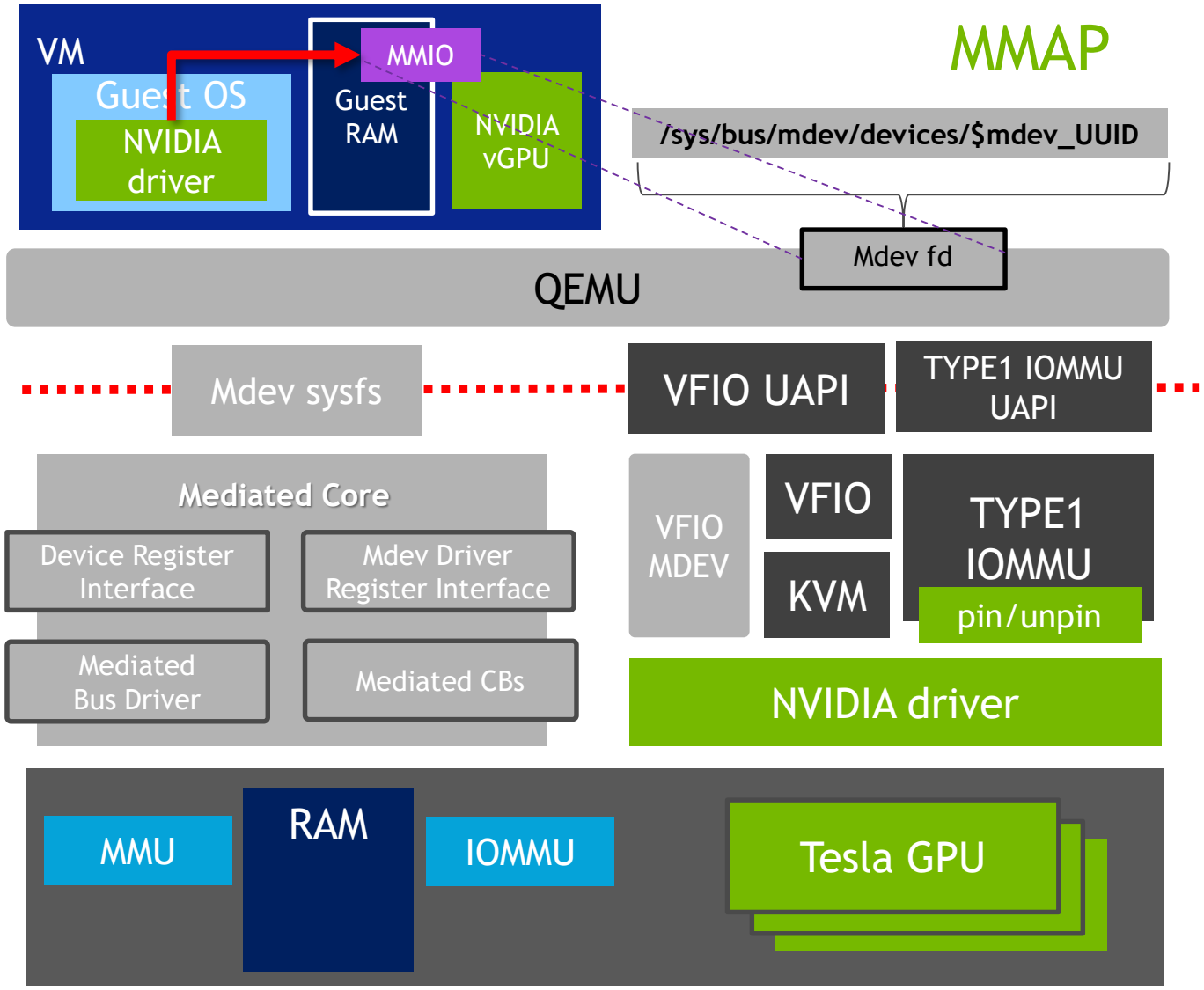
Virtual device memory region are presented inside guest for consistent view of vendor driver

Access to emulated regions are redirected to mediated vendor driver for virtualization support

Access to passthrough region are directly sent to device corresponding region for max performance

- 1st access redirected to mediated vendor driver for CPU page table setup

Mediated Device Access



QEMU gets region info via VFIO UAPI from vendor driver thru VFIO MDEV and Mediated CBs

NVIDIA driver accesses MDEV MMIO trapped region backed by mdev fd triggers EPT violation

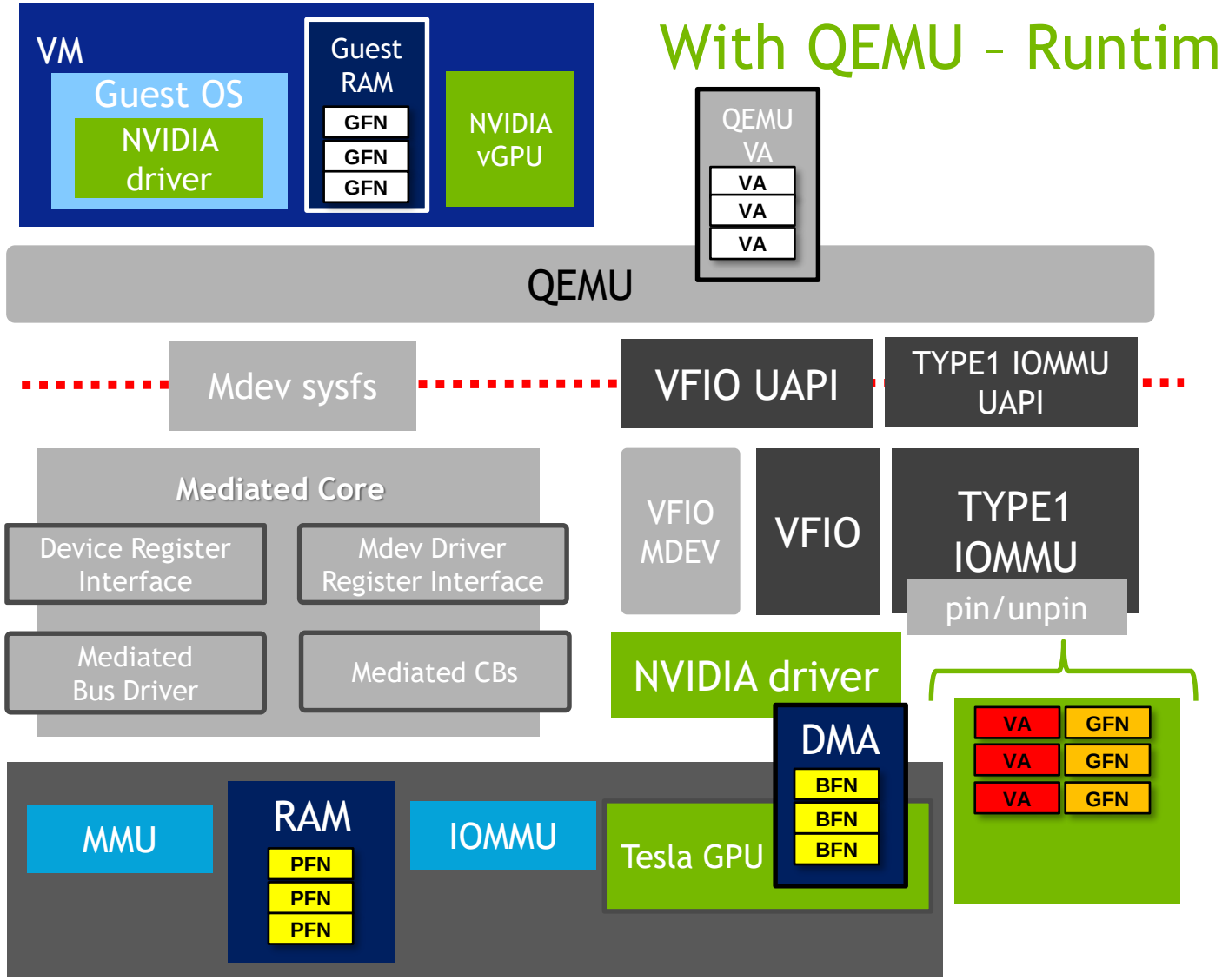
KVM services EPT violation and forwards to QEMU VFIO PCI driver

QEMU convert request from KVM to R/W access to MDEV fd

RW handled by NVIDIA driver via Mediated CBs and VFIO MDEV

Mediated DMA translation

With QEMU – Runtime Memory pinning



QEMU Starts

Memory regions gets added by QEMU

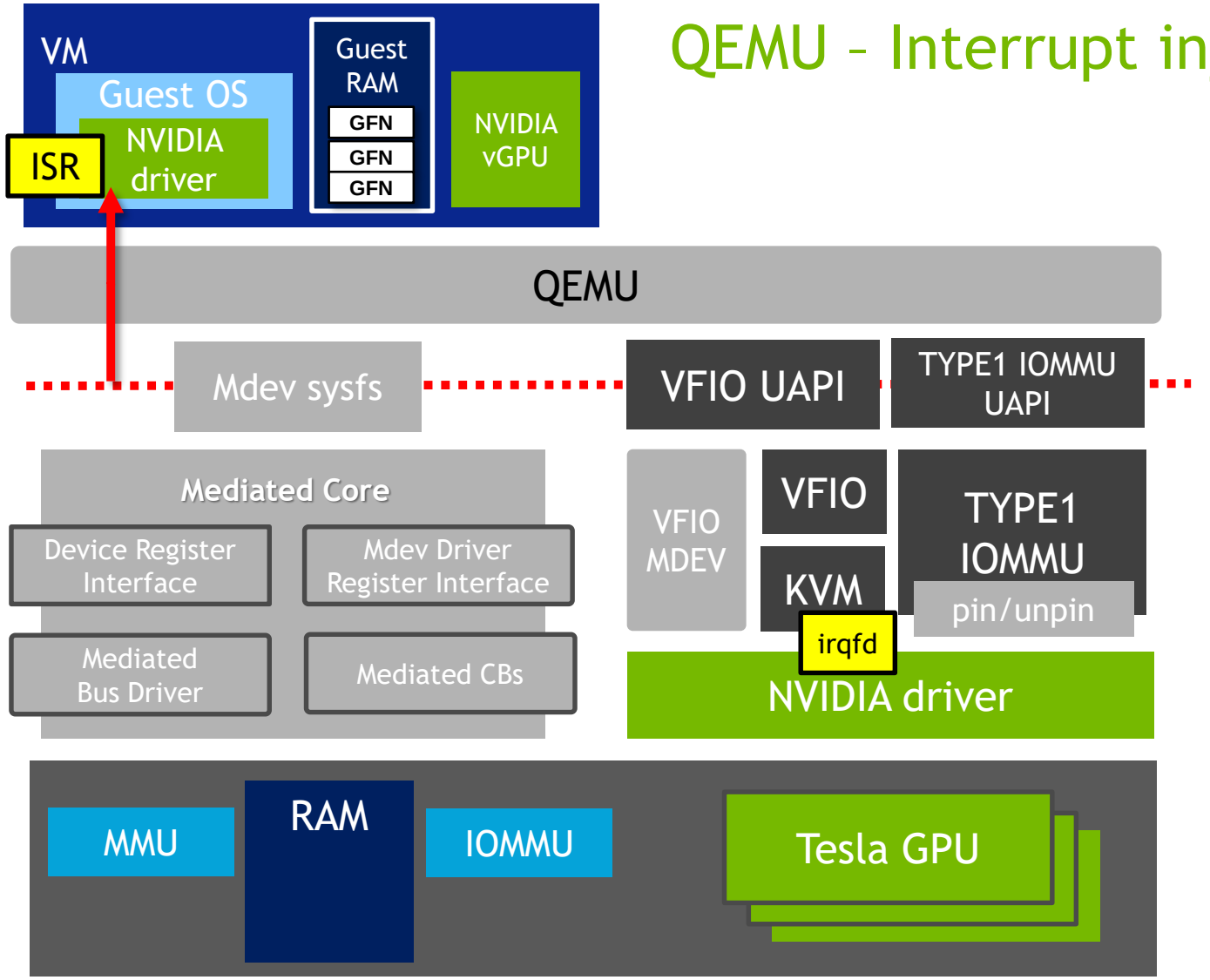
QEMU calls VFIO_DMA_MAP via Memory listener

TYPE1 IOMMU tracks <VA, GFN>

NVIDIA driver pin/translate GFN by TYPE1 IOMMU to get PFN

NVIDIA driver call pci_map_sg to map PFNs to BFN, program DMA

Mediated Device Framework



QEMU – Interrupt injection in runtime

QEMU query MDEV supported interrupt type, provided by NVIDIA driver

QEMU setups up KVM IRQFD

QEMU notifies the vendor driver IRQFD via VFIO PCI UAPI

NVIDIA driver inject interrupt by signaling on eventfd, trigger guest ISR

NVIDIA vGPU Migration

For KVM

Proposing a much more complete solution for VFIO MDEV -based migration

- Pre-copy support
- Dirty page tracking
- Iterative memory copy

Initial NVIDIA KVM vGPU patch posted in Oct 2018

[RFC,v1,0/4] Add migration support for VFIO device

How to Integrate to your Hypervisor?

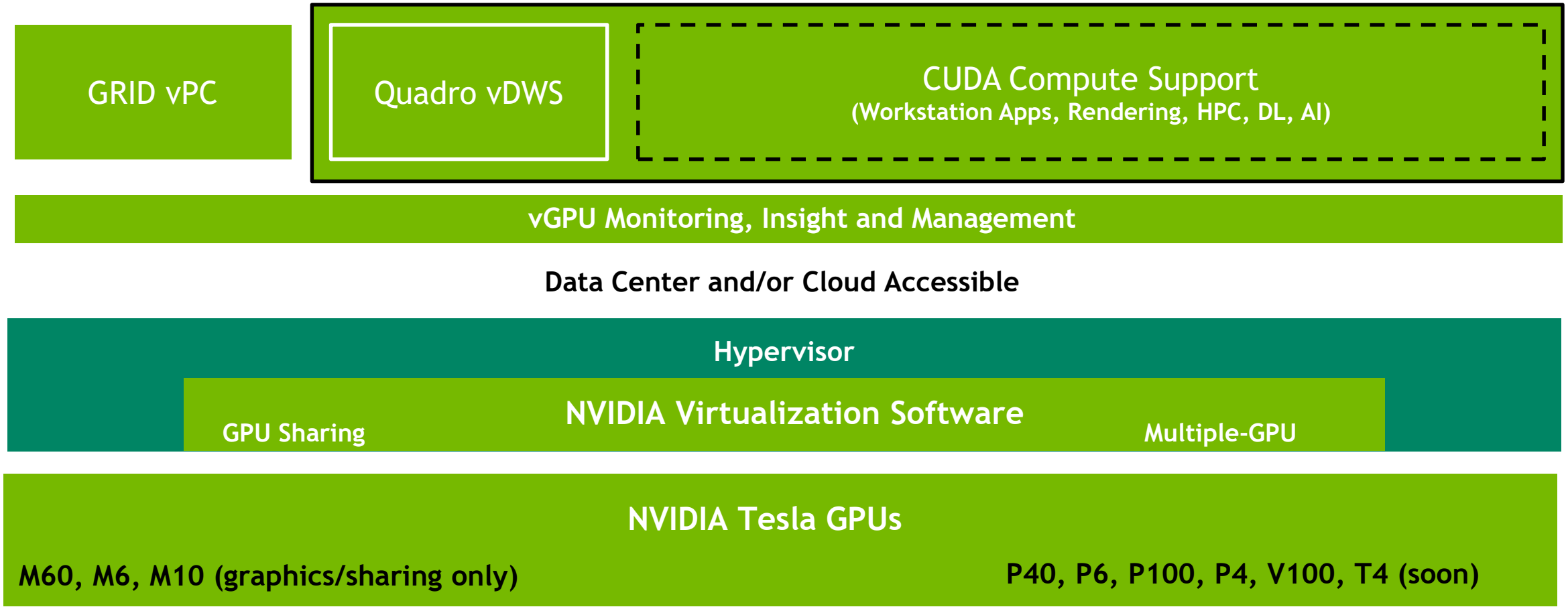
Integrate major VFIO MDEV, KVM and QEMU patches

Contact NVIDIA vGPU Product Management team for evaluation driver and license

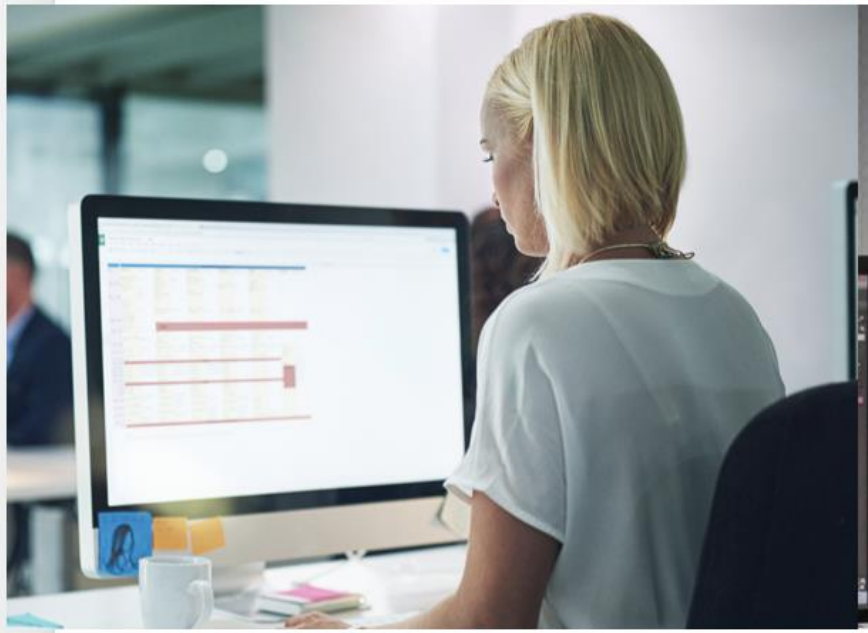
Let us know your integration experience

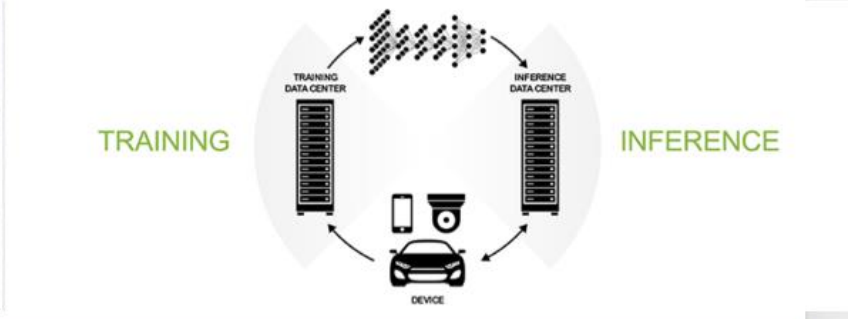
NVIDIA GRID GPU VIRTUALIZATION PLATFORM

Industry standard virtualization platform



VGPU EVERYWHERE FOR EVERYONE, EVERY WORKLOAD





GPU ACCELERATED DATA CENTER & CLOUD

CISCO · DELL · Hewlett Packard Enterprise · HUAWEI · Microsoft Azure · amazon web services · Google Cloud Platform

lenovo · NUTANIX · SUPERMICRO · IBM Bluemix · FUJITSU

NVIDIA VGPU PRODUCT OVERVIEW

NON-STOP INNOVATION

GRID

GPU in data center.

GRID 4

Virtual Desktop Infrastructure.
Density and performance.

vGPU 5

Provision and management.

vGPU 6/7

More hypervisors and CSPs.
Advanced data center features.

vGPU 8

AI/DL.

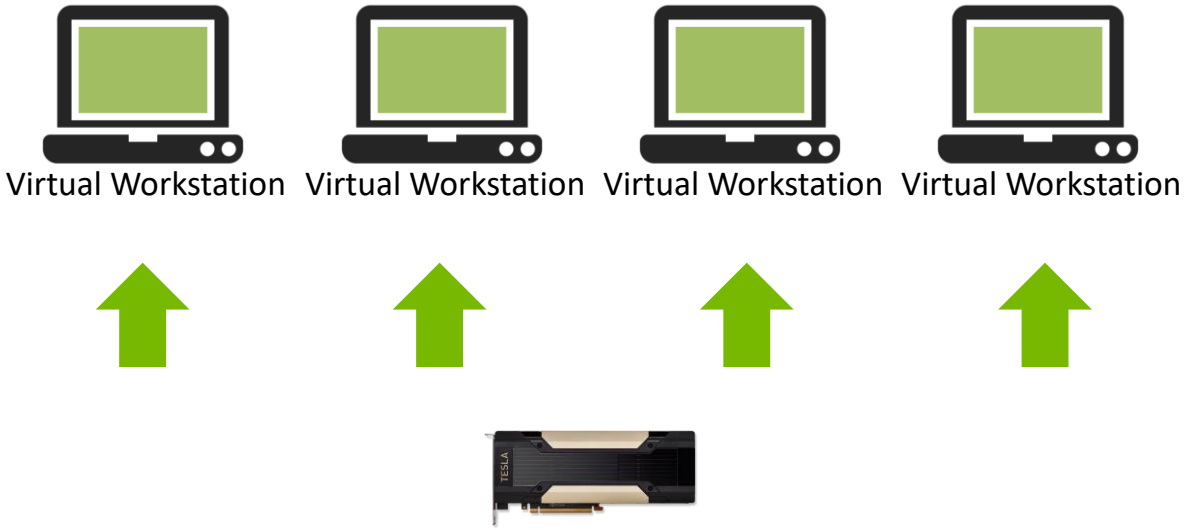
2013 2016 2017 2018 2019

Source: Source information is 8 pt, italic

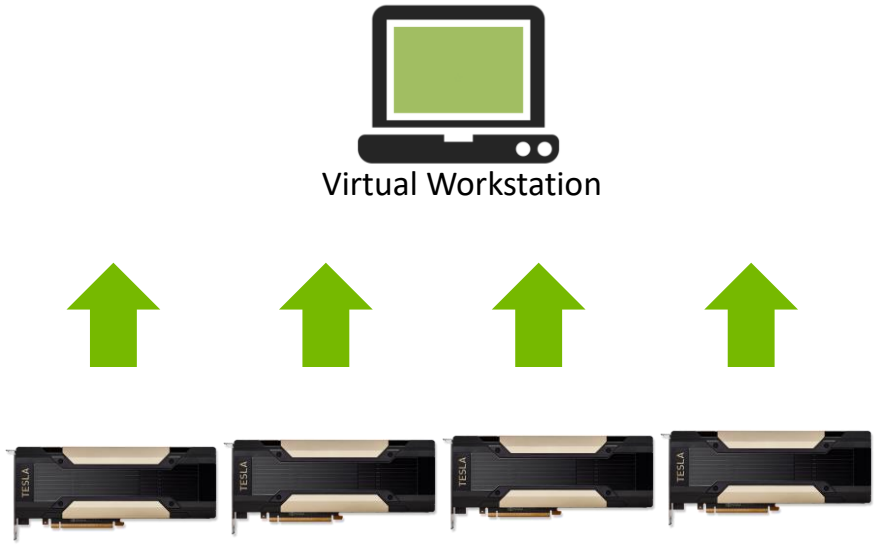
MULTI-vGPU

Delivering a More Powerful Virtual Workstation

Multiple VM's Sharing a GPU
with NVIDIA Quadro vDWS

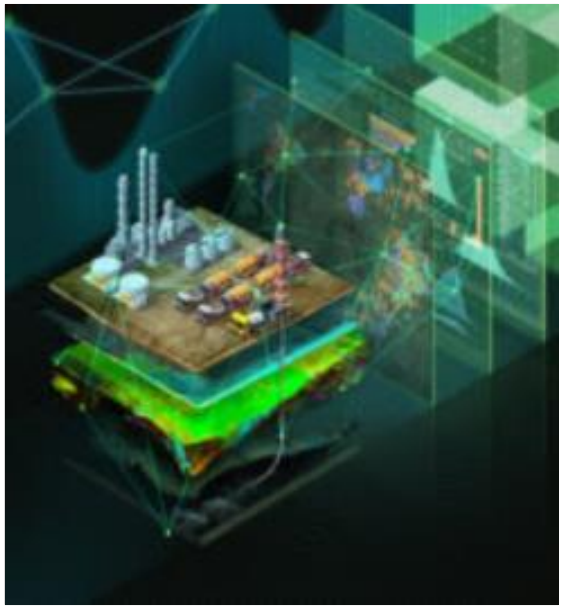


A Single VM Accessing Multiple Tesla GPUs
with NVIDIA Quadro vDWS (7.0)



QUADRO vDWS MULTI-vGPU

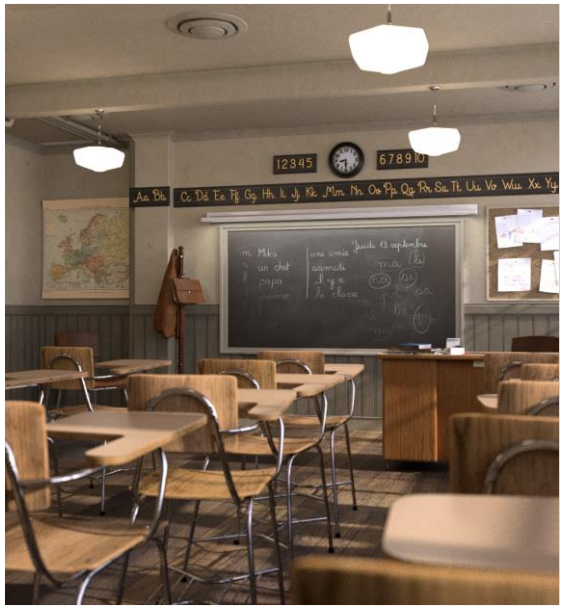
Enabling New Workflows in Strategic Markets



Oil & Gas
Seismic interpretation, simulation

Manufacturing
Simulation, modeling & design

Federal Government
Simulation & training

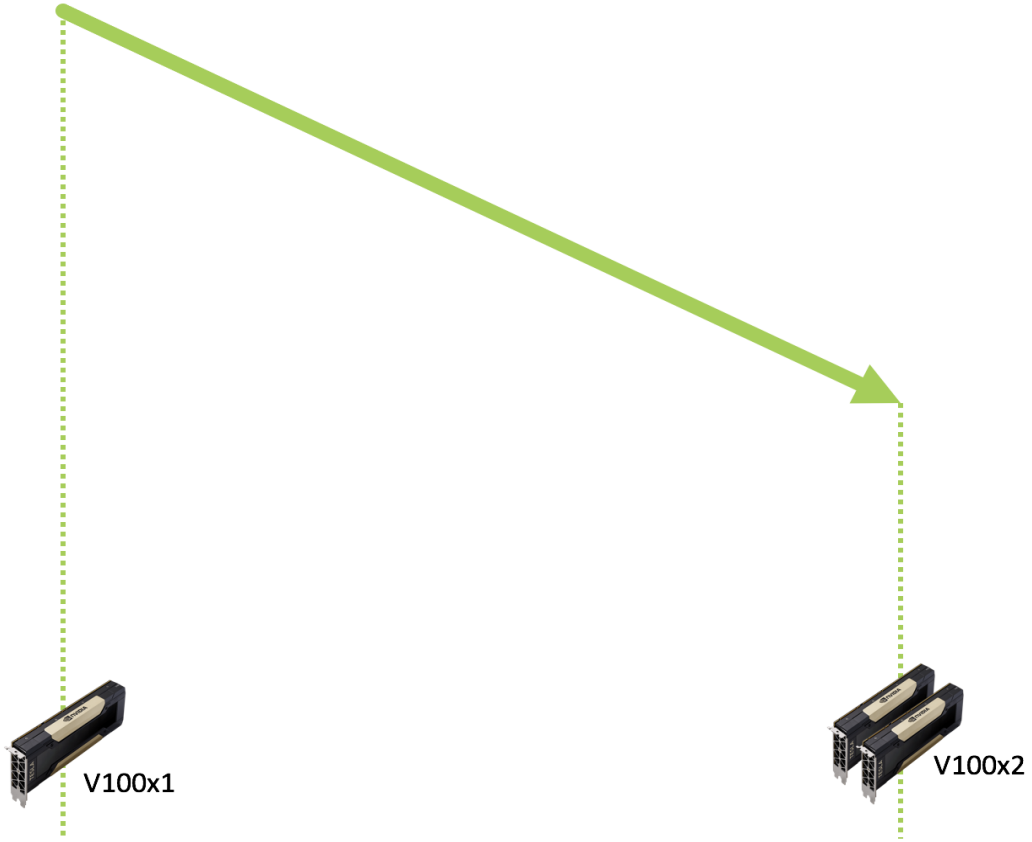


Media & Entertainment
Rendering

94% FASTER RENDERING USING MULTI-VGPU

SOLIDWORKS Visualize (Iray) Render Time

Up to **94%** faster render time using two Tesla V100-32Q GPUs versus a single Tesla V100-32Q

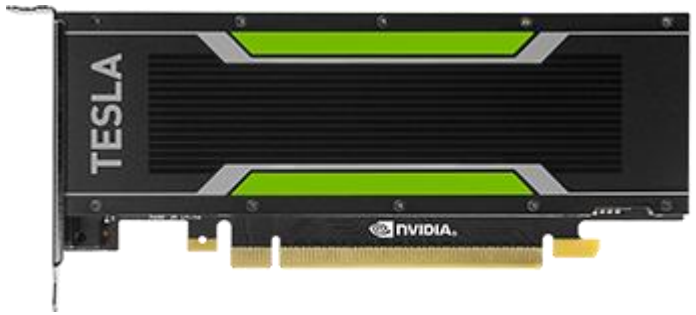


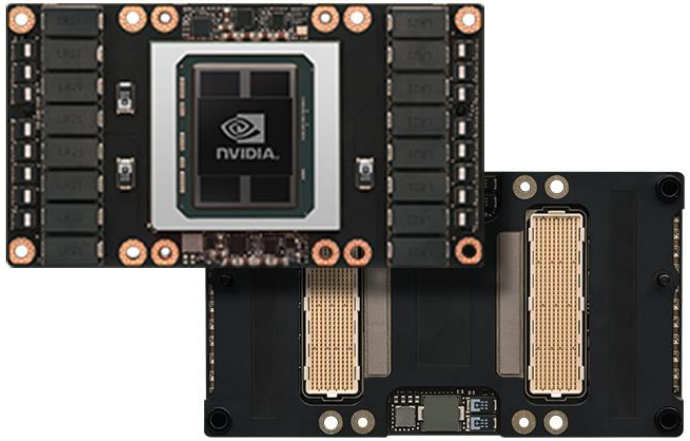
TAKEWAYS

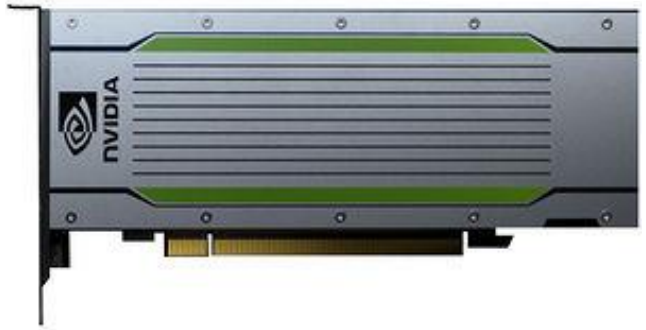
NVIDIA GRID VGPU

Runs on all Tesla GPUs

Maxwell
Pascal
Volta
Turing (Soon)







COMMITTED TO INNOVATION

Designers/Architects/Engineers

Business Users

Simulation/Photo Realism/3D Rendering

AI/DL

2013 | 2016 | 2017 | 2018 / 2019

Source: Source information is 8 pt, italic


NVIDIA®