

Virtuelle Desktops mit GPUs beschleunigen

Desktops mit Nachbrenner

von Dr. Bernhard Tritsch und Dr. Christian Knermann

Virtuelle Desktops mittels Grafikkarten auf der Serverseite zu beschleunigen, das ist längst keine Nische für besonders leistungshungrige Spezialanwendungen mehr. Auch typische Office-Nutzer profitieren signifikant davon, wenn unter der Haube ihrer Desktops im Rechenzentrum zusätzlich zur CPU auch Grafikkarten ihren Dienst tun. In diesem Beitrag beleuchten wir die Grundlagen der GPU-Beschleunigung für virtuelle Desktops.

Wer in gar nicht so weit zurückliegender Vergangenheit, also vor zehn bis 15 Jahren, einschlägige Konferenzen rund um Terminalserver und Virtual Desktop Infrastructure (VDI) besuchte, fand Vorträge zum Thema Grafikbeschleunigung nur am Rande. Typische Szenarien, die seinerzeit als Referenz dienten, waren High-End-Anwendungen, wie etwa CAD im Flugzeugbau. Solche Beispiele zeigten zwar damals schon eindrucksvoll das technisch Machbare, waren aber für typische Nutzungsprofile im Office-Sektor weder zwingend nötig noch erschwinglich.

Dass sich dies heutzutage gewandelt hat, liegt zum einen daran, dass sich die Art und Weise, wie Windows seine Oberfläche darstellt, verändert hat und der Aufbau eines typischen Desktops mit seinen Anwendungen komplexer geworden ist. Zum anderen haben sich damit einhergehend das Angebot an verfügbarer Grafikkarte sowie deren Unterstützung durch Hypervisoren vom Exoten zum Mainstream entwickelt.

Im Dschungel der Abkürzungen

Im Folgenden beschäftigen wir uns mit der Frage, ob und wie der Einsatz von

Grafikkarten, auch als Graphics Processing Units (GPU) bezeichnet, das Benutzererlebnis in virtuellen Desktops und entfernten Windows-Sitzungen verbessern kann. Wenn ein Hypervisor übrigens eine physische GPU partitioniert und mehreren virtuellen Maschinen arbeitsfähig zur Verfügung stellt, ist die Rede von virtuellen GPUs (vGPU).

Um deren Einfluss auf die Leistungsfähigkeit entfernter Desktops einzuschätzen, hilft ein Blick unter die Haube. Hier tummeln sich vor allem die Hersteller AMD, Intel und NVIDIA. Letzterer hat seine entsprechende Technologie schlicht "NVIDIA virtual GPU" (vGPU) getauft [1]. Derweil gab AMD dem Kind den Namen "MxGPU" [2] und Intel spricht von seiner "Graphics Virtualization Technology" (GVT-g) [3]. Wenngleich die Technik im Detail variiert, verfolgen alle dasselbe Ziel, nämlich Grafikkern an VMs durchzureichen, um die Darstellung virtueller Desktops zu verbessern.

Bauartbedingt eignen sich Grafikkern daneben auch sehr gut, um bestimmte mathematische Operationen spezieller Software effizient auszuführen, die sich gut parallelisieren lassen, etwa in der

Modellierung und Simulation. Werden GPUs hierzu zweckentfremdet, ist oftmals die Rede von General Purpose GPUs (GPGPU). Software, die davon Gebrauch macht, baut meist auf Frameworks wie die Open Computing Language (OpenCL) oder Open Accelerators (OpenACC) auf. NVIDIA unterstützt beide mit GPUs nach der hauseigenen Spezifikation der Compute Unified Device Architecture (CUDA). Solche Einsatzmöglichkeiten sollen im Folgenden jedoch nicht im Fokus stehen und wir wenden uns dem Einsatz von Grafikkarten für ihren angestammten Zweck zu.

Alle Komponenten im Blick

Dabei empfiehlt es sich, alle beteiligten Komponenten und ihren Einfluss auf jene Leistungsindikatoren genauer zu betrachten, die aus Sicht der Anwender eine gute Sitzungsqualität ausmachen. Insbesondere die CPU und GPU auf der Seite der Server spielen dabei eine wesentliche Rolle – neben dem Netzwerk und nicht zuletzt den eingesetzten Clients.

Die Beurteilung der resultierenden Sitzungsqualität ist nicht trivial, da jeder Benutzer eine sehr individuelle Wahrnehmung hat. Dies führt zu unterschiedlichen

und subjektiven Beurteilungen darüber, was gut, was gerade noch ausreichend und was inakzeptabel ist. Die Idealvorstellung ist die Darstellung aller grafischen Elemente eines Windows-Desktops in höchster Auflösung und ohne spürbare Verzögerungen bei der Interaktion.

CPU vs GPU

Die CPU ist als zentrale Prozessoreinheit verantwortlich für alle Computeraktivitäten und das Ausführen von Programm-instruktionen. Sie nutzt hierzu in der Regel mehrere Prozessorkerne, die unterschiedliche arithmetische und logische Operationen durchführen. Im Gegensatz dazu ist eine GPU spezialisiert auf grafische Aufgaben, wie beispielsweise die Farbe von Pixeln in komplexen 2D- oder 3D-Szenen zu berechnen und sie dann auf einem oder mehreren Bildschirmen auszugeben. Für solche Aufgaben greift die GPU auf hunderte oder gar tausende recht einfache Kerne zurück, die auch "Shader" genannt werden. Diese Shader sind optimiert für einen relativ kleinen Satz an Grafikberechnungen, die sie sehr schnell und energiesparend durchführen.

GPU muss sein

Bereits ab Windows Vista setzt der Window-Manager eine GPU voraus. Ist keine physische GPU vorhanden, dann emuliert die CPU eine virtuelle GPU. Das ist praktisch für Anwendungsentwickler, da der Window-Manager so nur eine einzige Code-Basis erfordert, die auf GPU-beschleunigten DirectX-Aufrufen basiert und trotzdem unabhängig davon funktioniert, ob eine physische GPU präsent ist.

Natürlich kann eine emulierte GPU niemals so leistungsfähig sein wie eine physische GPU – aber das System ist zumindest auch ohne GPU lauffähig. Fügen Sie zu einem PC mit einfachem Chipsatz ohne GPU eine Grafikkarte hinzu, verbessert dies zumeist die Geschwindigkeit von Windows-Desktop und Anwendungen um Größenordnungen.

Da heutzutage bereits Standardanwendungen in der Art des Microsoft-Office-Pakets ihre Oberfläche mithilfe von WPF-Steuer-elementen aufbauen, ist der Einsatz von GPUs längst zur dringenden Empfehlung geworden. Zumal im betrachteten Zeitraum die typischen Bildschirmauflösungen von ehemals VGA (640x480 Pixeln) bis hin zu Full HD oder gar 4K gewachsen sind.

Die Berechnung und die Einfärbung von individuellen Pixeln auf einer Bildschirmseite nennt sich "Rendering". Das Erzeugen der Pixeldaten erfolgt temporär in einem speziellen Grafikspeicher namens "Frame Buffer". Aufgrund der vielen ähnlichen grafischen Rechenoperationen lässt sich das Rendering einer ganzen Bildschirmseite sehr gut parallelisieren und somit auf eine große Zahl an Shadern verteilen.

Die Geschwindigkeit bei der Ausgabe von fertiggestellten Bildschirmseiten wird dann in der Einheit "Frames per Second" (FPS) gemessen. 30 bis 60 FPS gelten als gut genug für relativ anspruchslose Windows-Desktops. Für moderne Spiele, CAD-/CAM-Anwendungen und Virtual Reality dürfen es dagegen gerne 90 bis 120 FPS sein, um eine fließende und flimmerfreie Darstellung zu produzieren.

Dies alles gilt für konventionelle Windows-PCs oder Laptops im gleichen Maße wie für entfernte Sitzungen und virtuelle Desktops. Der einzige Unterschied besteht darin, dass sich im zweiten Fall zwischen Computer und Ausgabebildschirm eine Netzwerkverbindung befindet, die den Pixeldatenstrom der einzelnen Bildschirmseiten übertragen muss. Eine GPU beschleunigt jedoch in allen Fällen die Grafikausgabe, da ihre große Zahl an spezialisierten Shadern die Berechnung der vielen Pixel viel schneller erledigt, als eine CPU dies könnte.

Rendering zunächst auf den Clients

In den Anfangszeiten von entfernten Windows-Desktops versuchten die Pioniere wie Microsoft mit dem Remote Desktop Protocol (RDP) und Citrix mit der Independent Computing Architecture (ICA) noch, das Rendering von Bildelementen auf den Client zu verlagern. Solange ein Windows-Desktop mit seinen Anwendungsfenstern nur aus recht einfachen und zumeist statischen 2D-Grafikobjekten wie Linien, Rechtecken, Kreisen, Fensterrahmen und wenigen Schriftarten bestand, war das in Bezug auf die benötigte Netzwerkbandbreite auch effizient machbar. Ein Client mit installiertem Windows konnte das Rendering solcher über das Netzwerk

empfangenen Beschreibungen von Grafikobjekten mithilfe seiner lokalen Ressourcen und dem lokalen Frame Buffer durchführen. Die Last auf der Seite des Senders, des Desktop-Hosts, war dabei gering und erforderte keine GPU.

Was aber, wenn der Client kein Windows-Gerät ist? Und was, wenn immer mehr Windows-Anwendungen sehr dynamische Inhalte wie Videos oder Animationen anzeigen? Oder der Windows-Desktop verfügt über eine dritte Dimension mit Schattenwurf und Transparenzeffekten? Was ist, wenn die Bildschirmflächen und zugehörige Auflösungen immer größer werden? Alle diese Änderungen begannen fast gleichzeitig vor etwas weniger als 15 Jahren.

In dieser Zeit wurden Endgeräte von Apple und Videoplattformen wie YouTube immer populärer. Der Windows-Desktop-Manager und auch viele Anwendungen wechselten seit Einführung von Windows Vista und seinen Nachfolgern von einem 2D- zu einem 3D-Modell – konkret vom Graphics Device Interface (GDI) zur Windows Presentation Foundation (WPF). Letztere profitiert enorm vom Einsatz von GPUs.

All dies führte die Entwickler von Remote-Protokollen vor etwa zehn Jahren zu der Erkenntnis, entfernte Windows-Desktops wie interaktive Videos zu behandeln und damit auch das Übertragungsmodell zu ändern. Wer heute an einem Client mit geöffneter entfernter Benutzersitzung Maus oder Tastatur betätigt, verändert den empfangenen Videodatenstrom mit den Pixeln der Desktop-Oberfläche entsprechend.

Die technische Basis dafür bilden moderne Remote-Protokolle wie Microsoft RDP mit seinen RemoteFX-Erweiterungen, Citrix High Definition Experience (HDX) als Weiterentwicklung von ICA oder VMware Blast Extreme. Allen gemein ist, dass sie das Rendering komplett auf der Seite des Sitzungshosts durchführen und die Daten des Frame Buffers vor dem Abschieken wie ein Video komprimieren. Die hierzu nötigen Codecs, also Software und Algorithmen für das Komprimieren und Dekom-

primieren von hochauflösenden Videos, haben sich in den letzten Jahren signifikant verbessert und sind darüber hinaus völlig unabhängig vom Betriebssystem.

Auf den Clients dürfen entsprechend fast nach Belieben Windows, Linux, macOS oder mobile Vertreter wie iOS und Android zum Einsatz kommen – solange die passende Clientsoftware installiert ist und mit dem komplexen Zusammenspiel von Hard- und Software auf dem Host harmoniert. Die dort eingesetzten GPUs verfügen neben ihren Shadern auch über spezielle Video-Codec-Komponenten, die zur weiteren Entlastung der CPUs eingesetzt werden können.

Zeitdiebe auf der Strecke

An dieser Stelle drängt sich oftmals die benötigte Bandbreite beim Übertragen des Videodatenstroms als einzig maßgeblicher Faktor für die möglichst gute Qualität eines entfernten Desktops in den Vordergrund. Ausreichende Bandbreite oder möglichst starke Kompression ist jedoch nicht alles. Ein mindestens genauso wichtiger Faktor ist die Latenz, also die Zeit zwischen einer Benutzerinteraktion und der sichtbaren Reaktion auf dem Bildschirm.

Technisch gesehen wird die Latenz von der endlichen Geschwindigkeit definiert, mit der eine Nachricht oder ein Signal ein System durchläuft. In einem Netzwerk ist es die Zeit, die ein Datenpaket benötigt, um von einem Computerknoten zum anderen zu gelangen. Müssen die Daten sogar hin- und zurückfließen, dann spricht man von der Round Trip Time (RTT).

Latenzen von weniger als 20 Millisekunden sind für die meisten Benutzer nicht wahrnehmbar. Forschungsergebnisse zeigen jedoch, dass die Benutzerzufriedenheit ab einer Latenz von 75 Millisekunden anfängt nachzulassen [4].

Die Zeit, die Anwender benötigen, um eine Aufgabe am Desktop zu vervollständigen, erhöht sich messbar ab einer Latenz von 250 Millisekunden. Ab 400 Millisekunden entwickeln Benutzer eine lähmende Strategie des Abwartens bei der Interaktion mit dem Desktop. Für entfernte Desktops ist eine RTT von weniger als 50 Millise-

kunden wünschenswert, insbesondere für Remoting-Protokolle, die Prüfsummen hin- und herschicken. Steigt die Latenz, dann sind oftmals Effekte wie Bildartefakte, Farbfehler oder geringe Wiederholraten zu beobachten.

Zusätzliche Verzögerungen entstehen durch Signallaufzeiten von und zu Ein- und Ausgabegeräten, wie Maus, Tastatur und Monitor, beim Rendering sowie der Verarbeitung durch Codec und den Window-Manager. Diese können der RTT je nach Konstellation zwischen 25 und 250 Millisekunden hinzufügen. Abgesehen von den Signallaufzeiten hilft eine GPU hier bei der Reduktion von Latenz und sorgt damit für eine deutlich verbesserte Benutzererfahrung beim Einsatz von entfernten Desktops.

Abhilfe durch Hardware-GPUs

Wie aber lässt sich eine GPU in Kombination mit entfernten Desktops einsetzen? Im Grunde ist die Antwort recht einfach: Durch den Einbau spezieller GPU-Karten in die Server, auf denen entfernte Desktops bereitgestellt werden. Die Hosts können dabei entweder physische GPUs als Ganzes an virtuelle Maschinen durchreichen oder diese in logische vGPUs aufteilen.

AMD verfolgt dabei mit MxGPU einen hardwarenahen Ansatz und weist den virtuellen Desktops dedizierte physische Shader zu, die diese isoliert von anderen Desktops nutzen. Dies ermöglicht eine Technologie namens Single Root I/O Virtualization (SR-IOV), die auf BIOS-Ebene der Host-Hardware ein einzelnes PCIe-Gerät, in diesem Fall eine Grafikkarte, als mehrere Geräte erscheinen lässt. Ein Hypervisor kann diese dann VMs zuweisen, ohne dass dazu seine Software modifiziert werden müsste.

Demgegenüber implementieren Intel und NVIDIA ihre Lösungen in Software. Entsprechend muss jeweils eine zusätzliche Softwarekomponente für das Management der vGPUs auf der Ebene des Hypervisors installiert sein. Zertifizierte oder zumindest akzeptierte Erweiterungen sind für die gängigen Hypervisoren verfügbar, aber natürlich muss neben dem Virtualisie-

rungs-Host auch das Gastbetriebssystem für die virtuellen Maschinen mitspielen. Jeder der Hersteller hat mit seinem GPU-Ansatz für entfernte Desktops gewisse Vor- und Nachteile, doch alle erreichen eine Beschleunigung des Renderings und auch eine höhere Rate an FPS. Relevant ist dabei immer die Anzahl der Shader-Kerne und die Größe des Frame Buffers pro virtueller Maschine oder Benutzersitzung. Neuere GPUs verfügen auch über spezielle Codec-Einheiten, die von manchen Remoting-Protokollen genutzt werden können.

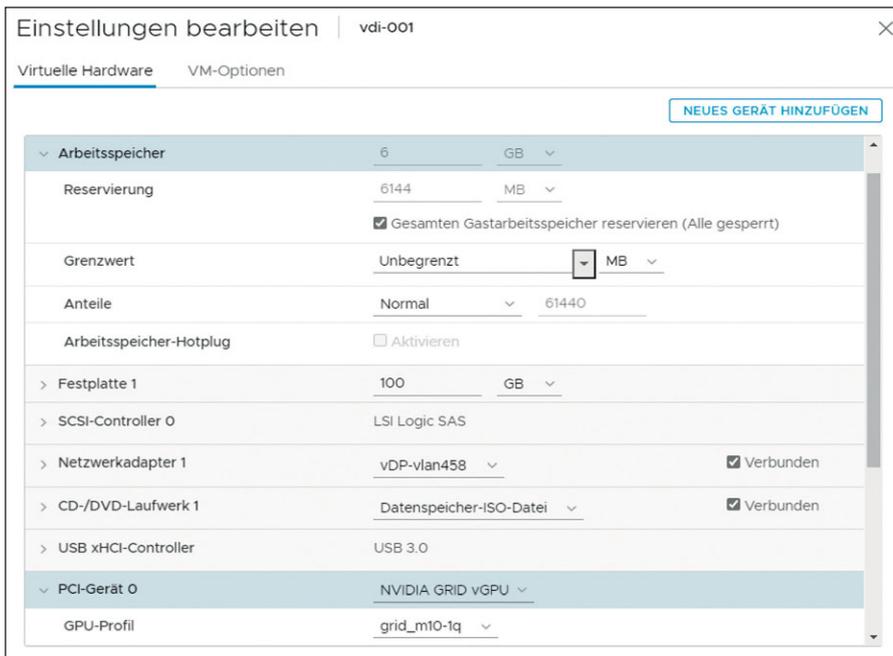
Fotorealistische Darstellung

Neben Shadern und Codec-Einheiten bringen NVIDIA-GPUs der neuesten Generation auch spezielle Einheiten für Raytracing mit. Die Idee bei Raytracing ist die Simulation einer hohen Zahl von Lichtstrahlen, um die Grafikqualität bei Spielen in die Nähe von Fotorealismus zu bringen. Bei entfernten Desktops spielt diese Technologie noch keine Rolle, was sich jedoch in den nächsten Jahren durchaus ändern könnte.

Um das Ganze mit Leben zu füllen, sind sämtliche Ebenen einer Virtualisierungsinfrastruktur zu betrachten – begonnen bei der Hardware der Hosts über den gewünschten Hypervisor, die Serverkomponenten der VDI, Gast-Betriebssysteme mitsamt VDI-Agenten und Treibern, Remote-Protokolle bis zu den Endgeräten mit deren Hardware, Betriebssystemen und VDI-Clients.

Versuch in einer Beispielumgebung

Wie komplex eine solche Umgebung ausfällt, sei an einem exemplarischen Szenario aus der Praxis erläutert. So wollten wir eine VDI auf Basis von Lenovo-Flex-Systemen um GPUs erweitern und hatten hierzu alle Compute-Nodes um Expansion-Nodes erweitert, die mittels einer PCIe-Bridge die einzelnen Server-Blades um NVIDIA-Tesla-M10-Grafikkarten erweitern. Auf der Ebene der Hardware war abgesehen vom physischen Einbau der Geräte denkbar wenig zu tun. In den Managementmodulen der Chassis und Blades konnten wir lediglich erkennen, dass die Systeme die GPUs eingebunden hatten.



vGPU-Profile bestimmen, wie viel Leistung und Frame-Buffer-Speicher virtuelle Maschinen beanspruchen dürfen.

Hardware und Lizenzen erforderlich

An dieser Stelle sei noch erwähnt, dass es mit der Anschaffung der GPU-Hardware allein nicht getan ist und NVIDIA je nach gewünschter Nutzung zusätzlich

noch obligatorische Lizenzen mitsamt passendem Support, Upgrade and Maintenance Program im Angebot hat. Wenngleich diese Lizenzen zwar alternativ zu einer Subscription auch in einem Perpetual-Modell erhältlich sind, zieht die zu-

gehörige Wartung jährliche Folgekosten nach sich.

NVIDIA offeriert verschiedene Lizenzen mit unterschiedlichem Funktionsumfang im Hinblick auf maximale Bildschirmauflösung, unterstützte Gast-Betriebssysteme oder auch GByte Frame Buffer pro Desktop [5]. Im einfachsten Fall reichen zur Bereitstellung virtueller Desktops Lizenzen vom Typ "GRID vPC". Wir entschieden uns stattdessen für die Variante "Quadro vDWS", die mehr Auswahl bei den vGPU-Profilen erlaubt.

Um die Lizenzen später in unserer Infrastruktur an die virtuellen Desktops zu verteilen, installierten wir NVIDIAS-Lizenzserver auf Basis der Software FlexLM, die unter der Haube noch eine Oracle Java Runtime mitbringt. Über das Lizenzportal im NVIDIA Application Hub [6] erhielten wir unsere Lizenzdateien und dort fanden sich auch die passenden Softwarepakete zur Integration in Hypervisor und Gastbetriebssystem. NVIDIA bietet hier eine breite Unterstützung. Neben VMware vSphere fanden wir auch Pakete für den

**Certified
Ethical Hacker**

**Unternehmen
gezielt schützen!**

Weil Cyber-Angriffe brandgefährlich sind.

Aktuelles Certified Ethical Hacker CEHv10 Training



- Tools und Methoden der Angreifer verstehen
- Scannen, testen, hacken, sichern
- International anerkannt
- Als Online Training, Präsenzkurs oder E-Learning

Weitere Infos: www.expertech.de/go/ECCE



EXPERTeatch

Citrix-Hypervisor, Microsoft Hyper-V, verschiedene Varianten von KVM sowie Nutanix AHV.

GPUs und Desktops integrieren

In unserem Fall sollte VMware vSphere als Hypervisor für virtuelle Desktops auf Basis von Citrix XenDesktop zum Einsatz kommen. Entsprechend installierten wir auf unseren ESXi-Hosts das NVIDIA vSphere Installation Bundle (VIB) [7]. Anschließend zeigte uns NVI-DIAs Kommandozeilen-Tool `nvidia-smi` die physischen GPU-Kerne mitsamt ihrem Framen-Buffer-Speicher. Eine entsprechende Auflistung fanden wir auch im VMware vCenter jeweils in der Konfiguration der Hosts wieder.

Sobald VMs anteilig Frame-Buffer-Speicher belegen, tauchen auch diese hier auf. Die entsprechende Zuweisung erledigten wir aber in den Eigenschaften der einzelnen VMs, wo wir jeweils ein neues PCI-Gerät vom Typ NVIDIA GRID vGPU hinzufügten und hierbei ein passendes vGPU-Profil wählten (siehe Bild). Die jeweils verfügbaren Profile hängen

vom Typ der Grafikkarte sowie der Lizenz ab und haben eher kryptische Namen, sodass die Onlinedokumentation von NVIDIA eine große Hilfe ist [8]. So eignet sich etwa das Profil "grid_m10-1q" für virtuelle Desktops und Workstations. Es erfordert dazu eine Lizenz des Typs Quadro vDWS, sichert einer VM 1 GByte Frame Buffer einer physischen M10-Grafikkarte zu und unterstützt Endgeräte mit bis zu vier Monitoren.

Erwähnenswert ist noch, dass zur Nutzung von vGPU der Arbeitsspeicher der VM fix reserviert sein muss und VMware diesen nicht mehr dynamisch verwalten darf. Weiterhin funktioniert der Zugriff per VMware-Remote-Console nicht mehr, sobald der vGPU-Treiber geladen ist. Letzteren mussten wir zu guter Letzt noch in den VMs installieren. Anschließend konnten wir uns auf dem NVIDIA-Lizenzserver davon überzeugen, dass die VMs sich dort mit ihrer virtuellen MAC-Adresse registrierten.

Mussten frühere Versionen des Citrix Virtual Delivery Agent (VDA) explizit mit Unterstützung für GPUs installiert werden, erkennen die Agenten aus aktuellen XenDesktop-Versionen die GPUs inzwischen dynamisch, sodass wir nur noch das Feintuning mittels Citrix-Richtlinien vornehmen mussten und anschließend mittels der Citrix Workspace App die verbesserte Grafikleistung sowohl an Windows-Clients als auch an Thin Clients unter Linux sowie auf iPads genießen konnten.

GPUs aus der Cloud

Doch dies war nur eines von zahllosen möglichen Szenarien. Aus der Menge der am Markt verfügbaren Hard- und Software ergibt sich eine nicht-triviale Matrix aus GPU-Herstellern und -Karten, Hypervisoren, VDI-Lösungen sowie Remote-Protokollen. Und die Qual der Wahl nimmt noch zu, da neben einem Aufbau vor Ort GPUs auch in der Cloud verfügbar sind. Angebote wie die Amazon Elastic Graphics [9] oder virtuelle Maschinen der N-Serie in der Microsoft Azure Cloud [10] bieten die Leistung von NVIDIA-Grafikkarten für nahezu jeden Anwendungsfall und Geldbeutel auf Abruf.

Performance messen

Nur Testreihen von Infrastruktur-Optionen und bevorzugten Anwendungstypen können helfen, die optimale Lösung für eine gewünschte Zielumgebung festzulegen. Doch gibt es dazu eine einzige Kennzahl, mit der sich die Qualität des Benutzererlebnisses in entfernten Desktops bestimmen ließe? Leider nicht. Die subjektiv empfundene Qualität setzt sich aus Bildwiederholrate, Bildauflösung, hoher Farbtreue, geringer Latenz bei der Interaktion, der Anzahl an unterstützten Grafikformaten, zeitlicher Synchronität bei multimedialen Inhalten (Audio/Video) und der Startzeit von Anwendungen zusammen. Dies alles erfordert massive Rechen- und Grafikleistung. Benchmarking-Werkzeuge, wie REX Analytics [11], wurden mit dem Ziel entwickelt, die Qualität von entfernten Desktops über einen Satz von Telemetriedaten und den korrelierten Bildschirmvideos zu ermitteln.

Bei solchen Benchmarks wird offensichtlich, dass sich durch den Einsatz von GPUs sowohl die wahrgenommene Qualität eines entfernten Desktops verbessert als auch eine nennenswert höhere Anzahl von gleichzeitigen Benutzersitzungen auf einer vorgegebenen Hardwareplattform ausgeführt werden kann. Der signifikante Unterschied zwischen Desktops mit Zugriff auf GPUs und solchen ohne wird deutlich sichtbar [12].

Fazit

Der Einsatz von GPUs reduziert die Last auf den CPUs der Sitzungshosts für virtuelle Desktops signifikant, indem die Grafikkommandos zur Visualisierung von gängigen Windows-Anwendungen durch die GPU ausgeführt werden. Zudem kann eine GPU Videos sehr effizient enkodieren und dekodieren. Einfach ausgedrückt verbessert die Übertragung von Grafikberechnungen und Video-Codern auf eine GPU das Benutzererlebnis in virtuellen Desktops immens. Somit haben GPUs im Rechenzentrum heutzutage ihre Daseinsberechtigung unabhängig vom Anwendungsszenario der Benutzer. Selbst grafisch vermeintliche anspruchsvolle Applikationen wie Microsoft Office profitieren davon. (dr)

Link-Codes

- [1] NVIDIA virtual GPU
k7p41
- [2] AMD MxGPU
k7p42
- [3] Intel Graphics
Virtualization Technology
k7p43
- [4] Benutzerzufriedenheit bei Latenzen
k7p44
- [5] NVIDIA-Lizenzen
k7p45
- [6] NVIDIA Application Hub
k7p46
- [7] NVIDIA vSphere Installation Bundle
k7p47
- [8] Onlinedokumentation von NVIDIA
k7p48
- [9] Amazon Elastic Graphics
k7p49
- [10] N-Serie in Microsoft Azure
k7p40
- [11] REX Analytics
k7p4a
- [12] Unterschied zwischen Desktops
mit und ohne GPUs
k7p4b