vGPU-虚拟化GPU软件使用说明

文档版本记录

版本	日期	作者	描述
v1.0	2022-09-06	Yang Liu	vGPU使用说明(v14.0 through 14.2 - Last updated July 29, 2022)
v1.1	2022-11-13	Yang Liu	更新MIG相关内容和技术
v1.2	2022-11-29	Yang Liu	更新支持产品列表
v1.3	2022-12-1	Yang Liu	校正部分错误用词

• vGPU-虚拟化GPU软件使用说明

- 。概述
- 。1. NVIDIA vGPU 软件介绍
 - 1.1 vGPU 的使用方法
 - 1.1.1. NVIDIA vGPU
 - 1.1.2. GPU 直通
 - 1.1.3. 裸金属部署
 - 1.2. NVIDIA vGPU 软件部署时, 主要显示适配器要求
 - 1.3. NVIDIA vGPU 软件特性介绍
 - 1.3.1. NVIDIA GPU 实例对 vGPU 的支持
 - 1.3.2. vGPU 软件支持的API
 - 1.3.3. vGPU 对 CUDA 和 OpenCL的支持
 - 1.3.4. vWS的额外特性
 - 1.3.5. NGC Container对vGPU的支持
 - 1.3.6. NVIDIA GPU Operator 支持
- 。 2. 安装和配置NVIDIA虚拟GPU管理器
 - 2.1. 关于 Nvidia vGPU
 - 2.1.1. Nvidia vGPU 架构
 - 2.1.1.1. 基时间分片的NVIDIA vGPU内部架构
 - 2.1.1.2. 基于GPU实例 (MIG) 的NVIDIA vGPU内部架构
 - 2.1.2. vGPU的类型
 - 2.1.3. Q-series和B-series vGPU的虚拟显示分辨率
 - 2.1.4. 单个GPU上有效的vGPU配置
 - 2.1.4.1. 单个GPU上有效的基于时间分片的vGPU配置
 - 2.1.4.2. 单个GPU上有效的基于MIG的vGPU配置
 - 2.1.5. 虚拟机的支持
 - 2.1.5.1 Windows虚拟机支持
 - 2.1.5.2 Linux虚拟机支持
 - 2.2. 使用NVIDIA vGPU的前提条件
 - 2.3. 切换支持多种显示模式的GPU的模式
 - 2.4. 切换Tesla M60或M6 GPU的模式
 - 2.5. 在Citrix Hypervisor中安装配置NVIDIA Virtual GPU Manager
 - 2.6. 在Linux KVN中安装NVIDIA Virtual GPU Manager
 - 2.7. 在Red Hat Enterprise Linux KVM 或者 RHV中安装配置NVIDIA Virtual GPU Manager
 - 2.8. 在Ubuntu中安装配置NVIDIA Virtual GPU Manager
 - 2.9. 在VMware Vsphere中安装配置NVIDIA Virtual GPU Manager
 - 2.10. 在KVM Hypervisor下的Linux中配置NVIDIA Virtual GPU Manager
 - 2.11. 为基于MIG的vgpu配置GPU
 - 2.11.1. 开启GPU的MIG模式
 - 2.11.2. 在基于MIG的GPU上创建GPU实例
 - 2.11.3. 可选:在GPU实例中创建计算实例
 - 2.12. 禁用一个或多个gpu的MIG模式
 - 2.13. 启用或禁用ECC

- 2.13.1. 如何禁用ECC内存
- 2.13.2. 如何启动ECC内存
- 2.14 配置vGPU虚拟机,使用NVIDIA GPUDirect存储技术
- 。 3. GPU直通
 - 3.1. 物理GPU的显示分辨率
 - 3.2. 在Citrix Hypervisor配置GPU直通
 - 3.3. 在Red Hat Enterprise Linux KVM or Ubuntu配置GPU直通
 - 3.4. 在Microsoft Windows Server配置GPU直通
 - 3.5. 在VMware vSphere配置GPU直通
- 。 4. 安装NVIDIA vGPU Software Graphics Driver
 - 4.1. Installing the NVIDIA vGPU Software Graphics Driver on Windows
 - 4.2. Installing the NVIDIA vGPU Software Graphics Driver on Linux
- 。 5. 授权NVIDIA vGPU
 - 5.1. 配置NVIDIA License系统的授权客户端
 - 5.1.1. 在Windows下配置客户端
 - 5.1.2. 在Linux下配置客户端
 - 5.1.3. 验证已授权客户端的NVIDIA vGPU软件License状态
 - 5.2. 从许可证服务器授权NVIDIA vGPU
 - 5.2.1. 在Windows上授权NVIDIA vGPU
 - 5.2.2. 在Linux上授权NVIDIA vGPU
- 。 6.监控GPU性能
 - 6.1. NVIDIA System Management Interface nvidia-smi
 - 6.2. 从Hypervisor监控GPU性能
 - 6.2.1. 从Hypervisor使用nvidia-smi监控GPU Performance
 - 6.2.1.1. 得到系统中所有物理GPU的信息
 - 6.2.1.2. 得到系统中所有vGPU的信息
 - 6.2.1.3. 得到vGPU的详细信息
 - 6.2.1.4. 监控vGPU引擎使用情况
 - 6.2.1.5. 根据应用监控vGPU的使用情况
 - 6.2.1.6. 监控Encoder的使用情况
 - 6.2.1.7. 列出支持的vGPU类型
 - 6.2.1.8. 列出当前可以创建的vGPU类型
 - 6.2.2. 使用Citrix XenCenter监控GPU性能
 - 6.3. 从虚拟机监控GPU性能
 - 6.3.1. 使用nvidia-smi从客户虚拟机监控GPU性能
 - 6.3.2. 使用nWindows Performance Counters从客户虚拟机监控GPU性能
 - 6.3.3. 使用NVWMI从客户虚拟机监控GPU性能
- 。7. 改变时间切片vGPU的调度行为
 - 7.1. 时间切片vGPU的调度策略
 - 7.2 时间切片vGPU调度器的时间切片
 - 7.3. RmPVMRL 注册表键值
 - 7.4. 获取当前所有GPU的时间切片vGPU调度行为
 - 7.5. 改变所有GPU的时间切片vGPU调度行为
 - 7.6. 为特定GPU的时间切片改变vGPU调度行为
 - 7.7. 恢复默认的时间切片vGPU调度程序设置
- 。 8.vGPU支持的虚拟化平台和产品
 - Citrix 虚拟化支持
 - Microsoft Windows Server支持
 - Red Hat Enterprise Linux with KVM 支持
 - ubuntu 支持
 - VMwarea平台支持

。总结

概述

NVIDIA vGPU软件是提供虚拟机访问NVIDIA GPU技术的图形虚拟化平台。

1. NVIDIA vGPU 软件介绍

1.1 vGPU 的使用方法

1.1.1. NVIDIA vGPU

NVIDIA vGPU可以使得多个虚拟机(VM)可以同时、直连同一个物理GPU, 它们同时使用一个在操作系统上的图形化驱动。通过这种方式, NVIDIA vGPU为虚拟机提供了无与伦比的图形性能、计算性能和应用兼容性, 以及在多种工作负载下共享GPU带来的成本效益和可扩展性。

1.1.2. GPU 直通

GPU直通模式下,整个物理 GPU 直接分配给一台虚拟机,无需使用 NVIDIA vGPU 管理器。在该模式下,GPU被分配被在虚拟机中运行的 NVIDIA 驱动独占,虚拟机不能共享GPU。

1.1.3. 裸金属部署

在裸金属部署场景下,您可以使用 NVIDIA vGPU 软件图形驱动,配合 vWS 或者 vApps license,实现远程虚拟桌面和应用。如果您打算在没有虚拟平台的情况下,使用 Tesla GPU实现这样的功能,请使用 NVIDIA vGPU 驱动程序,而不是其他 NVIDIA 驱动程序。

如果要在裸金属的情况下,使用 NVIDIA vGPU 软件驱动,其完成下列几个步骤:

- 1. 在物理主机上安装 vGPU 驱动:请安装 NVIDIA vGPU 软件驱动,详细步骤见链接Installing the NVIDIA vGPU Software Graphics Driver.
- 2. 获得 vGPU 软件授权:授权 vGPU 的步骤,详见链接Virtual GPU Client Licensing User Guide.
- 3. 配置远程链接平台:要使用 Tesla GPU 图形特性,必须使用支持的远程解决方案,例如,RemoteFX,Citrix虚拟应用和桌面,VNC,或类似的技术。
- 4. Tesla GPU 设置为主显示: NVIDIA Tesla通常在裸金属平台上作为辅助显示设备运行。
- 5. **禁用非NVIDIA适配器连接的显示设备**:可以通过主机操作系统的显示设置或远程解决方案来实现。在包括 Tesla 在内的 NVIDIA GPU上,默认都是显示设备是开启的。只有启用 NVIDIA 适配器驱动的显示后,用户才能启动需要 NVIDIA GPU 技术的应用,以增强用户体验。

1.2. NVIDIA vGPU 软件部署时, 主要显示适配器要求

设置为主显示适配器的 GPU 不能用于 NVIDIA vGPU 部署和 GPU 直通部署。主要显示是虚拟化平台(hypervisor)主机的启动显示,其中显示SBIOS控制台 消息,然后显示操作系统或 hypervisor 的启动。

任何用于 NVIDIA vGPU 部署或 GPU 直通部署的GPU都必须设置为次要显示。

1.3. NVIDIA vGPU 软件特性介绍

NVIDIA vGPU软件包括 vWS、vCS、vPC和vApps 4种不同的软件功能授权(lience)。

1.3.1. NVIDIA GPU 实例对 vGPU 的支持

NVIDIA vGPU 软件在 NVIDIA vGPU 和 GPU 直通部署中支持 MIG (Multi-Instance GPU)特性的GPU实例。MIG可以将一个物理GPU安全划分为多个独立的 GPU实例,为多个用户提供独立的GPU资源,使其应用加速。除了提供MIG的所有好处外,NVIDIA vGPU 软件还增加了虚拟机安全性和工作负载管理。 Single Root I/O Virtualization (SR-IOV)虚拟功能对配置了 vGPU 的虚拟机实现完全 IOMMU 保护。

图一展示了一个物理GPU被划分为3个GPU实例,每个实例被匹配一个 vGPU。尽管每个 GPU 实例被 hypervisor 进行管理同时与一个 vGPU 匹配,但是每个 虚拟机可以在每个 GPU 实例的基础,再次进行对计算资源进行划分,而且在每个虚拟机中,运行多个容器程序。



NVIDIA vGPU软件支持MIG支持的单片vGPU,支持DEC、JPG和OFA。一个物理GPU上只能有一个支持DEC、JPG和OFA的MIG-Based vGPU。该实例可以 与一个单片实例放置在同一位置,不支持DEC、JPG和OFA。

并不是所有的虚拟化平台(hypervisor)都支持 MIG GPU实例的 vGPU 部署,详细信息请查看链接

NVIDIA vGPU软件仅支持NVIDIA虚拟计算服务器和Linux客户操作系统下的GPU实例。

如果需要支持NVIDIA vGPU的GPU实例,则需要配置开启 MIG 模式的GPU,并在物理 GPU 上创建和配置 GPU 实例。详细信息请查看链接

1.3.2. vGPU 软件支持的API

NVIDIA vGPU 支持一下的APIs:

- Open Computing Language (OpenCL[™] software) 3.0
- OpenGL® 4.6
- Vulkan® 1.3
- DirectX 11
- DirectX 12 (Windows 10)
- Direct2D
- DirectX Video Acceleration (DXVA)
- NVIDIA® CUDA® 11.6
- NVIDIA vGPU software SDK (remote graphics acceleration)
- NVIDIA RTX (on GPUs based on the NVIDIA Volta graphic architecture and later architectures)

1.3.3. vGPU 对 CUDA 和 OpenCL的支持

CUDA 和 OpenCL 只在部分 vGPU 类型和GPU 型号上支持

如果您在Linux上使用NVIDIA vGPU软件与CUDA,请避免从一个独立于发行版的运行文件包。不要从特定于发行版的RPM或Deb包安装CUDA。 为保证安装CUDA时不覆盖NVIDIA vGPU软件显卡驱动,在选择要安装的CUDA组件时,请取消选择安装CUDA驱动。 支持OpenCL和CUDA的NVIDIA vGPU类型如下:

- The 8Q vGPU type on Tesla M6, Tesla M10, and Tesla M60 GPUs
- All Q-series vGPU types on the following GPUs:
 - NVIDIA A2
 - NVIDIA A10
 - NVIDIA A16
 - NVIDIA A40
 - NVIDIA RTX A5000
 - NVIDIA RTX A5500
 - NVIDIA RTX A6000
 - Tesla P4
 - Tesla P6
 - Tesla P40

- Tesla P100 SXM2 16 GB
- Tesla P100 PCIe 16 GB
- Tesla P100 PCIe 12 GB
- Tesla V100 SXM2
- Tesla V100 SXM2 32GB
- Tesla V100 PCIe
- Tesla V100 PCIe 32GB
- Tesla V100S PCIe 32GB
- ∘ Tesla V100 FHHL
- ∘ Tesla T4
- Quadro RTX 6000
- Quadro RTX 6000 passive
- Quadro RTX 8000
- Quadro RTX 8000 passive
 - -All C-series vGPU types

NVIDIA vGPU在部分GPU上支持以下NVIDIA CUDA Toolkit开发工具:

- Debuggers:
 - CUDA-GDB
 - Compute Sanitizer
- Profilers:
 - The Activity, Callback, and Profiling APIs of the CUDA Profiling Tools Interface (CUPTI)
 - Other CUPTI APIs, such as the Event and Metric APIs, are not supported.
 - NVIDIA Nsight[™] Compute
 - NVIDIA Nsight Systems
 - NVIDIA Nsight plugin
 - NVIDIA Nsight Visual Studio plugin
 - Other CUDA profilers, such as nvprof and NVIDIA Visual Profiler, are not supported. 另外,这些工具只在 Linux 的虚拟机上支持。

注意:默认情况下,NVIDIA CUDA Toolkit开发工具在NVIDIA vGPU上是关闭的。如果需要使用NVIDIA CUDA Toolkit开发工具,则需要通过设置vGPU插件参数,为每个需要使用NVIDIA CUDA Toolkit开发工具的虚拟机单独启用。

下列出了NVIDIA vGPU 上支持这些调试器和分析器的GPU:

GPU	vGPU Mode	Debuggers	Profilers
NVIDIA A2	Time-sliced	\checkmark	1
NVIDIA A10	Time-sliced	√	1
NVIDIA A16	Time-sliced	~	1
NVIDIA A30	Time-sliced	\checkmark	1
	MIG-backed	\checkmark	1
NVIDIA A30X	Time-sliced	√	√
	MIG-backed	~	1
NVIDIA A40	Time-sliced	\checkmark	1
NVIDIA A100 HGX 40GB	Time-sliced	✓	1
	MIG-backed	\checkmark	1
NVIDIA A100 PCIe 40GB	Time-sliced	~	1
	MIG-backed	\checkmark	1
NVIDIA A100 HGX 80GB	Time-sliced	√ 	1
	MIG-backed	\checkmark	1
NVIDIA A100 PCIe 80GB	Time-sliced	√ 	1
	MIG-backed	\checkmark	1
NVIDIA A100X	Time-sliced	~	1
	MIG-backed	\checkmark	1
NVIDIA RTX A5000	Time-sliced	√ 	1
NVIDIA RTX A5500	Time-sliced	~	1
NVIDIA RTX A6000	Time-sliced	√ 	1
Tesla T4	Time-sliced	~	1
Quadro RTX 6000	Time-sliced	√ 	1
Quadro RTX 6000 passive	Time-sliced	√	√
Quadro RTX 8000	Time-sliced	✓	1
Quadro RTX 8000 passive	Time-sliced	\checkmark	1
Tesla V100 SXM2	Time-sliced	1	1
Tesla V100 SXM2 32GB	Time-sliced	\checkmark	√
Tesla V100 PCIe	Time-sliced	1	1
Tesla V100 PCIe 32GB	Time-sliced	1	√
Tesla V100S PCIe 32GB	Time-sliced	1	1
Tesla V100 FHHL	Time-sliced	\checkmark	√

NVIDIA vGPU 支持的CUDA特性:

NVIDIA vGPU在vGPU类型、物理GPU以及虚拟化环境软件版本支持特性的情况下,支持以下NVIDIA CUDA Toolkit特性:

- Error-correcting code (ECC) memory
- Peer-to-peer CUDA transfers over NVLink 注意事项:当需要确定主机物理图形处理器与虚拟机分配的vgpu之间的NVLink拓扑时,可在主机或虚拟机上执行以下命令:

\$ nvidia-smi topo -m

- NVIDIA GPUDirect® Remote Direct Memory Access (RDMA) technology
- GPUDirect Storage technology
- Unified Memory
- 注:默认关闭统一内存。如果使用统一内存,则需要通过设置vGPU插件参数,为需要统一内存的每个vGPU单独开启统一内存。具体操作请参见链接。
- NVIDIA Nsight Systems GPU context switch trace

支持ECC内存的GPU上所有vGPU类型都支持动态页面退出,即使物理GPU关闭了ECC内存。

NVIDIA vGPU不支持的NVIDIA CUDA工具包特性: NVIDIA vGPU不支持NVIDIA CUDA工具包中的NVIDIA Nsight图形功能。 注意:NVIDIA Nsight图形特性支持GPU直通模式和裸金属部署。

1.3.4. vWS的额外特性

在vPC、vAPP特性的基础上,vWS还提供以下特性:

- 针对工作站特定的图形特性和加速特性
- 专业应用的认证驱动
- 用于工作站或专业3D图形的GPU直通部署
- 在直通模式下,vWS支持分辨率高达8K的多个虚拟显示端口,并根据可用像素的数量灵活的虚拟显示分辨率。详细信息请参见物理GPU的显示分辨率。
- Windows用户可最高使用10 bit 颜色。(HDR/10位颜色目前不支持Linux,支持NvFBC捕获,但已弃用。)

1.3.5. NGC Container对vGPU的支持

NVIDIA vGPU软件支持NGC容器,在NVIDIA vGPU和GPU直通类型下,部署在所有支持的hypervisor上。 在NVIDIA vGPU部署中,以下vGPU类型仅支持在Maxwell架构之后的NVIDIA GPU架构的图形处理器上:

- Q 系列 vGPU 许可
- C 系列 vGPU 许可

在GPU直通部署中,支持NGC Container,在 NVIDIA Maxwell™架构之后的NVIDIA GPU架构,同时支持NVIDIA vGPU软件的GPU上使用 支持Linux的操作系统

关于设置NVIDIA vGPU软件用于NGC容器的更多信息,请参见使用NGC与NVIDIA虚拟GPU软件设置指南。

1.3.6. NVIDIA GPU Operator 支持

NVIDIA GPU Operatpr简化了NVIDIA vGPU软件在Kubernetes容器编制引擎管理平台上的部署。NVIDIA GPU Operatpr为在安装了NVIDIA vGPU的虚拟机上 运行的容器平台提供了NVIDIA vGPU 软件图形驱动程序的自动化安装和更新。

NVIDIA GPU Operatpr通过检查NVIDIA vGPU软件图形驱动程序一起发布的驱动程序目录,自动确定与平台的虚拟GPU管理器兼容的NVIDIA vGPU软件图形 驱动程序版本。

NVIDIA GPU Operatpr简化了NVIDIA安装的任何驱动程序必须从NVIDIA授权门户下载到本地计算机。不支持NVIDIA GPU Operatpr自动访问NVIDIA授权门 户。

NVIDIA GPU Operator支持NVIDIA vGPU软件的自动化配置,并通过在虚拟机VM中运行的DCGM Exporter 提供自动探测支持。

NVIDIA GPU Operator仅在特定的Hypervisor、容器平台、vGPU类型和操作系统组合下支持。要确定您的配置是否支持NVIDIA GPU Operator,请参考你所 选择的Hypervisor。

更多信息,请参见NVIDIA GPU Operator概述。

2. 安装和配置NVIDIA虚拟GPU管理器

2.1. 关于 Nvidia vGPU

2.1.1. Nvidia vGPU 架构

NVIDIA vGPU的架构如下图所示。在hypervisor中运行的NVIDIA Virtual GPU Manager的控制下,NVIDIA物理GPU能够支持多个可以分配给客户虚拟机的 vGPU。



客户虚拟机使用NVIDIA vGPU的方式与物理GPU经过hypervisor直通方式相同:客户虚拟机中加载的NVIDIA驱动程序提供直接访问物理GPU的性能关键快速路 径,NVIDIA虚拟GPU管理器的半虚拟化接口用于非性能管理操作。

每个NVIDIA vGPU类似于传统GPU,具有大小的GPU framebuffer和一个或多个虚拟显示输出。在创建vGPU时,vGPU的framebuffer是从物理GPU的 framebuffer中分配的,并且vGPU保留对该framebuffer的独占使用,直到它被销毁。

根据物理GPU的不同,可以在vGPU上创建不同类型的vGPU:

- 在所有支持NVIDIA vGPU软件的gpu上,可以创建时间片vGPU。
- •此外,在支持多实例GPU (MIG)特性的GPU上,可以创建 MIG 支持的vGPU。MIG特性在基于NVIDIA Ampere 架构的GPU上的实现。

2.1.1.1. 基时间分片的NVIDIA vGPU内部架构

时间片vGPU是驻留在未划分为多个GPU实例的物理GPU上的vGPU。驻留在GPU上的所有时间片vgpu共享对GPU引擎的访问,包括图形(3D)、视频解码和视频编码引擎。如图所示:



在时间片vGPU中,运行在vGPU上的进程被调度为串行运行。当其他进程在其他vGPU上运行时,每个vGPU都会等待。当进程在vGPU上运行时,vGPU独占 GPU的引擎。可以修改默认的调度行为,具体操作请参见后续章节"更改时间片vGPU的调度行为"。

2.1.1.2. 基于GPU实例 (MIG) 的NVIDIA vGPU内部架构

支持MIG的vGPU是驻留在支持MIG的物理GPU中的GPU实例上的vGPU。每个驻留在GPU实例上的vGPU独占访问GPU实例的引擎,包括计算和视频解码引擎。

在MIG支持的vGPU中,运行在vGPU上的进程与运行在该GPU上的其他vGPU上的进程并行运行。进程同时在一个物理GPU上的所有vGPU上运行。



2.1.2. vGPU的类型

一块主板上的物理GPU数量取决于该板。每个物理GPU可以支持几种不同类型的虚拟GPU (vGPU)。vGPU类型具有固定数量的帧缓冲区、支持的显示输出数 量和最大分辨率。根据不同的工作负载,它们被分为不同的系列。每个系列由vGPU类型名称的最后一个字母标识。

- Q-series:为需要Quadro技术的性能和功能的创意和技术专业人员提供的虚拟工作站
- C-series: 计算密集型服务器工作负载,如人工智能、深度学习或高性能计算。
- B-series: 为业务专业人员和知识工作者提供的虚拟桌面
- A-Series: 为虚拟应用的用户,提供应用流或基于会话的解决方案

在vGPU类型名称中,GPU卡产品名后面的数字表示分配给该类型的vGPU的帧缓冲区的数量。例如,A16-4C是在NVIDIA A16板上分配了4096 mb的帧缓冲区的vGPU。

不同类型的vGPU对资源的需求不同,同一物理GPU上可以同时创建的vGPU的最大数量也不同。 当启用时,帧率限制器(FRL)限制vGPU每秒帧数(FPS)的最大帧速率,如下所示:

- 对于B-series vGPU, 最大帧率为45FPS。
- 对于A-series, C-series, Q-series 最大帧率为60FPS。

默认情况下,所有 GPU 都开启FRL。在支持替代vGPU调度器的GPU上,当vGPU调度行为从默认的调度器更改时,将禁用FRL。具体操作请参见"更改时间片 vGPU的调度行为"。在使用默认调度器的vGPU上,可以按照NVIDIA Virtual GPU软件文档中所述,在所选择的虚拟机Hypervisor中禁用FRL。

注意:

NVIDIA vGPU在其所有支持的GPU上是需要授权的产品。在客户虚拟机中启用所有vGPU特性时,需要获得软件许可证。license类型取决于vGPU类型:

- Q-series vGPU类型需要使用vWS license。
- C-series vGPU类型需要NVIDIA Virtual Compute Server (vCS) license, 但也可以使用vWS license。
- B-series vGPU类型需要vPC license,但也可以用vWS license使用。
- A-Series vGPU类型需要vApps license。

要了解每个支持的GPU提供的vGPU类型的详细信息,请查看链接Virtual GPU Types for Supported GPUs

2.1.3. Q-series和B-series vGPU的虚拟显示分辨率

Q-series和B-series vGPU支持基于可用像素数量的最大组合分辨率,而不是固定的每个显示器的最大分辨率,这是由它们的帧缓冲区大小决定的。你可以选 择使用少量的高分辨率显示器,或者使用大量的低分辨率显示器。

可以使用的虚拟显示的数量取决于以下因素的组合:

- vGPU类型
- GPU架构
- vGPU帧缓冲区的大小
- 显示分辨率

注意:

即使显示的组合分辨率小于vGPU的可用像素数,您也不能使用超过vGPU支持的最大显示数。例如,因为-0Q 和 -0B vGPU最多只支持两个显示器,所以不能 在这些vgGPU类型上使用四个1280×1024显示器,即使显示器的组合分辨率(6220800)小于这些vGPU的可用像素数(8192000)。

各种因素影响GPU帧缓存的消耗,从而影响用户体验。这些因素包括(但不限于)显示器的数量、显示分辨率、部署的工作负载和部署的程序、远程解决方案和 客户操作系统等。vGPU驱动显示组合的能力并不保证有足够的帧缓冲区空闲供所有应用程序运行。如果应用程序的帧缓冲区耗尽,请考虑使用以下方式之一 更改您的设置:

- 切换到具有更多帧缓冲区的vGPU类型
- 使用更少的显示器
- 使用低分辨率显示器

Virtual GPU Types for Supported GPUs中所列出的每个vGPU的最大显示数是基于所有显示具有相同分辨率的配置。有关混合显示分辨率的配置示例,请参见 B-series和Q-series vGPU的混合显示配置。

2.1.4. 单个GPU上有效的vGPU配置

单个GPU上有效的vGPU配置取决于vGPU是时间分片的,还是基于MIG的GPU实例。

2.1.4.1. 单个GPU上有效的基于时间分片的vGPU配置

此版本的NVIDIA vGPU支持同一物理GPU上不同时间片vGPU类型具有相同数量的帧缓存。例如,具有相同帧缓存数量的A-series, C-series, Q-series和Bseries vgpu可以同时驻留在同一物理GPU上。 然而,要求所有vGPU具有相同数量的帧缓存并不会扩展到同一主板上的所有物理gpu。同一主板上的不同物理GPU可以同时托管具有不同数量帧缓冲区的虚 拟GPU,只需要保证任意单独一个物理GPU上的vGPU类型具有相同的帧缓冲区, 例如,一个NVIDIA A16主板有四个物理GPU,可以支持多种类型的虚拟GPU:

- 在GPU0上混合使用A16-4C vGPU和A16-4Q vGPU是有效的。
- 在GPU0和GPU1上使用A16-16C vGPU,在GPU2上使用A16-8C vGPU,在GPU3上使用A16-4C vGPU的配置是有效的。
- 在GPU0上混合使用A16-8C和A16-4C vGPU的配置无效。

并不是所有的Hypervisor都支持同一物理GPU上不同vGPU类型具有相同帧缓存数量的时间片vGPU。要确定您选择的Hypervisor是否支持此功能,请参阅 NVIDIA Virtual GPU Software Documentation中有关虚拟机管理程序的发行说明。

2.1.4.2. 单个GPU上有效的基于MIG的vGPU配置

此版本的NVIDIA vGPU支持基于MIG的GPU实例配置的同质和混合的vGPU。例如,NVIDIA A100 PCIe 40GB卡有一个物理GPU,可以支持多种类型的虚拟 GPU。图5显示了在NVIDIA A100 PCIe 40GB上有效的同质和混合基于MIG的vGPU配置的以下示例。

- 一个有效的同质配置,在3个MIG 2g.10b GPU实例上分别运行3个A100-2-10C vGPU
- 一个有效的同构配置,在2个MIG 3g.20b GPU实例上分别运行2个A100-3-20C vGPU
- 一个有效的混合配置,在1个MIG 4g.20b GPU实例上运行A100-4-20C vGPU,在MIG 2.10b GPU实例上运行1个A100-2-10C vGPU,在MIG 1g.5b GPU 实例上运行一个A100-1-5C vGPU.

NVIDIA A100 PCIe 40GB							
Physical GPU 0							
Valid homogeneous configuration with 3 A100-2-10C vGPUs on 3 MIG.2g.10b GPU instances							
A100-2-10C on MIG.2g.10b	A100-2-10 MIG.2g.1)C on 10b	n A100-2-10C on MIG.2g.10b				
Valid homogeneous configuration	with 2 A100-3-20C v	GPUs on 3 MIG	.3g.20b GPU instances				
A100-3-20C on MIG.3g.20b			A100-3-20C on MIG.3g.20b				
Valid mixed configuration with 1 A100-4-20C vGPU on a MIG.4g.20b GPU instance, 1 A100-2-10C vGPU on a MIG.2.10b GPU instance, and 1 A100-1-5C vGPU on a MIG.1g.5b instance							
A100-4-20	A100-4-20C on A100-2-10C on A100-1-5C on UIC 4 50						

2.1.5. 虚拟机的支持

NVIDIA vGPU支持Windows和Linux客户虚拟机操作系统。不同操作系统支持的vGPU类型不同。

有关支持的Windows和Linux版本的详细信息,以及有关支持的配置的更多信息,请参阅NVIDIA Virtual GPU Documentation

2.1.5.1 Windows虚拟机支持

Windows客户虚拟机仅支持 Q-series、B-series、 A-series的NVIDIA vGPU类型。 C-series的NVIDIA vGPU类型不支持。

2.1.5.2 Linux虚拟机支持

64位Linux客户虚拟机仅支持 Q-series、C-series、B-series的NVIDIA vGPU类型。A-series的NVIDIA vGPU类型不支持。

2.2. 使用NVIDIA vGPU的前提条件

在使用vGPU之前,请确保满足以下先决条件:

- 有一个服务器,支持的虚拟机的Hypervisor和支持NVIDIA vGPU软件的NVIDIA GPU。
- 一个或多个支持NVIDIA vGPU软件的NVIDIA GPU安装在您的服务器上。
- 如果您使用基于NVIDIA Ampere架构的GPU,则在您的服务器平台上启用以下BIOS设置:
 - ∘ VT-D / IOMMU的
 - SR-IOV
- 为选择的虚拟机Hypervisor下载了NVIDIA vGPU软件包,该软件包由以下软件组成:
 - 。 NVIDIA Virtual GPU Manager用于您的管理程序
 - 。 客户操作系统支持的NVIDIA vGPU软件图形驱动程序
- 请按照x虚拟化软件厂商的文档说明安装以下软件:
 - 。选择Hypervisor, 例如Citrix hypervisor、Red Hat Enterprise Linux KVM、Red Hat Virtualization (RHV)或VMware vSphere hypervisor (ESXi)
 - 。管理选择的Hypervisor的软件,例如Citrix xcenter管理GUI或VMware vCenter Server
 - 。 您将在运行NVIDIA virtual GPU的虚拟机中使用的虚拟桌面软件,例如Citrix虚拟应用程序和桌面或VMware Horizon

注意:如果您使用的是VMware vSphere Hypervisor (ESXi),请确保将配置NVIDIA vGPU虚拟机的ESXi主机不是完全自动化的VMware Distributed Resource Scheduler (DRS)集群的成员。有关更多信息,请参见 Installing and Configuring the NVIDIA Virtual GPU Manager for VMware vSphere。

- 完成虚拟机的创建
- 注意:本指南中涉及的所有Hypervisor都支持在一个虚拟机上使用多个vGPU。
- 虚拟机已安装指定的操作系统。

有关支持的硬件和软件的信息,以及此版本的NVIDIA vGPU软件的任何已知问题,请参阅您选择的虚拟机管理程序的发行说明:

- Virtual GPU Software for Citrix Hypervisor Release Notes
- Virtual GPU Software for Red Hat Enterprise Linux with KVM Release Notes
- Virtual GPU Software for Ubuntu Release Notes
- Virtual GPU Software for VMware vSphere Release Notes

2.3. 切换支持多种显示模式的GPU的模式

一些GPU支持无显示和启用显示2种模式,但必须在NVIDIA vGPU软件部署中以无显示模式使用。

一下列出的GPU支持多种显示方式。所下列所示,有些GPU是出厂时以无显示模式提供的,而其他GPUY则是以显示模式提供的。

- NVIDIA A40 无显示模式
- NVIDIA RTX A5000 启用显示模式
- NVIDIA RTX A5500 启用显示模式
- NVIDIA RTX A6000 启用显示模式

工厂以无显示模式提供的GPU(如NVIDIA A40 GPU),如果其模式此前已更改,则可能处于启用显示模式。

要更改支持多种显示模式的GPU的模式,请使用 displaymodeselector 工具,您可以从NVIDIA Developer网站上的NVIDIA displaymode Selector页面下载该工 具。

注意:

只有以下gpu支持 displaymodeselector 工具:

- NVIDIA A40
- NVIDIA RTX A5000
- NVIDIA RTX A5500
- NVIDIA RTX A6000

其他支持NVIDIA vGPU软件的gpu不支持 displaymodeselector 工具,除非另有说明,否则不需要切换显示模式。

2.4. 切换Tesla M60或M6 GPU的模式

Tesla M60和M6 GPU支持计算模式和图形模式。对于支持两种模式的GPU, NVIDIA vGPU需要在图形模式下才能运行。

最近的Tesla M60和M6 GPU都以图形模式提供。但是,如果您的GPU是较旧的Tesla M60 GPU或M6 GPU,或者它的模式以前已更改,则可能处于计算模式。

配置Tesla M60和M6的GPU模式,请使用NVIDIA vGPU软件版本提供的 gpumodeswitch 工具。如果您不确定您的GPU处于哪种模式,请使用gpumodeswitch工具来查找模式。

注意:

仅Tesla M60和M6两款GPU支持 gpumodeswitch 工具。其他支持NVIDIA vGPU的GPU不支持 gpumodeswitch 工具,除非在切换支持多种显示模式的GPU的模式 中说明,否则不需要模式切换。

即使在计算模式下, Tesla M60和M6 GPU也不支持C-series的vGPU类型。此外, Citrix Hypervisor上的任何GPU都不支持vCS。

更多信息请参考gpumodeswitch用户指南。

2.5. 在Citrix Hypervisor中安装配置NVIDIA Virtual GPU Manager

2.6. 在Linux KVN中安装NVIDIA Virtual GPU Manager

- 2.7. 在Red Hat Enterprise Linux KVM 或者 RHV中安装配置NVIDIA Virtual GPU Manager
- 2.8. 在Ubuntu中安装配置NVIDIA Virtual GPU Manager

2.9. 在VMware Vsphere中安装配置NVIDIA Virtual GPU Manager

2.10. 在KVM Hypervisor下的Linux中配置NVIDIA Virtual GPU Manager

2.11. 为基于MIG的vgpu配置GPU

为了支持NVIDIA vGPU的GPU实例,必须配置GPU启用MIG模式,并且必须在物理GPU上创建和配置GPU实例。您可以在GPU实例中创建计算实例。如果您 没有在GPU实例中创建计算实例,则可以稍后从客户虚拟机中为单个vgpu添加计算实例。

使用前提条件如下:

- 虚拟化主机Hypervisor上已安装NVIDIA GPU管理器。
- 在Hypervisor主机上拥有root用户权限。
- 已经确定了哪些GPU实例对应于将要创建的基于MIG vGPU的vGPU类型。要获取此信息,请查询您的GPU的MIG支持的vgpu表。
- GPU没有被任何其他进程使用,例如CUDA应用程序、监控应用程序或nvidia-smi命令。

要为基于MIG的vGPU配置GPU,请遵循以下步骤:

• 开启GPU的MIG模式

注意:VMware vSphere自带GPU实例,启动虚拟机并安装客户驱动程序后,会自动在虚拟机上创建一个计算实例,因此只需要开启MIG模式即可。

- 在支持MIG的GPU上创建GPU实例。
- 可选:在GPU实例中创建计算实例。

为基于MIG的vGPU配置GPU后,需要创建所需的vGPU,并将其添加到虚拟机中。

2.11.1. 开启GPU的MIG模式

在Hypervisor命令shell中执行此任务。

- 以root用户身份在Hypervisor主机上打开一个shell。在所有支持的管理程序中,您可以使用SSH (secure shell)来实现此目的。各个Hypervisor可以提供额 外的登录方法。有关详细信息,请参阅您的虚拟机Hypervisor的文档。
- 确定是否启用MIG模式。使用nvidia-smi命令进行配置。缺省情况下,未启用MIG模式。 这个例子表明MIG模式在GPU 0上是禁用的。

```
$ nvidia-smi -i 0
```

```
NVIDIA-SMI 510.85.03 Driver Version: 510.85.03 CUDA Version: 11.6
                                        |-----
   | GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
               _____
                           MIG M. |
0 A100-SXM4-40GB On | 00000000:36:00.0 Off |
                                     0 |
| N/A 29C P0 62W / 400W | 0MiB / 40537MiB | 6% Default |
               1
                                  Disabled |
```

• 如果未启用MIG模式,请启用MIG模式。

```
nvidia-smi -i [gpu-ids] -mig 1
```

gpu-ids:

一个逗号分隔的GPU索引、PCI总线id或uuid列表,指定要启用MIG模式的GPU。如果不指定gpu-ids,则表示系统中所有GPU开启MIG模式。 启动MIG:

\$ nvidia-smi -i 0 -mig 1 Enabled MIG Mode for GPU 0000000:36:00.0 All done.

注意:如果GPU正在被其他进程使用,此命令将失败,并显示一个警告信息,说明该GPU的MIG模式处于pending enable状态。在这种情况下,停止所有正在使用GPU的进程,然后重试该命令。

- 仅VMware vSphere ESXi:重启虚拟化主机Hypervisor。
- 查询已开启MIG模式的GPU,确认已开启MIG模式。这个例子向GPU 0查询(comma-separated values, CSV)格式的PCI总线ID和MIG模式。

\$ nvidia-smi -i 0 --query-gpu=pci.bus_id,mig.mode.current --format=csv pci.bus_id, mig.mode.current 00000000:36:00.0, Enabled

2.11.2. 在基于MIG的GPU上创建GPU实例

注意:如果您使用的是VMware vSphere,请忽略此任务。VMware vSphere自动创建GPU实例。

在Hypervisor命令shell中执行此任务。

- 如果需要,以root用户身份在Hypervisor主机上打开一个命令shell。
- 列出GPU上可用的GPU实例配置文件。在创建配置文件时,您需要通过它们的id而不是名称来指定它们。

```
$ nvidia-smi mig -lgip
+------
| GPU instance profiles:
GPU Name ID Instances Memory P2P SM DEC ENC |
         Free/Total GiB CE JPEG OFA |
|-----|
0 MIG 1g.5gb 19 7/7 4.95 No 14 0 0
                     1 0 0 |
+-----
| 0 MIG 2g.10gb 14 3/3 9.90 No 28 1 0 |
                     2
                       0
                         0 |
0 MIG 3g.20gb 9 2/2
             19.79 No 42 2 0
                     3 0
                         0 |
+-----+
 0 MIG 4g.20gb 5 1/1 19.79
                      2
1
                 No 56
                         0
                           4
                       0
                         0 |
+-----
| 0 MIG 7g.40gb 0 1/1 39.59 No 98 5 0 |
                     7 1 1 |
+------
```

• 创建与所需的基于MIG的vGPU类型对应的GPU实例。

```
$ nvidia-smi mig -cgi gpu-instance-profile-ids
```

gpu-instance-profile-ids:

一个逗号分隔的GPU实例配置文件id列表,指定要创建的GPU实例。 这个例子创建了两个类型为2g.10gb的GPU实例,配置文件ID为14。

```
$ nvidia-smi mig -cgi 14,14
Successfully created GPU instance ID 5 on GPU 2 using profile MIG 2g.10gb (ID 14)
Successfully created GPU instance ID 3 on GPU 2 using profile MIG 2g.10gb (ID 14)
```

2.11.3. 可选:在GPU实例中创建计算实例

在GPU实例中创建计算实例是可选的。如果您没有在GPU实例中创建计算实例,则可以稍后从客户虚拟机中为单个vgpu添加计算实例。 注意:如果您使用的是VMware vSphere,请忽略此任务。虚拟机启动并安装客户驱动后,会自动在虚拟机上创建一个计算实例。 在hypervisor命令shell中执行此任务。

- 如果需要,以root用户身份在Hypervisor主机上打开一个命令shell。
- 列出可用的GPU实例。

```
$ nvidia-smi mig -lgi
+----+
| GPU instances: |
| GPU Name Profile Instance Placement |
| ID ID Start:Size |
|-------|
| 2 MIG 2g.10gb 14 3 0:2 |
+----+
| 2 MIG 2g.10gb 14 5 4:2 |
+----+
```

• 在每个GPU实例中创建所需的计算实例。

```
$ nvidia-smi mig -cci -gi gpu-instance-ids
```

gpu-instance-ids:一个逗号分隔的GPU实例id列表,指定要在其中创建计算实例的GPU实例。

警告

为避免客户虚拟机与虚拟化主机Hypervisor状态不一致,请不要在虚拟机运行的GPU实例上创建虚拟化环境下的计算实例。相反,从客户VM中创建计算 实例,如修改基于MIG的vGPU的配置中所述。 \$ nvidia-smi mig -cci -gi 3,5 Successfully created compute instance on GPU 0 GPU instance ID 1 using profile ID 2 Successfully created compute instance on GPU 0 GPU instance ID 2 using profile ID 2

• 验证计算实例是在每个GPU实例中创建的

\$ nvid	ia-sm	i														
+																+
MIG	devic	es:														I
+				+				+			-+					+
GPU	GI	CI	MIG	I		Memo	ory-l	Jsage		Vo	1			Share	d	
1	ID	ID	Dev	1		BA	4R1-U	Jsage	SM	Un	c C	E	ENC	DEC	OFA	JPG
1				1						EC	C					
		===		+==		===:		=====+			=+==			=====		====
2	3	0	0	L	ØMi	в/	998	84MiB	28	0	1	2	0	1	0	0
Í.				İ.	ØMi	в/	1638	B3MiB			Ì.					I.
+				+				4			-+					+
2	5	0	1	L	ØMi	в/	998	84MiB	28	e	1	2	0	1	0	01
1 -	-	-	-	i	0Mi	- , R /	1638	RAMIR		-	i	-	-	-	-	
					0111	.,	1050	. 01110								
+				+				4			-+					+
+																+
Proc	esses	:														
GPU	GI		CI		PID	Тур	be	Proces	s name	9				GP	U Mem	ory
1	ID)	ID											Us	age	1
				===		===:						==				====

注意:当虚拟机关闭或重启时,在虚拟机上创建的额外计算实例将被销毁。关机或重启后,虚拟机中只保留一个计算实例。安装NVIDIA vGPU软件图形驱动程序后,自动创建该计算实例。

2.12. 禁用一个或多个gpu的MIG模式

如果要用于基于时间片vGPU或GPU直通的GPU之前,已为基于MIG的vGPU配置过,请禁用该GPU的MIG模式。 禁用步骤,请参见链接

2.13. 启用或禁用ECC

支持NVIDIA vGPU软件的一些GPU支持使用NVIDIA vGPU的ECC (error correcting code)内存。ECC内存通过检测和处理双比特错误来提高数据的完整性。然而,并不是所有的GPU、vGPU类型和Hypervisor软件版本都支持NVIDIA vGPU的ECC内存。

在支持NVIDIA vGPU ECC内存的GPU上, C-series和Q-series系列的vGPU支持ECC内存,但A-series和B-series的vGPU不支持。虽然A-series和B-series的 vGPU是在启用了ECC内存的物理GPU上启动的,但在不支持ECC的vGPU上启用ECC会带来一些开销。

在没有HBM2内存的物理GPU上,可以被vGPU使用的帧缓存数量减少。所有类型的vGPU都会受到影响,而不仅仅是支持ECC内存的vGPU。

在物理GPU上启用ECC内存的效果如下:

在物理GPU上所有支持的vGPU上, ECC内存都是一个特性。

- 支持ECC内存的虚拟机是启用ECC内存的,在虚拟机中有关闭ECC的选项。
- 支持开启或关闭单个虚拟机的ECC内存。开启或关闭虚拟机的ECC内存不会影响vGPU可用的帧缓存大小。
- •基于Pascal和之后的GPU架构的GPU支持NVIDIA vGPU的ECC内存。GPU执行nvidia-smi -q命令,查看是否开启了ECC内存。

Tesla M60和M6 GPU在不使用GPU虚拟化的情况下支持ECC内存,而NVIDIA vGPU在这些GPU上不支持ECC内存。在图形模式下,这些GPU默认提供禁用的ECC内存。

一些Hypervisor软件版本不支持NVIDIA vGPU的ECC内存。

如果您使用的Hypervisor软件版本或GPU,不支持NVIDIA vGPU中使用ECC内存,但是ECC内存却处于启用状态,NVIDIA vGPU无法启动。在这种情况下,如 果您要使用NVIDIA vGPU,则必须确保在所有GPU上禁用ECC内存。

2.13.1. 如何禁用ECC内存

2.13.2. 如何启动ECC内存

2.14 配置vGPU虚拟机,使用NVIDIA GPUDirect存储技术

要使用NVIDIA vGPU的NVIDIA®GPUDirect Storage®技术,您必须在配置了NVIDIA vGPU的虚拟机中安装所有所需的软件。 具体软件的安装步骤,详见链接

3. GPU直通

GPU直通用于直接将整个物理GPU分配给一个虚拟机,绕过NVIDIA虚拟GPU NVIDIA Virtual GPU Manager。在这种操作模式下,GPU被分配到的虚拟机中运行的NVIDIA驱动程序独占访问;虚拟机之间不共享GPU。

直通模式下,基于NVIDIA Maxwell架构之后的GPU支持ECC (error-correcting code)。

GPU直通可以与NVIDIA vGPU一起在服务器平台上使用,但有一些限制:

- 物理GPU可以托管NVIDIA vGPU,也可以用于直通,但不能同时做这两件事。有些管理程序,例如VMware vSphere ESXi,需要重启主机才能将GPU从 直通模式改为vGPU模式。
- 单个虚拟机不能同时配置vGPU和GPU直通。
- 直通到虚拟机的物理GPU的性能只能在虚拟机内部进行监控。这样的GPU无法被通过虚拟机管理程序运行的工具监控,例如xcenter或nvidia-smi(请参见 监控GPU性能)。
- 您的服务器平台必须启用以下BIOS设置:
 - 。VT-D / IOMMU的
 - 。高级选项中的SR-IOV
- 通过NVLink直接连接的所有GPU必须分配给同一个虚拟机。 注意:如果您打算将服务器平台中的所有GPU配置为直通,则不需要安装NVIDIA Virtual GPU Manager。

3.1. 物理GPU的显示分辨率

物理GPU支持的显示分辨率取决于NVIDIA GPU架构和应用于该GPU的NVIDIA vGPU软件许可证。

vWS物理GPU分辨率

使用vWS许可的GPU支持基于可用像素数量的最大组合分辨率,这是由NVIDIA GPU架构决定的。你可以选择使用少量的高分辨率显示器,或者使用这些gpu 使用大量的低分辨率显示器。

下表列出了在所有显示器分辨率相同的配置下,每个GPU支持的每种显示分辨率下的最大显示数。

NVIDIA GPU Architecture	Available Pixels	Display Resolution	Displays per GPU	
Pascal and later	662EE200	7680×4320	2	
rascal and later	66355200	5120×2880 or lower	4	
Maxwell	25280.4.40	5120×2880	2	
Maxwell	35369440	4096×2160 or lower	4	

下表提供了混合显示分辨率的配置示例。

NVIDIA GPU Architecture	Available Pixels	Available Pixel Basis	Maximum Displays	Sample Mixed Display Configurations	
Pascal and later	66355200	2 7680×4320 displays	4	1 7680×4320 display plus 2 5120×2880 displays	
				1 7680×4320 display plus 3 4096×2160 displays	
Maxwell	35389440	4 4096×2160 displays	4	1 5120×2880 display plus 2 4096×2160 displays	

注意:即使显示器的组合分辨率小于GPU的可用像素数,也不能使用超过四个显示器。例如,您不能使用基于NVIDIA Pascal架构的GPU的5个4096×2160显示 器,即使显示器的组合分辨率(44236800)小于GPU的可用像素数量(66355200)。

vApps或vCS物理GPU分辨率

使用vApps或vCS许可的gpu支持具有固定最大分辨率的单个显示。最大分辨率取决于以下因素。

- NVIDIA GPU的体系结构
- GPU使用的NVIDIA vGPU软件license
- 在分配GPU的系统上运行的操作系统
- •

License	NVIDIA GPU Architecture	Operating System	Maximum Display Resolution	Displays per GPU
vApps	Pascal or later	Linux	2560×1600	1
	Pascal or later	Windows	1280×1024	1
	Maxwell	Windows and Linux	2560×1600	1
vCS	Pascal or later	Linux	4096×2160	1

3.2. 在Citrix Hypervisor配置GPU直通

3.3. 在Red Hat Enterprise Linux KVM or Ubuntu配置GPU直通

3.4. 在Microsoft Windows Server配置GPU直通

3.5. 在VMware vSphere配置GPU直通

4. 安装NVIDIA vGPU Software Graphics Driver

安装NVIDIA vGPU软件图形驱动程序的过程取决于您使用的操作系统。但对于任何操作系统,在配置了vGPU的虚拟机、运行直通GPU的虚拟机、裸金属部署 的物理主机上安装驱动程序的过程都是一样的。 安装NVIDIA vGPU软件图形驱动程序后,您可以授权您正在使用的任何NVIDIA vGPU软件许可产品。

4.1. Installing the NVIDIA vGPU Software Graphics Driver on Windows

4.2. Installing the NVIDIA vGPU Software Graphics Driver on Linux

5. 授权NVIDIA vGPU

NVIDIA vGPU是一个授权产品。当在支持的GPU上启动时,vGPU最初以全速运行,但如果虚拟机无法获得license,其性能会随着时间的推移而下降。当 vGPU性能下降时,重新获取license后会恢复vGPU的能力。有关未授权vGPU性能如何下降的信息,请参见《虚拟GPU客户端授权用户指南》。 在获得NVIDIA vGPU许可后,设置为使用NVIDIA vGPU的虚拟机能够运行全系列的DirectX和OpenGL图形应用程序。 如果配置了license,虚拟机在这些GPU上启动vGPU时,会从license server获取license。虚拟机在关闭前将保留license。然后它将license释放回license server。license server设置在重启后仍然有效,只有当license server地址改变或虚拟机切换到运行GPU Pass时才需要修改。 如何授权NVIDIA vGPU取决于您的许可是来自NVIDIA license system还是传统的NVIDIA vGPU软件许可服务器。 注意:有关配置和使用NVIDIA vGPU软件许可特性(包括vGPU)的完整信息,请参见《虚拟GPU客户端授权用户指南》。

5.1. 配置NVIDIA License系统的授权客户端

要使用NVIDIA vGPU软件许可产品,每个分配了物理或虚拟GPU的客户端系统必须能够从NVIDIA license系统获得许可。客户端系统可以是配置了NVIDIA vGPU的虚拟机、配置了GPU直通的虚拟机、裸金属部署了物理GPU的主机。

在配置授权客户端之前,需要满足以下前提条件:

- 客户端已安装NVIDIA vGPU软件图形驱动程序。
- 您要在客户端上部署的客户端配置令牌已经从NVIDIA许可门户或DLS创建,如<<NVIDIA许可系统用户指南中所述>>。
- 防火墙或代理中允许服务实例和授权客户端之间的HTTPS通信的端口必须是开放的。防火墙或代理中必须开放的端口取决于服务实例是CLS实例还是DLS 实例:
 - 。对于CLS实例,端口443和80必须开放。
 - 。对于DLS实例,443、80、8081和8082端口必须是开放的。

图形驱动程序创建一个默认位置,用于在客户端上存储客户端配置令牌。 对于CLS和DLS实例,配置授权客户端的过程是相同的,但取决于客户端上运行的操作系统。

5.1.1. 在Windows下配置客户端

从客户端执行此任务。

1. 将 FeatureType DWord (REG_DWORD)注册表值添加到Windows注册表键 HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing。 注意:如果您正在升级现有的驱动程序,此值已经设置。 要设置的值取决于分配给您正在配置的授权客户端的GPU类型:

NVIDIA vGPU:不要更改此注册表项的值。NVIDIA vGPU软件根据vGPU类型自动选择license类型。

物理GPU: 直通模式或裸金属部署时GPU的特征类型: 0-NVIDIA虚拟应用程序 2-NVIDIA RTX虚拟工作站

2. 可选·如果您想将客户端配置令牌存储在自定义位置,请将 ClientConfigTokenPath 字符串(REG_SZ)注册表值添加到Windows注册表

键 HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing。

将值设置为要在其中存储客户端配置令牌的文件夹的完整路径。对于文件夹的路径,可以使用语法 \\fully-qualified-domain-name\share-name 。默认情况下, 客户端在 %SystemDrive%:\Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken 文件夹中搜索客户端配置令牌。

通过指定映射在客户机上的共享网络驱动器,可以简化在多个客户机上部署相同的客户机配置令牌。您可以在共享网络驱动器中只保留一个副本,而不是将客 户端配置令牌单独复制到每个客户端。

- 如果要在自定义位置存储客户端配置令牌,请创建要在其中存储客户端配置令牌的文件夹。
 如果是共享网络驱动器,请确保满足以下条件:
- 该文件夹在客户端本地映射到 ClientConfigTokenPath 注册表值中指定的路径。
- 计算机对象具有访问共享网络驱动器上的文件夹的权限。计算机对象需要这些权限,因为许可证服务在任何用户登录之前就会运行。

如果将客户端配置令牌存储在默认位置,则忽略此步骤。安装图形驱动后,会自动创建存放客户端配置令牌的默认文件夹。

- 4. 将客户端配置令牌复制到要存储客户端配置令牌的文件夹中。确保此文件夹只包含要在客户端上部署的客户端配置令牌,而不包含其他文件或文件夹。如果文件夹包含多个客户端配置令牌,则客户端使用文件夹中最新的客户端配置令牌。
- 如果要将客户端配置令牌存储在默认位置,请将客户端配置令牌复制 到 %SystemDrive%: Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken 文件夹。
- 如果要将客户端配置令牌存储在自定义位置,请将令牌复制到上一步中创建的文件夹中。
- 5. 重新启动NvDisplayContainer服务。

客户端上的NVIDIA服务现在应该自动从CLS或DLS实例获得许可证。 在配置了Windows许可客户端之后,NVIDIA控制面板中不再提供为基于网络的许可服务器配置许可的选项。

5.1.2. 在Linux下配置客户端

从客户端执行此任务。

1. 以root用户,使用纯文本编辑器(如vi)打开 /etc/nvidia/grid.conf 文件。

\$ sudo vi /etc/nvidia/grid.conf

注意:您可以通过复制提供的模板文件 /etc/nvidia/grid.conf.template 来创建 /etc/nvidia/grid.conf 文件。

- 2. 将FeatureType配置参数添加到 /etc/nvidia/grid.conf 文件中,在新行添加为FeatureType="value"。值取决于分配给您正在配置的授权客户端的GPU类型。
- NVIDIA vGPU : NVIDIA vGPU软件根据vGPU类型自动选择license类型
- 物理GPU:GPU直通模式或裸金属部署的FeatureType: 0-NVIDIA虚拟应用程序, 2-NVIDIA RTX虚拟工作站, 4-NVIDIA虚拟计算服务器 下例演示如何为NVIDIA Virtual Compute Server配置授权Linux客户端。
 - # /etc/nvidia/gridd.conf.template Configuration file for NVIDIA Grid Daemon
 - # Description: Set Feature to be enabled
 - # Data type: integer
 - # Possible values:
 - # 0 => for unlicensed state
 - # 1 => for NVIDIA vGPU
 - # 2 => for NVIDIA RTX Virtual Workstation
 - # 4 => for NVIDIA Virtual Compute Server
 FeatureType=4

....

3. **可选**:如果您想将客户端配置令牌存储在自定义位置,请将 ClientConfigTokenPath 配置参数添加到 /etc/nvidia/gridd.conf 文件的新行中,作为 ClientConfigTokenPath="path"

路径

要在其中储存客户端配置令牌的目录的完整路径。默认情况下,客户端在 /etc/nvidia/clientconfigtoken/ 目录中搜索客户端配置令牌。

通过指定在客户端本地挂载的共享网络目录,可以简化在多个客户端上部署相同的客户端配置令牌。您可以在共享网络目录中只保留一个副本,而不是将客户 端配置令牌单独复制到每个客户端。 这个例子展示了如何配置一个授权的Linux客户端,以便在 /mnt/nvidia/ClientConfigToken/ 目录中搜索客户端配置令牌。该目录是共享网络目录在客户机上的 挂载点。

/etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon

ClientConfigTokenPath=/mnt/nvidia/ClientConfigToken/

•••

- 4. 保存修改后的 /etc/nvidia/grid.conf 文件并关闭。
- 5. 如果要在自定义位置存储客户端配置令牌,请创建要在其中存储客户端配置令牌的目录。 如果该目录是共享网络目录,请确保在客户端本地挂载到ClientConfigTokenPath配置参数指定的路径。 如果将客户端配置令牌存储在默认位置,则忽略此步骤。图形驱动安装完成后,会自动创建存放客户端配置令牌的默认目录。
 6. 将客户端配置令牌复制到要存储客户端配置令牌的目录中。确保这个目录只包含要部署在客户端上的客户端配置令牌,而不包含其他文件或目录。如果目

录包含多个客户端配置令牌,则客户端使用目录中最新的客户端配置令牌。 如果想将客户端配置令牌保存在默认位置,请将客户端配置令牌复制到 /etc/nvidia/ClientConfigToken 目录。 如果要将客户端配置令牌存储在自定义位置,请将令牌复制到上一步创建的目录中。

- 7. 确保客户端配置令牌的文件访问模式允许Token拥有者读取、写入和执行令牌,而组和其他人只能读取令牌。
- 确定客户端配置令牌的当前文件访问模式。
 - # ls -1 client-configuration-token-directory
- 如果有必要,将客户端配置令牌的模式更改为744。
 # chmod 744 client-configuration-token-directory/client_configuration_token_*.tok
 client-configuration-token-directory
 在上一步中将客户端配置令牌复制到的目录。
- 8. 重新启动nvidia-grid服务。

客户端上的NVIDIA服务现在应该自动从CLS或DLS实例获得许可证。 在配置了Linu认证客户端之后,NVIDIA X服务器设置中不再提供为基于网络的许可服务器配置许可的选项。

5.1.3. 验证已授权客户端的NVIDIA vGPU软件License状态

在客户端配置NVIDIA vGPU软件license后,通过显示已授权的产品名称和状态来验证license状态。 执行nvidia-smi命令,带-q或——query选项,可以查看已授权客户端的license状态。如果产品已授权,则有效期显示在license状态中。

======NVSMI LOG=========	==	
Timestamp	:	Wed Mar 31 01:49:28 2020
Driver Version	:	440.88
CUDA Version	:	10.0
Attached GPUs	:	1
GPU 0000000:00:08.0		
Product Name	:	Tesla T4
Product Brand	:	Grid
Display Mode	:	Enabled
Display Active	:	Disabled
Persistence Mode	:	N/A
Accounting Mode	:	Disabled
Accounting Mode Buffer Size	:	4000
Driver Model		
Current	:	WDDM
Pending	:	WDDM
Serial Number	:	0334018000638
GPU UUID	:	GPU-ba2310b6-95d1-802b-f96f-5865410fe517
Minor Number	:	N/A
VBIOS Version	:	90.04.21.00.01
MultiGPU Board	:	No
Board ID	:	0×8
GPU Part Number	:	699-2G183-0200-100
Inforom Version		
Image Version	:	G183.0200.00.02
OEM Object	:	1.1
ECC Object	:	5.0
Power Management Object	:	N/A
GPU Operation Mode		
Current	:	N/A
Pending	:	N/A
GPU Virtualization Mode		
Virtualization mode	:	Pass-Through
vGPU Software Licensed Product		
Product Name	:	NVIDIA Virtual Compute Server
License Status	:	Licensed (Expiry: 2021-11-13 18:29:59 GMT)

5.2. 从许可证服务器授权NVIDIA vGPU

5.2.1. 在Windows上授权NVIDIA vGPU

在分配了vGPU的虚拟机中执行该任务。

用于执行此任务的NVIDIA控制面板工具检测到vGPU被分配给虚拟机,因此不提供选择许可证类型的选项。授权vGPU后,NVIDIA vGPU软件会根据vGPU类型自动选择正确的license类型。

1. 打开英伟达控制面板:

nvidia-smi -q

- 右键单击Windows桌面,从菜单中选择NVIDIA Control Panel。
- 打开Windows控制面板,双击NVIDIA控制面板图标。
- 2. 在NVIDIA控制面板中,在导航窗格的Licensing部分选择 Manage License 任务。

注意:如果在NVIDIA控制面板中没有显示Licensing和Manage License,则系统已经设置为在NVIDIA控制面板中隐藏许可控制。有关注册表设置的信息,请参考《虚拟GPU客户端授权用户指南》。

NVIDIA Control Panel		
<u>F</u> ile <u>E</u> dit Des <u>k</u> top <u>H</u> elp		
🕞 Back 👻 🚱		
Select a Task - 3D SettingsAdjust image settings with preview	Manage License	<u> </u>
Manage 3D settings Set PhysX Configuration Display	You can enable additional features by applying a license.	
	License Edition: Your system does not have a valid GRID vGPU license. To access GRID vGPU features, enter license server details and apply.	
Adjust video color settings Adjust video image settings	Primary License Server:	
	Port Number:	E
	Secondary License Server:	
	Port Number:	
	Description:	
	Typical usage scenarios:	
System Information	< III	→

- 3. 在"Primary License Server"字段中,输入主NVIDIA vGPU软件许可服务器的地址。地址可以是一个全称域名,如 gridlicense1.example.com,也可以是一个IP地址,如10.31.20.45。如果您只配置了一个许可证服务器,请在此字段中输入其地址。
- 4. 不设置主许可证服务器字段下的端口号字段。端口默认为"7070",即NVIDIA vGPU software License Server使用的默认端口号。
- 5. 在Secondary License Server字段中,输入次要NVIDIA vGPU软件许可服务器的地址。如果您只配置了一个许可证服务器,请不要设置该字段。地址可以 是一个全称域名,如 gridlicense2.example.com,也可以是一个IP地址,如10.31.20.46。
- 6. 保留备用License服务器字段下的端口号字段不设置。端口默认为"7070",即NVIDIA vGPU software License Server使用的默认端口号。
- 7. 单击Apply分配设置。系统从配置的许可证服务器为当前vGPU请求许可授权。

只要vGPU获得许可,VM中的vGPU现在就应该以完全的能力运行,不会随着时间的推移出现任何性能下降。 如果无法获取license,请参考《虚拟GPU客户端授权用户指南》进行故障处理。

5.2.2. 在Linux上授权NVIDIA vGPU

在分配了vGPU的虚拟机中执行该任务。

用于执行此任务的NVIDIA X服务器设置工具检测到vGPU被分配给虚拟机,因此不提供选择license类型的选项。授权vGPU后,NVIDIA vGPU软件会根据 vGPU类型自动选择正确的license类型。

请按照《虚拟GPU客户端授权用户指南》中的说明,启用"Manage License"选项。

注意:不要在Red Hat Enterprise Linux 6.8和6.9或CentOS 6.8和6.9中启用Manage License选项。为了防止DBus代码中的分割故障导致nvidia-grid服务退出, 必须在这些操作系统版本中禁用授权GUI。

- 1. 通过使用Linux发行版提供的启动应用程序的方法来启动NVIDIA X服务器设置。例如,在Ubuntu桌面上,打开Dash,搜索NVIDIA X服务器设置,然后单击NVIDIA X服务器设置图标。
- 2. 在打开的"NVIDIA X服务器设置"窗口中,单击"Manage GRID License"。NVIDIA X服务器设置窗口的License Edition部分显示NVIDIA vGPU目前未获得许可。
- 3. 在Primary Server字段中,输入主NVIDIA vGPU软件许可服务器的地址。地址可以是一个全称域名,如gridlicense1.example.com,也可以是一个IP地址,如10.31.20.45。如果您只配置了一个许可证服务器,请在此字段中输入其地址。
- 4. 不设置主服务器字段下的端口号字段。端口默认为"7070",即NVIDIA vGPU software License Server使用的默认端口号。

- 5. 在Secondary License Server字段中,输入辅助NVIDIA vGPU软件许可服务器的地址。如果您只配置了一个许可证服务器,请不要设置该字段。地址可以 是一个全称域名,如gridlicense2.example.com,也可以是一个IP地址,如10.31.20.46。
- 6. 保留备用服务器字段下的端口号字段不设置。端口默认为"7070",即NVIDIA vGPU software License Server使用的默认端口号。
- 7. 单击Apply分配设置。系统从配置的许可证服务器为当前vGPU请求适当的许可证。 只要vGPU获得许可,VM中的vGPU现在就应该以完全的能力运行,不会随着时间的推移出现任何性能下降。 如果无法获取license,请参考《虚拟GPU客户端授权用户指南》进行故障处理。

6.监控GPU性能

NVIDIA vGPU软件使您能够从Hypervisor和从单个客户虚拟机中监视物理gpu和虚拟gpu的性能。 可以使用几个工具来监控GPU性能:

- 从任何受支持的管理程序和运行64位版本Windows或Linux的客户VM中,您可以使用NVIDIA系统管理界面NVIDIA -smi。
- 在Citrix Hypervisor中,您可以使用Citrix XenCenter。
- 从Windows guest VM,您可以使用这些工具:
 - 。 Windows性能监视器
 - 。Windows管理工具(WMI)

6.1. NVIDIA System Management Interface nvidia-smi

NVIDIA System Management Interface, NVIDIA-smi, 是一个命令行工具, 用于报告NVIDIA gpu的管理信息。

nvidia-smi工具包含在以下软件包中:

- 每个支持的管理程序的NVIDIA虚拟GPU管理器包
- 每个支持的客户操作系统的NVIDIA驱动程序包

报告的管理信息的范围取决于您从哪里运行nvidia-smi:

- NVIDIA-smi通过Hypervisor命令shell,例如Citrix hypervisor dom0 shell或VMware ESXi host shell,报告系统中存在的NVIDIA物理gpu和虚拟gpu的管理信息。
 - 注意:当从管理程序命令shell运行时, nvidia-smi将不会列出任何当前分配给GPU直通的GPU。
- nvidia-smi从客户虚拟机中检索分配给该虚拟机的vgpu或直通gpu的使用统计信息。

在Windows guest虚拟机中,您可以通过切换到 C:\Program Files\NVIDIA Corporation\NVSMI 文件夹并运行 nvidia-smi.exe 命令,从命令提示符运行nvidiasmi。

6.2. 从Hypervisor监控GPU性能

可以通过使用NVIDIA系统管理接口NVIDIA-smi命令行实用程序从任何受支持的管理程序监视GPU性能。在Citrix Hypervisor平台上,您还可以使用Citrix xcenter来监控GPU性能。

注意:不能从Hypervisor上监控用于GPU直通的GPU的性能。您只能从使用直通GPU的客户VM中监视它们的性能。

6.2.1. 从Hypervisor使用nvidia-smi监控GPU Performance

您可以通过从Hypervisor命令shell(例如Citrix hypervisor dom0 shell或VMware ESXi host shell)运行NVIDIA -smi来获取系统中存在的NVIDIA物理GPU和vGPU 的管理信息。

在没有子命令的情况下, nvidia-smi为物理gpu提供管理信息。要更详细地检查vGPU, 请使用nvidia-smi和vGPU子命令。 从命令行, 您可以获得有关nvidia-smi工具和vGPU子命令的帮助信息。

- nvidia-smi -h : nvidia-smi工具支持的子命令列表。请注意,并非所有子命令都适用于支持NVIDIA vGPU软件的gpu。
- nvidia-smi vgpu -h : vgpu子命令支持的所有选项列表。

6.2.1.1. 得到系统中所有物理GPU的信息

要获得系统中所有物理GPU的信息,以及PCI总线id、电源状态、温度、当前内存使用情况等等,无需额外参数即可运行nvidia-smi。 每个vGPU实例在Compute processes部分中报告,以及它的物理GPU索引和分配给它的帧缓冲内存数量。 本例中,系统中运行3个vGPU:物理GPU 0、1、2各运行1个vGPU。

[root@vgpu ~]# nvidia-smi

Fri Aug 12 09:26:18 2022

NVIDIA-	SMI	510.8	35.03			Driver Ve	ersion: 5	51	0.85.03	
GPU Na Fan Te	ime imp	Perf	Persi Pwr:l	lst Jsa	ence-M ge/Cap	Bus-Id Memor	Disp.A ry-Usage		Volatile GPU-Util	Uncorr. ECC Compute M
0 Te N/A 3	sla 1C	M60 P8	23	ı /	On 150W	0000:83:00.0 1889MiB /	Off 8191MiB	 	7%	Of Defaul
1 Te N/A 2	sla 6C	M60 P8	231	1 /	On 150W	0000:84:00.0 926MiB /	Off 8191MiB		9%	Of Defaul
2 Te N/A 2	sla 3C	M10 P8	101	. /	On 53W	0000:8A:00.0 1882MiB /	Off 8191MiB		12%	N// Defaul
3 Te N/A 2	sla 6C	M10 P8	100	1 /	On 53W	0000:8B:00.0 10MiB /	Off 8191MiB		0%	N// Default
4 Te N/A 3	sla 4C	M10 P8	100	. /	On 53W	0000:8C:00.0 10MiB /	Off 8191MiB	 	0%	N// Defaul
5 Te N/A 3	sla 2C	M10 P8	100	1 /	On 53W	0000:8D:00.0 10MiB /	Off 8191MiB	 	0%	N// Defaul
Process GPU	es:	PID	Туре	Pr	ocess n	ame				GPU Memory Usage
0 1 2	11 11 11	1924 1903 1908	C+G C+G C+G	/u /u /u	sr/lib6 sr/lib6 sr/lib6	4/xen/bin/vgp 4/xen/bin/vgp 4/xen/bin/vgp	1 1 1			1856Mil 896Mil 1856Mil

6.2.1.2. 得到系统中所有vGPU的信息

要获得系统中当前运行在每个物理GPU上的vGPU的信息,请运行nvidia-smi vgpu,不带其他参数。

[root@ Fri Au	[root@vgpu ~]# nvidia-smi vgpu Fri Aug 12 09:27:06 2022										
+				Driver Version: 510.85.03							
GPU 	Name vGPU ID	Name	 	Bus-Id VM ID	VM Name	GPU-U	Jtil -Util				
0 	Tesla M60 11924	GRID M60-2Q	 	0000:83: 3	00.0 Win7-64 GRID test 2	7% +	6%				
1	Tesla M60 11903	GRID M60-1B	 	0000:84: 1	00.0 Win8.1-64 GRID test 3	9% 	 8%				
2	Tesla M10 11908	GRID M10-2Q	 	0000:8A: 2	00.0 Win7-64 GRID test 1	12% 	 10%				
3	Tesla M10			0000:8B:	00.0	0%					
4	Tesla M10		 _+	0000:8C:	00.0	0%					
5 +	Tesla M10		 -+	0000:8D:	00.0	0%	 ++				
[root@	vgpu ~]#										

6.2.1.3. 得到vGPU的详细信息

要获取平台上所有vgpu的详细信息,请使用nvidia-smi vgpu命令,并带-q或——query选项。 要将检索到的信息限制在平台上GPU的一个子集,可以使用-i或——id选项来选择一个或多个vgpu。

[root@vgpu ~]# nvidia-smi vgpu	•	-q -i 1
GPU 0000000:86:00.0		
Active vGPUs	:	1
vGPU ID	:	3251634178
VM ID	:	1
VM Name	:	Win7
vGPU Name	:	GRID M60-8Q
vGPU Type	:	22
VGPU UUID	:	b8c6d0e1-d167-11e8-b8c9-55705e5a54a6
Guest Driver Version	:	411.81
License Status	:	Unlicensed
Accounting Mode	:	Disabled
Accounting Buffer Size	:	4000
Frame Rate Limit	:	3 FPS
FB Memory Usage	:	
Total	:	8192 MiB
Used	:	675 MiB
Free	:	7517 MiB
Utilization	:	
Gpu	:	3 %
Memory	:	0 %
Encoder	:	0 %
Decoder	:	0 %
Encoder Stats	:	
Active Sessions	:	0
Average FPS	:	0
Average Latency	:	0
FBC Stats	:	
Active Sessions	:	1
Average FPS	:	227
Average Latency	:	4403
[root@vgpu ~]#		

6.2.1.4. 监控vGPU引擎使用情况

要监视多个vGPU的GPU引擎使用情况,请使用-u或——utilization选项运行nvidia-smi vGPU。

[1	root@v	gpu ~]#	nvidia	-smi	vgpu -u		
#	gpu	vgpu	sm	mem	enc	dec	
#	Idx	Id	%	%	%	%	
	0	11924	6	3	0	0	
	1	11903	8	3	0	0	
	2	11908	10	4	0	0	
	3	-	-	-	-	-	
	4	-	-	-	-	-	
	5	-	-	-	-	-	
	0	11924	6	3	0	0	
	1	11903	9	3	0	0	
	2	11908	10	4	0	0	
	3	-	-	-	-	-	
	4	-	-	-	-	-	
	5	-	-	-	-	-	
	0	11924	6	3	0	0	
	1	11903	8	3	0	0	
	2	11908	10	4	0	0	
	3	-	-	-	-	-	
	4	-	-	-	-	-	
	5	-	-	-	-	-	
^(^C[root@vgpu ~]#						

6.2.1.5. 根据应用监控vGPU的使用情况

要监控多个vGPU的应用程序对vGPU引擎的使用情况,请使用-p选项运行nvidia-smi vGPU。

[r	[root@vgpu ~]# nvidia-smi vgpu -p								
#	GPU	vGPU	process	process	sm	mem	enc	dec	
#	Idx	Id	Id	name	%	%	%	%	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	37408	4232	DolphinVS.exe	32	25	0	0	
	1	257869	4432	FurMark.exe	16	12	0	0	
	1	257969	4552	FurMark.exe	48	37	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	37408	4232	DolphinVS.exe	16	12	0	0	
	1	257911	656	DolphinVS.exe	32	24	0	0	
	1	257969	4552	FurMark.exe	48	37	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	257869	4432	FurMark.exe	38	30	0	0	
	1	257911	656	DolphinVS.exe	19	14	0	0	
	1	257969	4552	FurMark.exe	38	30	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	257848	3220	Balls64.exe	16	12	0	0	
	1	257869	4432	FurMark.exe	16	12	0	0	
	1	257911	656	DolphinVS.exe	16	12	0	0	
	1	257969	4552	FurMark.exe	48	37	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	257911	656	DolphinVS.exe	32	25	0	0	
	1	257969	4552	FurMark.exe	64	50	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	37408	4232	DolphinVS.exe	16	12	0	0	
	1	257911	656	DolphinVS.exe	16	12	0	0	
	1	257969	4552	FurMark.exe	64	49	0	0	
	0	38127	1528	dwm.exe	0	0	0	0	
	1	37408	4232	DolphinVS.exe	16	12	0	0	
	1	257869	4432	FurMark.exe	16	12	0	0	
	1	257969	4552	FurMark.exe	64	49	0	0	
[1	root(∮vgpu ~]‡	ŧ						

6.2.1.6. 监控Encoder的使用情况

注意:编码器只能监控分配给Windows虚拟机的vGPU。分配给Linux虚拟机的vgpu不上报编码器会话统计信息。

要监视在多个VGPU上运行的进程的编码器会话,请使用 -es 或 —encodersessions 选项运行 nvidia-smi vgpu 。 要修改报告频率,请使用-l或——loop选项。

要将监视限制在平台上的一个GPU子集,可以使用-i或——id选项来选择一个或多个vgpu。

[root@vgpu ~]# nvidia-smi vgpu -es										
#	GPU	vGPU	Session	Process	Codec	н	v	Average	Average	
#	Idx	Id	Id	Id	Туре	Res	Res	FPS	Latency(us)	
	1	21211	2	2308	H.264	1920	1080	424	1977	
	1	21206	3	2424	H.264	1920	1080	0	0	
	1	22011	1	3676	H.264	1920	1080	374	1589	
	1	21211	2	2308	H.264	1920	1080	360	807	
	1	21206	3	2424	H.264	1920	1080	325	1474	
	1	22011	1	3676	H.264	1920	1080	313	1005	
	1	21211	2	2308	H.264	1920	1080	329	1732	
	1	21206	3	2424	H.264	1920	1080	352	1415	
	1	22011	1	3676	H.264	1920	1080	434	1894	
	1	21211	2	2308	H.264	1920	1080	362	1818	
	1	21206	3	2424	H.264	1920	1080	296	1072	
	1	22011	1	3676	H.264	1920	1080	416	1994	
	1	21211	2	2308	H.264	1920	1080	444	1912	
	1	21206	3	2424	H.264	1920	1080	330	1261	
	1	22011	1	3676	H.264	1920	1080	436	1644	
	1	21211	2	2308	H.264	1920	1080	344	1500	
	1	21206	3	2424	H.264	1920	1080	393	1727	
	1	22011	1	3676	H.264	1920	1080	364	1945	
	1	21211	2	2308	H.264	1920	1080	555	1653	
	1	21206	3	2424	H.264	1920	1080	295	925	
	1	22011	1	3676	H.264	1920	1080	372	1869	
	1	21211	2	2308	H.264	1920	1080	326	2206	
	1	21206	3	2424	H.264	1920	1080	318	1366	
	1	22011	1	3676	H.264	1920	1080	464	2015	
	1	21211	2	2308	H.264	1920	1080	305	1167	
	1	21206	3	2424	H.264	1920	1080	445	1892	
	1	22011	1	3676	H.264	1920	1080	361	906	
	1	21211	2	2308	H.264	1920	1080	353	1436	
	1	21206	3	2424	H.264	1920	1080	354	1798	
	1	22011	1	3676	H.264	1920	1080	373	1310	
^(^C[root@vgpu ~]#									

6.2.1.7. 列出支持的vGPU类型

如果需要列出当前系统中图形处理器支持的虚拟图形处理器类型,请执行nvidia-smi vgpu命令,并使用-s或——supported选项。 要将检索到的信息限制在平台上GPU的一个子集,可以使用-i或——id选项来选择一个或多个GPU。

[root@vgpu ~]# nvidia-smi vgpu -s -i 0 GPU 0000:83:00.0 GRID M60-0B GRID M60-0Q GRID M60-1A GRID M60-1B GRID M60-1Q GRID M60-2A GRID M60-2Q GRID M60-2Q GRID M60-4A GRID M60-4Q GRID M60-8A GRID M60-8A GRID M60-8Q [root@vgpu ~]#

如果需要查看支持的vGPU类型的详细信息,请添加-v或——verbose选项。

[root@vgpu ~]# nvidia-smi vg	ou -s -i 0 -v less
GPU 0000000:83:00.0	
vGPU Type ID	: 0xb
Name	: GRID M60-0B
Class	: NVS
Max Instances	: 16
Device ID	: 0x13f210de
Sub System ID	: 0x13f21176
FB Memory	: 512 MiB
Display Heads	: 2
Maximum X Resolution	: 2560
Maximum Y Resolution	: 1600
Frame Rate Limit	: 45 FPS
GRID License	: GRID-Virtual-PC,2.0;GRID-Virtual-WS,2.0;GRID-Virtual-WS-Ext,2.0;Quadro-Virtual-DWS,5.
vGPU Type ID	: 0xc
Name	: GRID M60-0Q
Class	: Quadro
Max Instances	: 16
Device ID	: 0x13f210de
Sub System ID	: 0x13f2114c
FB Memory	: 512 MiB
Display Heads	: 2
Maximum X Resolution	: 2560
Maximum Y Resolution	: 1600
Frame Rate Limit	: 60 FPS
GRID License	: GRID-Virtual-WS,2.0;GRID-Virtual-WS-Ext,2.0;Quadro-Virtual-DWS,5.0
vGPU Type ID	: 0xd
Name	: GRID M60-1A
Class	: NVS
Max Instances	: 8
… [root@vgpu ~]#	

6.2.1.8. 列出当前可以创建的vGPU类型

要列出当前可以在系统中的GPU上创建的虚拟GPU类型,请运行nvidia-smi vgpu,并使用-c或——creatable选项。

这个属性是一个动态的属性, 它取决于每个GPU是否启用MIG模式。

- 如果GPU未启用MIG模式,或者GPU不支持MIG,则该属性反映已经在GPU上运行的vGPU的数量和类型。
 - 。如果当前图形处理器没有运行vGPU,则列出该图形处理器支持的所有vGPU类型。
 - 。如果该GPU上运行了一个或多个vgpu,但该GPU未加载满,则只列出已经运行的vgpu的类型。
 - 。GPU加载完成后,不显示vGPU类型。
- 如果GPU开启MIG模式,则该结果反映了当前未运行vGPU的GPU实例数量和类型。
 - 。如果没有创建GPU实例,则列表中不显示vGPU类型。
 - 。如果已创建GPU实例,则只列出未运行vGPU的GPU实例对应的vGPU类型。
 - 。如果vGPU运行在每个GPU实例上,则不会列出vGPU类型。

要将检索到的信息限制在平台上GPU的一个子集,可以使用-i或——id选项来选择一个或多个GPU。

```
[root@vgpu ~]# nvidia-smi vgpu -c -i 0
GPU 0000:83:00.0
GRID M60-2Q
[root@vgpu ~]#
```

6.2.2. 使用Citrix XenCenter监控GPU性能

如果您使用Citrix Hypervisor作为您的Hypervisor,您可以在XenCenter中监视GPU性能。

- 单击服务器的Performance选项卡。
- 右键单击graph窗口, 然后选择Actions和New graph。
- 为图形提供一个名称。在"可用指标资源"列表中,选择一个或多个GPU指标。



6.3. 从虚拟机监控GPU性能

您可以在单个客户VM中使用监视工具来监视分配给VM的vGPU或直通GPU的性能。这些工具的作用范围仅限于使用它们的客户VM。您不能在单个客户VM中 使用监视工具来监视平台中的任何其他gpu。

对于vGPU,只有这些指标在客户VM中报告:

- 3D/计算
- 内存控制器
- 视频编码器
- 视频解码器
- 帧缓冲区的使用

通常出现在GPU中的其他指标不适用于vGPU,并报告为零或N/a,这取决于您使用的工具。

6.3.1. 使用nvidia-smi从客户虚拟机监控GPU性能

6.3.2. 使用nWindows Performance Counters从客户虚拟机监控GPU性能

6.3.3. 使用NVWMI从客户虚拟机监控GPU性能

7. 改变时间切片vGPU的调度行为

基于NVIDIA Maxwell™图形架构的NVIDIA GPU实现了一个最大努力(Best Effort)的vGPU调度程序,旨在平衡vGPU之间的性能。最大努力(Best Effort)调度程 序允许vGPU使用其他vGPU不使用的GPU处理周期。在某些情况下,运行一个图形密集型应用程序的虚拟机可能会对运行在其他虚拟机上的轻图形应用程序 的性能产生不利影响。

基于NVIDIA GPU架构的GPU在Maxwell架构之后,额外支持相等份额和固定份额的vGPU调度器。这些调度器限制了vGPU使用的GPU处理周期,从而防止在 一个虚拟机中运行的图形密集型应用程序影响在其他虚拟机中运行的图形轻应用程序的性能。在支持多个vGPU调度器的GPU上,可以选择要使用的vGPU调

度器。您还可以为相等份额和固定份额vGPU调度器设置时间片的长度。

注意:如果你使用相等份额或固定份额vGPU调度程序, 帧速率限制器(FRL)是禁用的。

最大努力(Best Effort)调度器是所有支持GPU架构的默认调度器。

7.1. 时间切片vGPU的调度策略

除了默认的最大努力(Best Effort)调度器外,在Maxwell架构之后基于NVIDIA GPU架构的GPU还支持相等份额或固定份额vGPU调度器。

• 相等份额调度器

物理GPU在其上运行的vGPU之间平均共享。随着vGPU的添加或删除,分配给每个vGPU的GPU处理周期的份额也会相应变化。因此,当同一GPU上的 其他vGPU停止时,该vGPU的性能可能会增加,或者当同一GPU上的其他vGPU启动时,该vGPU的性能可能会下降。

• 固定份额调度器

每个vGPU都被赋予了物理GPU处理周期的固定份额,其数量取决于vGPU类型,而vGPU类型又决定了每个物理GPU的最大vGPU数量。例如,每个物理 GPU最大支持4个T4-4C vGPU。当调度策略为固定共享时,每个T4-4C vGPU被赋予1/4,即物理GPU处理周期的25%。随着vGPU的添加或删除,分配 给每个vGPU的GPU处理周期的份额保持不变。因此,当其他vGPU在同一GPU上停止或启动时,vGPU的性能保持不变。

7.2 时间切片vGPU调度器的时间切片

当多个虚拟机访问同一个GPU上的vGPU时,GPU会依次为每个虚拟机执行工作。vGPU调度程序时间片表示一个虚拟机的工作在被抢占并执行下一个虚拟机 的工作之前,允许在GPU上运行的时间量。

对于相等份额和固定份额v的vGPU调度器,可以设置时间片的长度。时间片的长度会影响延迟和吞吐量。时间片的最佳长度取决于GPU正在处理的工作负载。 对于需要低延迟的工作负载,更短的时间片是最佳的。通常,这些工作负载是必须以固定间隔生成输出的应用程序,例如以60帧/秒的帧率生成输出的图形应 用程序。这些工作负载对延迟很敏感,应该允许每个间隔至少运行一次。更短的时间片通过使调度程序在vm之间更频繁地切换来减少延迟和提高响应性。 对于需要最大吞吐量的工作负载,较长的时间片是最优的。通常,这些工作负载是必须尽快完成其工作且不需要响应的应用程序,例如CUDA应用程序。较长 的时间片可以防止虚拟机之间的频繁切换,从而提高吞吐量。

7.3. RmPVMRL 注册表键值

RmPVMRL注册表项通过设置调度策略和时间片长度来控制NVIDIA vGPU的调度行为。 注意:只有在支持多个vGPU调度器的GPU上才可以改变vGPU的调度行为,即在Maxwell架构之后基于NVIDIA GPU架构的GPU。

Туре

Dword

Content

- 0x00 (default) Best effort scheduler
- 0x01 Equal share scheduler with the default time slice length
- 0x00TT0001 Equal share scheduler with a user-defined time slice length TT
- 0x11 Fixed share scheduler with the default time slice length
- 0x00TT0011 Fixed share scheduler with a user-defined time slice length TT

默认时间片长度取决于vGPU类型允许的每个物理GPU的最大vGPU数量。

- 小于等于8个-2 ms
- 大于8个-1 ms

тт

两位范围为01到1E的十六进制数字,用于设置相等份额和固定份额调度的时间片长度,以毫秒为单位。最小长度为1ms,最大长度为30ms。

当TT为00时,长度设置为vGPU类型的默认长度。 如果TT大于1E,则设置为30ms。

Examples

RmPVMRL=0x01:这个例子将vGPU调度器设置为与默认时间片长度相等的相等份额调度器。 RmPVMRL=0x00030001:这个例子将vGPU调度器设置为与3毫秒长的相等份额共享调度器。 RmPVMRL=0x11:这个例子将vGPU调度器设置为使用默认时间片长度的固定份额调度器。 RmPVMRL=0x00180011:这个例子将vGPU调度器设置为固定份额调度器,时间片为24 (0x18)毫秒。

7.4. 获取当前所有GPU的时间切片vGPU调度行为

在更改一个或多个GPU的调度行为之前获取当前的调度行为,以确定是否需要更改它,或在更改后获取当前调度行为,以确认更改。 在Hypervisor命令shell中执行此任务。

- 1. 在Hypervisor打开命令shell。在所有支持的Hypervisor上,您可以使用secure shell (SSH)来实现这一目的。各个管理程序可能提供额外的登录方法。有关详细信息,请参阅您的管理程序的文档。
- 2. 使用dmesg命令显示来自内核的消息,其中包含字符串NVRM和scheduler 调度行为在这些消息中通过以下字符串表示:

BEST_EFFORT
 EQUAL_SHARE
 FIXED_SHARE
 M果调度行为为相等份额或固定份额,也以毫秒(ms)中的调度器时间片长度。
 本例获取了一个系统中GPU的调度行为,其中一个GPU的行为被设置为best effort,一个GPU被设置为equal share,一个GPU被设置为fixed share。
 \$ dmesg | grep NVRM | grep scheduler
 2020-10-05T02:58:08.9282 cpu79:2100753)NVRM: GPU at 0000:3d:00.0 has software scheduler DISABLED with policy BEST_EFFORT.
 2020-10-05T02:58:09.8182 cpu79:2100753)NVRM: GPU at 0000:5e:00.0 has software scheduler ENABLED with policy EQUAL_SHARE.
 NVRM: Software scheduler timeslice set to 1 ms.
 2020-10-05T02:58:12.1152 cpu79:2100753)NVRM: GPU at 0000:88:00.0 has software scheduler ENABLED with policy FIXED_SHARE.
 NVRM: Software scheduler timeslice set to 1 ms.

7.5. 改变所有GPU的时间切片vGPU调度行为

注意:只有在支持多个vGPU调度器的GPU上才可以改变vGPU的调度行为,即在Maxwell架构之后基于NVIDIA GPU架构的GPU。 在管理程序命令shell中执行此任务。

- 1. 在Hypervisor上打开命令shell。在所有支持的hypervisor上,您可以使用secure shell (SSH)来实现这一目的。各个管理程序可能提供额外的登录方法。有 关详细信息,请参阅您的管理程序的文档。
- 2. 将RmPVMRL注册表项设置GPU调度策略和您想要的时间片长度的值
 - 在Citrix Hypervisor、Red Hat Enterprise Linux KVM或Red Hat Virtualization (RHV)环境下,在"/etc/modprobe.d/nvidia.conf"文件中添加如下条目。
 options nvidia NVreg_RegistryDwords="RmPVMRL=value"
 - 。在VMware vSphere环境中,使用esxcli set命令。
 - esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwords=RmPVMRL=value"
- 3. 设定值
 - 。RmPVMRL=0x01:这个例子将vGPU调度器设置为与默认时间片长度相等的相等份额调度器。
 - 。RmPVMRL=0x00030001:这个例子将vGPU调度器设置为与3毫秒长的相等份额共享调度器。
 - 。RmPVMRL=0x11:这个例子将vGPU调度器设置为使用默认时间片长度的固定份额调度器。
 - 。RmPVMRL=0x00180011:这个例子将vGPU调度器设置为固定份额调度器,时间片为24 (0x18)毫秒。
- 4. 重启主机

7.6. 为特定GPU的时间切片改变vGPU调度行为

注意:只有在支持多个vGPU调度器的GPU上才可以改变vGPU的调度行为,即在Maxwell架构之后基于NVIDIA GPU架构的GPU。 在管理程序命令shell中执行此任务。

- 1. 在Hypervisor上打开命令shell。在所有支持的hypervisor上,您可以使用secure shell (SSH)来实现这一目的。各个管理程序可能提供额外的登录方法。有 关详细信息,请参阅您的管理程序的文档。
- 2. 使用lspci命令获取需要改变调度行为的GPU的PCI域和总线/设备/功能(BDF)
 - 。在Citrix Hypervisor、Red Hat Enterprise Linux KVM或Red Hat Virtualization (RHV)上,添加-D选项以显示PCI域,添加-D 10de:选项以仅显示 NVIDIA gpu的信息
 - # lspci -D -d 10de:
 - 。在VMware vSphere上,将lspci的输出管道连接到grep命令,仅显示NVIDIA GPU的信息
 - # lspci | grep NVIDIA

本例中列出的NVIDIA GPU具有PCI地址范围0000和BDF 86:00.0。

0000:86:00.0 3D controller: NVIDIA Corporation GP104GL [Tesla P4] (rev a1)

- 3. 使用模块参数NVreg_RegistryDwordsPerDevice为每个GPU设置pci和RmPVMRL注册表项。
 - 。在Citrix、Red Hat Enterprise Linux KVM或RHV环境下,在"/etc/modprobe.d/nvidia.conf"文件中添加如下条目:
 - options nvidia NVreg_RegistryDwordsPerDevice="pci=pci-domain:pci-bdf;RmPVMRL=value [;pci=pci-domain:pci-bdf;RmPVMRL=value...]"
 - 。在VMware vSphere环境中,使用esxcli set命令。

esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwordsPerDevice=pci=pci-domain:pci-bdf;RmPVMRL=value [;pci=pci-domain:pci-对于每个GPU, 提供以下信息:

pci-domain The PCI domain of the GPU. pci-bdf The PCI device BDF of the GPU.

Value

例如,设置GPU调度策略和您想要的时间片长度的值

0x01:这个例子将vGPU调度器设置为与默认时间片长度相等的相等份额调度器。

0x00030001:这个例子将vGPU调度器设置为与3毫秒长的相等份额共享调度器。

0x11:这个例子将vGPU调度器设置为使用默认时间片长度的固定份额调度器。

0x00180011:这个例子将vGPU调度器设置为固定份额调度器,时间片为24 (0x18)毫秒。

。 在/etc/modprobe.d/nvidia.conf文件中增加一个条目来改变单个GPU的调度行为。该条目设置GPU at PCI domain 0000和BDF 86:00.0的GPU调度策 略为固定份额调度程序,默认时间片长度。

options nvidia NVreg_RegistryDwordsPerDevice="pci=0000:86:00.0;RmPVMRL=0x11"

。在/etc/modprobe.d/nvidia.conf文件中增加一个条目来改变单个GPU的调度行为。该条目将PCI domain 0000和BDF 86:00.0的GPU的调度策略设置为 固定份额调度程序,时间片为24 (0x18) ms长。

options nvidia NVreg_RegistryDwordsPerDevice="pci=0000:86:00.0;RmPVMRL=0x00180011"

。修改虚拟化环境中运行VMware vSphere的主机的单个GPU调度行为。该命令将PCI domain 0000和BDF 15:00. 00 GPU的调度策略设置为固定份额 调度策略,并使用默认的时间片长度。

esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwordsPerDevice=pci=0000:15:00.0;RmPVMRL=0x11[;pci=0000:15:00.0;RmPVMRL=0: 0 修改虚拟化环境中运行VMware vSphere的主机的单个GPU调度行为。该命令将PCI domain 0000和BDF 15:00的GPU的调度策略设置为24 (0x18) ms

。修改虚拟化环境中运行VMware VSphere的主机的单个GPU崩度行为。该命令将PCI domain 0000和BDF 15:00的GPU的响度束略设置为24 (0X18) ms 的固定份额调度器。

esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwordsPerDevice=pci=0000:15:00.0;RmPVMRL=0x11[;pci=0000:15:00.0;RmPVMRL=0: 4. 重启主机

7.7. 恢复默认的时间切片vGPU调度程序设置

在管理程序命令shell中执行此任务。

- 1. 在Hypervisor主机上打开命令shell。在所有支持的hypervisor上,您可以使用secure shell (SSH)来实现这一目的。各个管理程序可能提供额外的登录方法。有关详细信息,请参阅您的管理程序的文档。
- 2. 取消设置RmPVMRL注册表项
 - 。在Citrix Hypervisor、Red Hat Enterprise Linux KVM或Red Hat Virtualization (RHV)上, 注释掉/etc/modprobe.d/nvidia.conf文件中通过#前缀设置 RmPVMRL的条目。
 - 。在VMware vSphere环境下,设置该模块参数为空字符串。

esxcli system module parameters set -m nvidia -p "module-parameter="

module-parameter

要设置的模块参数,它取决于调度行为是否为所有GPU更改或选择GPU:

对于所有GPU,设置 NVreg_RegistryDwords 模块参数。

对于select GPU,设置 "NVreg_RegistryDwordsPerDevice" 模块参数。

例如,要在更改所有gpu的vGPU调度程序设置后恢复默认vGPU调度程序设置,输入以下命令:

- # esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwords="
- 3. 重启Hypervisor主机

8.vGPU支持的虚拟化平台和产品

Citrix 虚拟化支持

Driver Package	Hypervisor or Bare-Metal OS	Software Product Deployment	Hardware Supported	Guest OS Support	Supported Virtualization Products_, 4
NVIDIA vGPU for XenServer 8.25	Citrix Hypervisor 8.2	NVIDIA vGPU GPU pass through	• A2 • A10 • A16	• Windows 10 21H2 ⁶	Citrix Virtual Apps and Desktops Teradici Cloud Access Software
			 A30⁷, A30X⁷ A40⁶, A100⁷, A100X⁷ M6, M10, M60 P4, P6, P40, P100, P100 12GB T4 V100 	14.0, 14.1 only: Windows Server 2016 1607, 1709 Windows Server 2019 Windows Server 2019 Windows Server 2022	Citrix Virtual Apps and Desktops RDSH Teradici Cloud Access Software
			 RTX A5000⁶ RTX A5000⁶ RTX A6000⁶ RTX 6000, RTX 6000 passive, RTX 8000, RTX 8000 passive⁶ 	 Since 14.3: Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.6 14.0 only: Red Hat Enterprise Linux 8.2, 8.4, 8.5 Red Hat Enterprise Linux 7.9 Ubuntu 20.04 LTS Ubuntu 16.04 LTS Ubuntu 16.04 LTS Ubuntu 14.04 LTS 	VNC Teradici Cloud Access Software
14.0, 14.1 only: NVIDIA vGPU for XenServer	r Citrix XenServer 7.1	NVIDIA vGPU GPU pass through	 M6, M10, M60 P4, P6, P40, P100 P100 12GB 	• Windows 10 21H2	Citrix Virtual Apps and Desktops Teradici Cloud Access Software
			 V100 RTX 6000, RTX 8000⁶ 	14.0, 14.1 only: Windows Server 2016 1607, 1709 Windows Server 2019 Windows Server 202210, 11	Citrix Virtual Apps and Desktops RDSH Teradici Cloud Access Software
				Red Hat Enterprise Linux 7.9 Ubuntu 18.04 LTS ¹² Ubuntu 16.04 LTS Ubuntu 16.04 LTS Ubuntu 14.04 LTS	VNC Teradici Cloud Access Software

Microsoft Windows Server支持

Driver Package	Hypervisor or Bare-Metal OS	Software Product Deployment	Hardware Supported	Guest OS Support	Supported Virtualization Products
NVIDIA vGPU for Windows	Windows 11 21H2 Windows 10	 Bare Metal¹³ RemoteFX vGPU 	 A2 A10 A16 A66 P4, P6, P40, P100 P100 12GB T4 V100 RTX A5000⁶ RTX A5000⁶ RTX A5000⁶ RTX A5000⁶ RTX A6000, RTX 6000 passive, RTX 8000, RTX 8000 passive⁶ 		Citrix Virtual Apps and Desktops
	 Windows Server 2019¹⁴ 14.0, 14.1 only: Windows Server 2019¹⁸ 14.0, 14.1 only: Windows Server 2016 1803 14.0, 14.1 only: Windows Server 2016 1607 	 Bare Metal¹³ RemoteFX VGPU¹⁴ DDA 	 A2 A10 A16 A30, A30X¹⁵ A40⁶ A100, A100X¹⁵ M6, M10, M60 P4, P6, P40, P100, P100 12GB T4 T4 T10 RTX A5500⁶ RTX A5500⁶ RTX A500⁶ RTX 6000, RTX 6000 passive, RTX 8000, RTX 8000 passive⁶ 	 Windows 11 21H2¹⁶ Windows 10 21H2⁸ 14.0, 14.1 only: Windows Server 2019 Windows Server 2019 Windows Server 2022 Ubuntu 20.04 LTS Ubuntu 16.04 LTS Ubuntu 16.04 LTS Since 14.3; Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.0 only:Red Hat Enterprise Linux 8.4, 8.6 14.0 only:Red Hat Enterprise Linux 7.9 SUSE Linux Enterprise Server 12 SP2+ 	Citrix Virtual Apps and Desktops Horizon ROSH RemoteFX

Red Hat Enterprise Linux with KVM 支持

Driver Package	Hypervisor or Bare-Metal OS	Software Product Deployment	Hardware Supported	Guest OS Support ¹ , ²	Supported Virtualization Products 4
Since 14.1: NVIDIA VGPU for RHEL KVM 9.0	Red Hat Enterprise Linux with KVM 9.0	NVIDIA VGPU NVIDIA Virtual Compute Server_ GPU pass through	 A2 A10 A10 A20⁽³⁾/₂ A100⁽³⁾/₂ A100⁽³⁾/₂ /li> A100⁽³	• Windows 10 21H2 <mark>8, 30</mark>	HP RGS Mechdyne TGX NICE DCUV Teradici Cloud Access Software SPICE is not supported.
				 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RCS Mechdyne TGX NICE DCV Teradici Cloud Access Software SPICE is not supported.
				Red Hat Enterprise Linux 9.0 Since 14.3: Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.6 Red Hat Enterprise Linux 7.9	HP RGS Mechdyne TGX NICE DCV Teradici Cloud Access Software SPICE is not supported.
Since 14.3: NVIDIA vGPU for RHEL KVM 8.7	Red Hat Enterprise Linux with KVM.8.7 RHV 4.4	NVIDIA VGPU NVIDIA Virtual Compute Server_ GPU pass through	 A2 A10 A16 A30⁽¹⁾, A30⁽¹⁾, A100⁽¹⁾, A100	• Windows 10 21H2 ⁰ , 20	HP RGS Mechdyne TGX NCE DCV Teradici Cloud Access Software SPICE is not supported.
				Windows Server 2019	HP RGS Mechdyne TGX NICE DCU Teradici Cloud Access Software SPICE is not supported.
				Red Hat Enterprise Linux 8.4, 8.6, 8.7 Red Hat Enterprise Linux 7.9	HP RGS Mechdyne TGX McEcDV EGX NCE DCV Teradici Cloud Access Software SPICE is not supported.
Since 14.1: NVIDIA vGPU for RHEL KVM 8.6	VIDIA vGPU for RHEL KVM 8.6 • Red Hat Enterprise Linux with KVM 8.6 • NVIDIA vGPU • A2 • NVIDIA vGPU for RHEL KVM 8.6 • NVIDIA vGPU • A10 • GPU pass through • A2 • A10 • A10 • GPU pass through • A2 • A10 • A10 • GPU pass through • A2 • A10 • A10 • A2 • A10 • A10 • A2 • A10	- A2 - A10 - A16 - A30 ¹ , A30V ¹⁹ - A40 ² , A100 ¹⁷ , A100V ¹⁹ - M5, M10, M50 - M50, M50 - M50, M10, M10, M50 - M50, M10, M10, M50 - M50, M10, M10, M10, M10, M10, M10, M10, M1	• Windows 10 21H2 <mark>⁶, 30</mark>	HP RGS Mechdyne TGX NICE DCV Teradici Cloud Access Software SPICE is not supported.	
			н т.	 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RGS Mechdyne TGX NRCE DCV Teradici Cloud Access Software SPICE is not supported.
				 Since 14.3: Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.6 Red Hat Enterprise Linux 7.9 	HP ROS Mechdyne TGX MCE DCV Tenadici Cloud Access Software SPICE is not supported.

14.0 only: NVIDIA vGPU for RHEL KWM 8.5	Red Hat Enterprise Linux with KVM 8.5 RHV 4.4 ¹⁷	NVIDIA vGPU NVIDIA Virtual Compute Server ¹⁸ GPU pass through	. 42 . 40 . 410 . 416 . 400 ¹⁰ , 4100 ¹⁰ , 4100X ¹⁰ . 40 ¹⁰ , 4100X ¹⁰ . 40 ¹⁰ , 4100X ¹⁰ . 400, 410, 4100, 4100 126B . 4 . 4000 . RTX.4500 ² . RTX.4500 ² . RTX.4500 ² . RTX.45000 ² . RTX.45000, RTX 6000 passive, RTX 8000, RTX	• Windows 10 21H2 ⁸ , 20	HP RGS Mechdyme TGX NICE DCV Teradici Cloud Access Software SPICE is not supported.
				 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RGS Mechdyne TGX McEchdyne TGX NCE DCV Teradici Cloud Access Software SPICE is not supported.
				Red Hat Enterprise Linux 8.2, 8.4, 8.5 Red Hat Enterprise Linux 7.9	HP RGS Mechdyne TGX MicE DCV Teradici Cloud Access Software SPICE is not supported.
NVIDIA vGPU for RHEL KVM 8.4	Red Hat Enterprise Linux with KVM 8.4 RHV 4.4 ¹⁷	NVIDIA VGPU NVIDIA Virtual Compute Server GPU pass through	 A2 A10 A16 A00[*]₂, A100[*]₂ A00[*]₂, A100[*]₂ A00[*]₂, A100[*]₂ A00[*]₂, A100[*]₂ P4, p5, P4, P4, P4, P4, P00, P100 12G8 T7 T40 VTA VTA S00[*] RTX, A500[*] RTX, A500[*] RTX, A500[*] RTX, 4500[*] 	• Windows 10 21H2 ⁶ , 30	HP RGS Mechdyne TGX McEchdyne TGX NCE DCV Teradici Cloud Access Software SPICE is not supported.
				 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RGS Mechdyne TGX MCEChdyne TGX NCE DCV Teradici Cloud Access Software SPICE is not supported.
				 Since 14.3: Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.6 only:Red Hat Enterprise Linux 8.2, 8.4, 8.5 Red Hat Enterprise Linux 7.9 	HP RGS Mechdyne TGX MICE DCV Teradici Cloud Access Software SPICE is not supported.
14.0 only: NVIDIA vGPU for RHEL KVM 8.2	Red Hat Enterprise Linux with KVM 8.2 RHV 4.4 $^{77}_{}$	NVIDIA VGPU NVIDIA Virtual Compute Server GPU pass through	 A2 A10 A16 A30¹⁰, A300¹⁰, A100¹⁰, A100¹⁰ A40², A100¹⁰, A100¹⁰, A100¹⁰ A40², A10¹⁰, A100¹⁰, A100¹⁰ A40², A10¹⁰, A100¹⁰ A40¹⁰, A10¹⁰ A40¹⁰, A10¹⁰ A40¹⁰, A10¹⁰ A40¹⁰ A40	• Windows 10 21H2 ⁹ , <u>50</u>	HP RGS Mechdyne TGX NicE DCV Teradici Cloud Access Software SPICE is not supported.
				 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RGS Mechdyne TGX McEchdyne TGX Teradici Cloud Access Software SPICE is not supported.
				Red Hat CoreOS 4.7 Red Hat Enterprise Linux 8.2, 8.4, 8.5 Red Hat Enterprise Linux 7.9	HP RGS Mechdyne TGX NiCE DCV Teradici Cloud Access Software SPICE is not supported.

NVIDIA vGPU for RHEL KVM 7.9	Red Hat Enterprise Linux with KVM 7.9	NVIDIA VGPU NVIDIA Virtual Compute Server_1 GPU pass through	 A2²¹ A10²¹ A30⁷, A300⁷, A100⁷, A100⁷, A100⁷, A40²¹, ¹⁰, ¹⁰	• Windows 10 21H2 ⁶ , ²⁰	HP RCS Mcchdyne TGX NICE DCY Teradici Cloud Access Software SPICE is not supported.
				 14.0, 14.1 only: Windows Server 2016 1607, 1709²⁰ Windows Server 2019 	HP RGS Mechdyne TGX, NICE DCV Teradici Cloud Access Software SPICE is not supported.
				Red Hat Enterprise Linux 7.9	HP RGS Mechdyne TGX NICE DCV Terradici Cloud Access Software SPICE is not supported.

ubuntu 支持

Driver Package	Hypervisor or Bare-Metal OS	Software Product Deployment	Hardware Supported	Guest OS Support ^{1, 2}	Supported Virtualization Products ^{2, 4}
Since 14.1: Ubuntu 22.04 LTS	Ubuntu 22.04 LTS	NVIDIA Victual Compute Server ¹⁸ GPU pass through	A2 A10 A16 A00, A300, A300, A100, A1000, A100, A1	Ubuntu 22.04 LTS Ubuntu 20.04 LTS Ubuntu 18.04 LTS	Teradici Cloud Access Software SPICE is not supported.
Ubuntu 20.04 LTS	Ubuntu 20.04 LTS	NVIDIA VGPU NVIDIA VGPU GPU pass through GPU pass through	A2 A10 A1	Ubuntu 20.04 LTS Ubuntu 18.04 LTS	Teradici Cloud Access Software SPICE is not supported.
Ubuntu 18.04 LTS	Ubuntu 18.04 LTS Support is limited to HWE kernels 5.4.0-77 and later.	NVIDIA VGPU NVIDIA Virtual Compute Server ¹⁸ GPU pass through	A2 A10 A1	Ubuntu 20.04 LTS Ubuntu 18.04 LTS	Teradici Cloud Access Software SPICE is not supported.

VMwarea平台支持

Driver Package	Hypervisor or Bare-Metal OS	Software Product Deployment	Hardware Supported	Guest OS Support ^{1, 2}	Supported Virtualization Products ³ , 4
NVIDIA vGPU for vSphere 7.0 ²²	ESXI 7.0 Update 2 and later compatible updates ESXI 7.0 and 7.0 Update 1 are not supported.	NVIDIA vGPU Vitiware v9GA Vitiware v9GA	 A2 A10 A16 A40⁶ 	Windows 11 21H2 Windows 10 21H2	Horizon Citrix Virtual Apps and Desktops Teradici Cloud Access Software
			 M6, M0, M60 P4, F6, P40, P100, P100 1208 T4 Y100 RT, A5000⁶ RT, A5000⁶ RTX 6000, RTX 6000 passive, RTX 8000, RTX 8000 passive⁵ 	 14.0, 14.1 only: Windows Server 2016 1607, 1709 Windows Server 2019 Windows Server 2022 	Citrix Virtual Apps and Desktops RDSH Teradici Cloud Access Software
				Debian 10 ²⁸ Ped Hat CoreOS 4.7 ²⁶ Since 14.3: CoreOS 4.7 ²⁶ Since 14.3: Red Hat Enterprise Linux 8.4, 8.6, 8.7 14.1, 14.2, only: Red Hat Enterprise Linux 8.4, 8.6 Only: Red Hat Enterprise Linux 7.9 SUSE Linux Enterprise Server 15 SP2 SUSE Linux Enterprise Server 15 SP2 Ubuntu 20.24 LTS Ubuntu 20.24 LTS Ubuntu 20.24 LTS Ubuntu 14.04 LTS	Horizon Teradici Cloud Access Software
14.0, 14.1 only: NVIDIA vGPU for vSphere 6.7 ²⁴	14.1 only: NVIDIA vGPU for vSphere 6.74 ESXI 6.7 NVIDIA vGPU support requires VMmare ESXI 6.7, Patch • NVIDIA vGPU NVIDIA vGPU support requires VMmare ESXI 6.7, Patch • NVIDIA vGPU • Willow vGPU support requires VMmare ESXI 6.7, Patch • NVIDIA vGPU • Will vGPU support requires VMmare ESXI 6.7, Patch • NVIDIA vGPU • Will vGPU support requires VMmare vGA • NVIDIA vGPU • VMmare vGA • NVIDIA vGPU	 A2²¹ A10²¹ A16²¹ 	• Windows 10 21H2 ⁸	Horizon Citrix Virtual Apps and Desktops Teradici Cloud Access Software	
			Add	14.0, 14.1 only: Windows Server 2016 1607, 1709 Windows Server 2019 Windows Server 2022 ²⁰	Citrix Virtual Apps and Desktops RDSH Teradici Cloud Access Software
				 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.4 0 only: Red Hat Enterprise Linux 8.2, 8.4, 8.5 Bed Hat Enterprise Linux 7.9 SUSE Linux Enterprise Enver 12 SP3+ Ubuntu 8.0.4 LTS Ubuntu 14.0.4 LTS 	Horizon Teradici Cloud Access Software
14.0, 14.1 only: NVIDIA vGPU for vSphere 6.524	ESXi 6.5 NVIDIA vGPU support requires <u>VMware ESXi 6.5</u> , Patch Release ESXi650-202102001, build 17477841 or later from VMware.	NVIDIA vGPU VMware vDGA VMware vSGA ²³	- A2 ²¹ - A10 ²¹ - A14 ²¹	Windows 10 21H2 ⁸	Horizon Citrix Virtual Apps and Desktops Teradici Cloud Access Software
			 A40^{21, 6} M6, M10, M60 P4, P6, P40, P100, P100 12GB T4 V100 PTV 45000²¹, 6 	 14.0, 14.1 only: Windows Server 2016 1607, 1709 Windows Server 2019²⁹ 	Citrix Virtual Apps and Desktops RDSH Teradici Cloud Access Software
			= nin AB002T, i = RTX 46002T, i = RTX 46002T, i = RTX 6000, RTX 6000 passive, RTX 8000, RTX 8000 passive [®]	 14.1, 14.2 only: Red Hat Enterprise Linux 8.4, 8.6³⁰ 14.0 only:Red Hat Enterprise Linux 8.2, 8.4, 8.6³¹ Red Hat Enterprise Linux 7.9 SUSE Linux Enterprise Server 12 SP3+ Ubantu 16.24 LT5 Ubantu 16.24 LT5 Ubantu 14.24 LT5 	Horizon Teradici Cloud Access Software

总结

本文主要介绍vGPU的基本特点,原理以及一些高级特性,希望读者可以通过本文对vGPU解决方案有个整体认识。同时本文在第九章节,总结了目前针对不同 产品和平台的支持情况,可供作资料进行查阅。本文不涉及具体产品和平台下的安装部署过程,相关内容可参考其他同事的工作《vGPU配置安装手册 v1.5.pdf》和相应链接,其中有详细介绍,本文不再做赘述。