An overview of Remote 3D visualisation VDI technologies

As applied in the HPC, cloud and scientific / technical computing space

Authors: Dr. Paul Calleja, Stuart Rankin, Filippo Spiga, Cambridge HPCS

Table of Contents

Introduction

Recent developments in both hardware and software within the VDI and 3D graphics technology domains now make it possible for virtual machines to make full use of the 3D graphics card on server side and for this accelerated graphical desktop to be viewed on any remote client device. Whereas more traditional HPC and technical computing solutions based on Dell PowerEdge compute nodes driven by the powerful and energy-effiicient Intel® Xeon® processors provide an excellent platform to make compute power available to HPC users, this functionality is very useful within the scientific and technical computing domain since it can be used to greatly increase the productivity of scientists and engineers and other power users that need real time graphical access to the HPC systems and for downstream analysis of data within their workflow process.

Remote 3D VM increases HPC user productivity

This paper will look at how remote 3D virtual desktop technology can be applied in two usage modes within the scientific and technical computing arena. First, as an HPC remote visualisation platform deployed to provide real time graphical access to large scale data sets stored within the HPC data center. Second, to provide a full 3D capable virtual workstation solution as a replacement of traditional fixed under-desk workstation solutions.

VM recap

In the Virtual Desktop paradigm there are obviously two sides involved: a server-side (or data center side) often powered by Intel® Xeon® processors where all the compute power is aggregated and a client-side represented by remote workspaces supported by thin- or zero-clients. Hardware and software components are tightly coupled to deliver a usable remote graphical experience at application or desktop level. Remoting protocols are specifically designed to permit communication between peers in a reliable and fast way, preserving as far as possible the quality of the graphics.

Scope of the paper

This document aims to provide a broad overview of the wide range of potential software technology solutions that can be used to provide remote 3D VM functionality. We concentrate our study on solutions that can leverage (directly or indirectly) recent NVIDIA GPU technology capabilities. Currently, NVIDIA GPU technology is considered to be a de-facto standard when high-end 3D graphic applications are considered. A future paper will look at the rapidly developing Intel® graphics solutions space in this domain.

There are many possible technology solutions in this space. They all have a high degree of overlap, the field is rapidly developing, but there are some significant differences in the solutions which effect suitability for use within different usage environments. The document will develop a generic classification of graphics remoting technologies incorporating high-end 3D rendering

This classification is then used to structure a comparison exercise. The aim of this comparison exercise is to identify critical factors that influence the quality of experience from a user and service provider point of view.

The final output from this document is the suggestion of two remote 3D virtualisation POC platforms designed for suitability within the HPC and scientific / technical computing domains. These two POC platforms will be constructed and tested with representative use cases and the details written-up in a follow-on paper.

Key deliverables of the paper

- Provide background and overview of remote graphics technologies
- Introduce the concept of remote 3D virtual machines
- Describe remote 3D visualisation use-cases and test cases within the HPC and scientific and technical computing domain
- Develop a classification system for remote 3D virtual technology
- Use classification system to undertake a comparison of current remote 3D virtualisation technologies.
- Select a number of test remote 3D VM test platforms to take forward as POC and run representative user test cases for analysis. This will be written-up in a follow-on paper

A brief history

Early remote graphics solutions

Whereas the graphical user interfaces (GUIs) of early PC era were firmly confined to the hardware physically connected to the display, the X windows system adopted by UNIX workstation vendors at the same time was by design more flexible - an application with a GUI could execute on one machine, but display on and be controlled from another connected to the first over a network. In fact this was so easy to do that it constituted a serious security flaw, and no-one today would use this feature (all network connections to X displays should be protected by an encrypted tunnel, so that all connections become effectively local - SSH is commonly used, painlessly, for this purpose). Another problem with this design, however, is that it requires drawing commands to be sent over the network (and interpreted on the user side by the display software). For simple, 2D graphics with few colours this worked well, but the introduction of 3D capable graphics systems (i.e. systems able to generate complex images based on 3D geometry in real time) placed a far greater demand on the network connection (particularly for models involving many vertices and large textures). The need to encrypt these instructions for transmission through an SSH tunnel furthermore adds to the CPU load on each side.

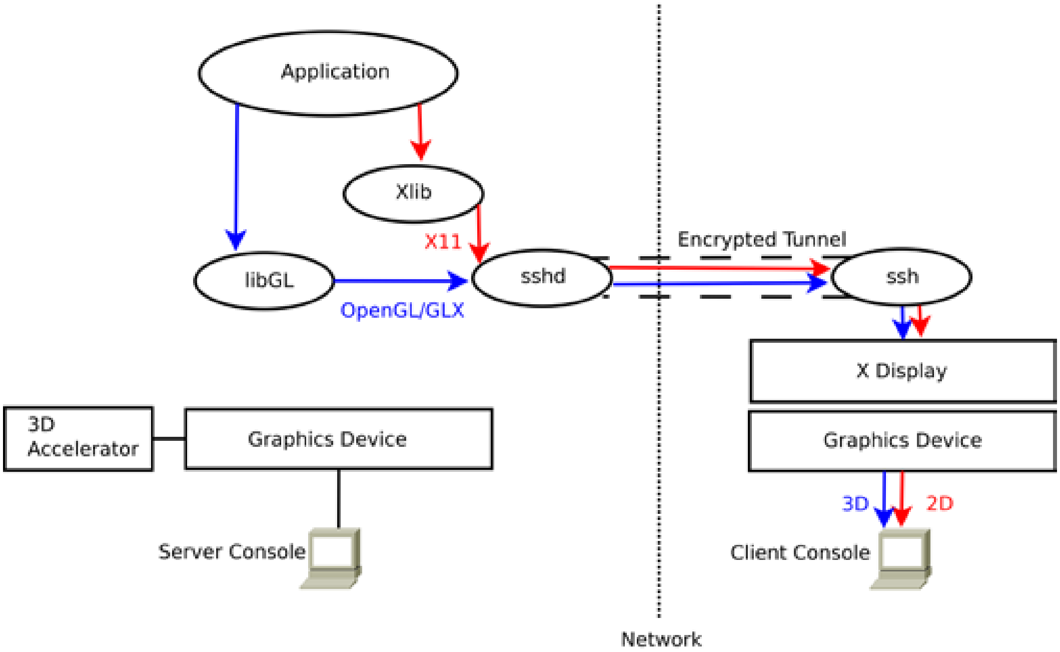

Figure 1: Indirect rendering with X Windows

The general setup for this mode of X windows is shown in Figure 1, which reveals some further limitations. Although the user's system is not necessarily a UNIX/Linux system (provided it has X windows software supporting the GLX extension required for 3D commands), the applications themselves must be written in OpenGL and X11; in practice this limits the server side to UNIX/Linux. More seriously, the presence of 3D graphics hardware on either side is wasted - on the server because the 3D commands are not interpreted there, and on the client because the X display system interprets the commands, and traditionally this is done entirely in software (without the aid of hardware acceleration). Because the X display, rather than the application, drives the rendering, this mode is known as indirect rendering. Finally, not all OpenGL applications are amenable to indirect rendering at all and assume the 3D device is local; such applications cannot be remotely displayed using these methods. Instead they must be operated from a local graphics console using direct rendering - in this case at least, the power of the 3D card can be harnessed (see Figure 2 below). This essentially describes the situation for UNIX/Linux 3D applications in the early 2000s.

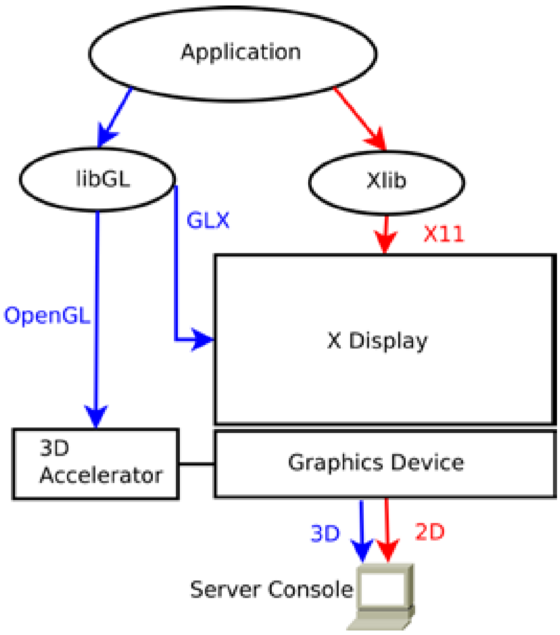

Figure 2: Direct rendering with X Windows

Remote desktops

In parallel to this UNIX/Linux-centric line of development, there was more O/S-agnostic progress in remote desktop technology. In this model, an entire graphical desktop environment running on a central server is displayed to a remote client, using some custom network protocol. In this design, the drawing commands are rendered on the server, thus pushing the heavy lifting back to the server room, and consequently the client machine need have no native support for the native graphics environment of the server, or special hardware features (it needs just a viewer application and enough CPU power to decode a stream of encoded pixels). This is illustrated, for a UNIX/Linux server although this is not essential, in figure 3. In this case the desktop itself is generated by a proxy X display process which accepts drawing requests from X applications, but instead of pumping pixels to a local framebuffer encodes and transmits them to one or more clients - in effect it provides a virtual X desktop. Examples of this are provided by the various flavours of VNC and also NX (from NoMachine). The corresponding component for a Windows server would be a screen-scraping process copying pixels back from the local framebuffer (VNC for Windows is again an example), or a Terminal Services Remote Desktop session.

Although there are appealing aspects of this design in terms of centralization and O/S flexibility, the obvious critical deficiency in figure 3 is the lack of 3D support. Not only is the server's 3D hardware not involved in the generation of the images sent to the client (this is equally true for a Windows screen-scraper) there is typically no support for 3D rendering of any kind (an exception is the NX proxy server which will render UNIX OpenGL applications in software, i.e. without hardware acceleration).

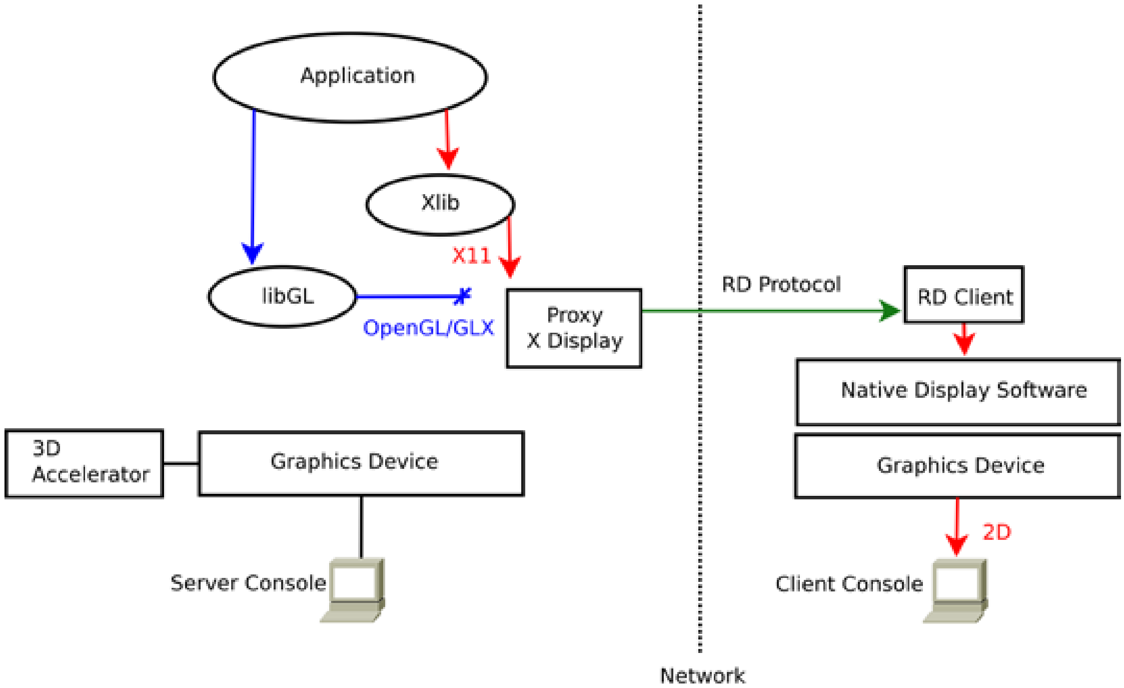

Figure 3: Remote graphics with remote desktop software

Towards a 3D remote desktop

In order for the user on the client side to usefully run 3D-graphics intensive applications on the server remotely, at least two key elements must be added to the remote desktop design illustrated in Figure 3:

A. The virtual desktop generated on the server requires a mechanism for acquiring service from the host server's 3D hardware in order to render on behalf of 3D applications. Ideally, several virtual desktops connected to different users would be able to share the resources of the GPU (this is a natural expectation in a UNIX/Linux context but potentially more troublesome under Windows).

B. An efficient compression/decompression mechanism is needed for the remoting protocol. Whereas a remote desktop without 3D images might only need to transmit rectangles containing a small number of colours which don't change rapidly, adding real-time 3D output places greater demands. Highly varying images of acceptable quality need to be transmitted by the protocol, while maintaining a sufficient level of interactive responsiveness for the user to consider the remote connection usable.

In the classification sections of the paper we survey the various approaches taken to fulfilling (A) and (B) on the way to providing a 3D-capable virtual desktop.

Remote 3D virtual machine HPC use cases

Remote 3D virtual machines have two main use cases within scientific and technical computing; remote visualisation of large data sets and for virtualised 3D workstations.

HPC remote visualisation

Scientific and technical HPC centers produce large amounts of data and this data is stored within the HPC data center which is often remote from the users. In order for the users to work with large data sets remotely without the need to transfer large data sets HPC data centers have been deploying remote Visualisation technologies for some time. (see previous Cambridge–Dell remote Visualisation paper to be found at www.dell.co.uk/hpc under Dell/Cambridge Solution Centre). This usage mode is becoming much more prevalent now that cloud computing solutions for HPC are starting to become more common . This issue of remote working with large scale data sets is seen as one of the barriers to uptake of cloud HPC thus technology improvements in this space are of great interest and are seen as enabling the cloud HPC ecosystem.

3D virtual workstations

Also we are now seeing a rapid increase in the number and breadth of HPC users. This is a result of the ubiquitous cloud HPC solutions and a function of a lowered cost of entry to HPC systems and an increase in the range of HPC applications becoming useful for a broad arena of research and product development workflow. With this rise in HPC users there is a rise in the number of power users needing access to high specification workstations to work with the outputs from HPC systems. For organisations with large numbers of such power users this presents problems in terms of workstation roll-out costs, management, user working flexibility / efficiency and data security. A full 3D virtual workstation would be of great interest in this regard since it can help address all these issues. With this solution there is a migration of computer hardware out of the local office environment and into the data center. Also there is a move away from fixed one person to one machine to shared back end infrastructure which can project a fully capable 3D workstation session to any device wherever the user may be (subject to network constraints). This has the potential to deliver both capital equipment and operation efficiencies in terms of hardware, but also presents greater flexibility for the user who can work in a much more flexible way.

Not a new idea

There is of course nothing new about separating users and CPUs (recall the mainframes and green-screen terminals of the 1970s-1980s). In addition to the enormous improvements in machine performance over the intervening decades, a major change took place in the late 1980s with the arrival of graphical user interfaces. This brought with it the desire to operate these interfaces from remote displays, a desire which naturally grew to include applications the output of which (as opposed to merely the control interface) is inherently graphical by nature (see the next section). The technical issues encountered when trying to make this model consistent with the current ecosystem of standard applications, and the particular operating systems for which they were written, has led to the complexity described in this overview.

Classification of remote 3D VM user test cases

Whether the use case is remote visualisation or 3D virtual workstation the usage requirements on the platform will be depend on the type of activity the user has undertaken. This usage requirement can be classified through example user test case scenarios.

It is natural to start thinking about how to partition the population under examination by considering several criteria:

- people who have a permanent office versus people who usually travel for business or work from many locations (including home);
- people who create new contents from scratch versus people who visualize or review those contents, applying only small changes;

- people who are permanent employees of the organization versus contractors hired for short periods of time;
- people who have a flexible workflow versus people who perform regular tasks during their working hours;

A classification based on too many criteria can generate a lot of sub-classes generating an over complicated view. Thus, we have decided to base usage classification on how demanding the user is likely to be and generate small range test cases that can be simulated with example workloads and used in quantifying the performance of test platforms under a range of user demand situations.

Based on this, relevant test cases for remote 3D graphics are as follows:

Scientist, Engineer or Designer:
A person who needs dedicated 3D GPU power because of demanding 3D applications. These applications can handle 3D models with millions of 3D elements, requiring rendering capabilities in real time and high fidelity graphical effects.

- Mainly Windows ISV applications running on high-end workstations.

Scientific or Engineering Manager or Field Engineer:
A person who usually visualizes models created by other engineers and eventually applies small changes but without necessarily re-rendering the full model. Sharing a slice of GPU resources is still acceptable, assuming that the resources are sufficient to avoid degradation of the overall Quality of Experience.

- Mainly Windows ISV applications running on workstation or high-end laptop.

Office Worker or Task worker:
Within the department, there are people who spend most of their time filling spreadsheet, drafting reports, preparing presentations and other typical office tasks. These people share GPU resources with graphic undemanding (but possibly 3D) applications.

- Windows desktop applications (e.g. Office suite) running on a PC desktop or a laptop.

Engineers still expect the best graphics capabilities delivered by a workstation even if it is not physically accessible. In this situation, technical constraints and additional requirements have to be considered (e.g. the latency of the network) during the design of the IT infrastructure.

The Quality of Service (QoS) is a quantitative measure of various performance metrics associated to a specific IT system (like the network bandwidth, the network latency, the average CPU load, etc). The Quality of Experience (QoE) is instead an indicator of the overall level of user satisfaction based on expectation. It is clear from the definition itself that the Quality of Experience is very subjective and very difficult to quantify.

The ultimate goal is to design an IT infrastructure, both at the software and hardware level that preserves the Quality of Experience of any category of users when the IT infrastructure changes from a one-to-one workstation to a so called “Workstation-as-a-Service”.

Classification of solutions by how the user resources are shared. (VM or no VM)

Current solutions can be usefully categorised according to the level at which resources are shared between user sessions, which is related to what form of virtualization a single user session receives. This amounts to identifying the (A) section on page 7 as the major distinguishing characteristic.

No sharing - dedicated non-virtual machine

This is the simplest, most obvious approach - dedicate an entire GPU-equipped bare-metal system to a single user who happens to be located somewhere else. For example, the Teradici host card into which is physically plugged the output from the graphics card. The host card then encodes the output of the native console session and transmits it over its own network connection to the client (using PCoIP in the case of Teradici). The obvious drawback is the 1-1 mapping between expensive servers and remote users but in principle it will work for any operating system/application combination supported by the server plus the GPU. In this case there is essentially no virtualisation.

Sharing the O/S - multiple user sessions on the same non-virtual machine

The same GPU-equipped bare-metal system supports multiple remote desktop sessions - this is the usual UNIX indirect rendering or VNC scenario, but in this case some way must also be found to share the full power of the GPU between the local user seated at the console and all remote users, and perhaps also to enhance the remote protocol transmitting the images the GPU generates. For historical reasons these techniques are mainly UNIX/Linux/OpenGL-only - examples are provided by the open source VirtualGL (VGL) and the closed source NICE Desktop Cloud Visualization (DCV). Citrix XenApp supports a similar model with multiple users using Terminal Services on a Windows bare-metal host, although each session consists of a single application rather than an entire desktop. In this case, there is virtualisation of either the application or of the entire desktop (but not of the operating system). This method has been used by many HPC centres for many years to allow remote graphics capability and has been written up in a previous Cambridge-Dell Solution Centre paper.

Sharing the bare metal - multiple virtual machines, one user session each

This is the most recently developed paradigm - a GPU-equipped bare-metal system hosts multiple virtual machines, each of which supports a single user session. Although virtualising the entire O/S seems likely to introduce significant additional overhead, for reasons of history this is the most natural model in a Windows context, and in principle facilitates compatibility, support and certifiability of standard industrial applications which expect to be run from a native Windows console. Hence this approach is being actively developed for use primarily with Windows applications. A virtual machine also provides a container in which each user is isolated from the behaviour (good or inadvertently bad) of others. Here, the operating system is virtualised, but there remains the question of whether the 3D GPU is virtualised, and if so, in what fashion.

Figure 4 summarizes these three ways of sharing graphic resources among users.

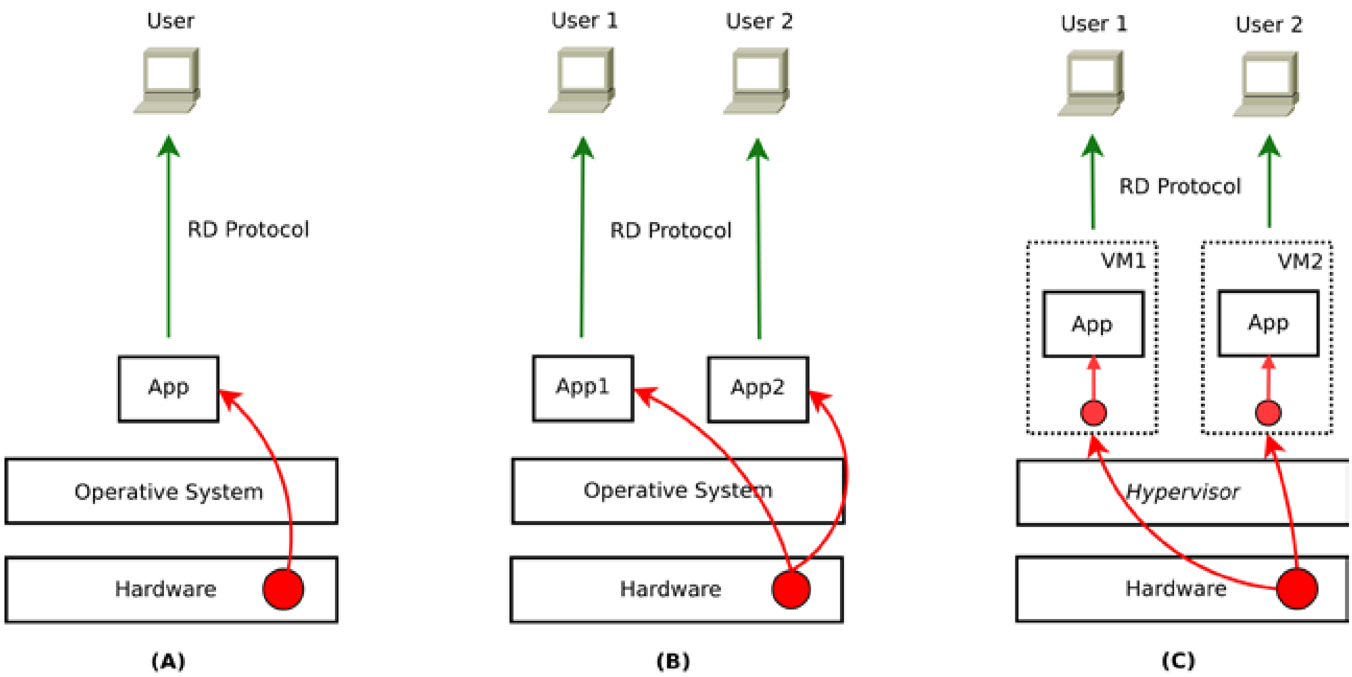

Figure 4: Classification by shared resources: A - No sharing; B - Sharing the O/S; C - sharing the bare-metal. The red circle represents a GPU device (A physical on in hardware, virtualized or software emulated in the VM).

Classification of virtual machine solution by how 3D graphics is passed to VM

An advantage of leveraging a virtual machine infrastructure is that since all virtual machines are inherently remote (there is no real machine at which to sit), a remote access protocol will already be supported (e.g. RFB, PCoIP, ICA) and client applications exist. These may require enhancement for use with 3D however, for the reasons already noted. This provides sub classification of the VM classification (again in section (A) on page 7)

The most interesting architectural variations arise from the need to expose the resources of the bare-metal 3D GPU to multiple virtual machines. Four different methods for exposing the 3D graphics resource to the VM are classified below:-

Library interposing/In-process forking

This approach (taken by NICE DCV) uses library loading path interception to redirect OpenGL calls made by an application running inside a virtual machine, over an internal network connection, to a specialised rendering virtual machine on the same bare-metal host with direct GPU pass-through access (see below) to the host's 3D card. This rendering host renders the 3D image, compresses the pixels and sends them over a direct connection to the remote client viewer. Each 3D window is sent over its own connection, whereas the non-3D parts of the desktop are transmitted to the client using a modified version of RealVNC. Multiple virtual machines can use the services of the rendering VM simultaneously. In this case, the GPU is virtualised for each VM but all 3D rendering is redirected to the bare-metal GPU at the application level.

The redirection of OpenGL calls is a similar mechanism to that used by VGL and NICE DCV within a single Linux server, and indeed the rendering server runs Linux although the other VMs may run Windows or Linux. The hypervisor is not directly involved. GPU resources are thus shared and the output aggressively compressed, at the price of interfering with application execution and possibly introducing support and certification issues.

Examples: NICE DCV coupled with KVM.

Software GPU sharing/Out-of-process forking

This approach allows each application to send all commands to a virtual graphics card presented by the VM, which passes the 3D commands down to the hypervisor. The hypervisor has a customised driver from the graphics card vendor and performs the rendering on behalf of all VMs. The rendered pixels when passed back to the VM must then be encoded and transmitted to the remote client by the remote protocol of the VM infrastructure. In this case, the GPU is virtualised in software for each VM but the 3D rendering is redirected to the bare-metal GPU in the hypervisor.

This method requires explicit GPU vendor support and also a custom driver for the virtual graphics card presented by the host to the guest operating system. This virtualisation in software of the GPU also introduces application compatibility and certification issues.

Examples: VMware View with vSGA (PCoIP), MS RemoteFX (RDP), NVIDIA VCA.

A hybrid example of the previous two cases is provided by Virtual Box and VirtualGL, which together can use a combination of software GPU in the VM and library interposing in the hypervisor to output 3D images to a TurboVNC desktop running on the bare metal host (which acts as the render server); in this way the entire VM desktop displaying 3D applications is transmitted to the VM user over a TurboVNC connection to the bare metal host. This approach can be used with either a Windows or Linux guest OS.

GPU Pass-through

This exposes the 3D card belonging to the bare-metal server directly to one of the virtual machines. That VM may then use the standard GPU vendor driver to exploit the GPU in the standard way. Again, the VM infrastructure must take care of the transmission of the rendered pixels to the end user.

The advantages here lie in the reduction in overheads in communicating directly with the GPU, which should lead to higher performance and full application compatibility and certifiability. The obvious disadvantage is that only one VM at a time may be attached to each 3D GPU.

Examples: Citrix XenServer, Citrix XenDesktop (ICA/HDX 3D pro), VMware View vith VDGA (PCoIP).

Hardware-assisted GPU Virtualization

Ideally, multiple VMs would be able to communicate using the standard GPU driver for their guest O/S to the host 3D card, with minimal overhead. This would in theory combine good performance with full application compatibility/certifiability. Once again, the VM infrastructure must be able to handle the compression of the encoded 3D outputs. This is achieved by virtualising the GPU with the assistance of special hardware in the bare-metal GPU which allows it to handle all rendering directly for all VMs.

This requires hardware support in the 3D card plus support in the hypervisor. Currently this is available only for NVIDIA Kepler (GRID) GPUs with Citrix XenServer or Citrix XenDesktop (using the ICA/HDX 3D pro protocol). This approach therefore shares the efficiency and application compatibility/certifiability of GPU pass-through but without suffering from the lack of scalability.

Examples: NVIDIA GRID vGPU with Citrix XenServer (ICA/HDX 3D pro).

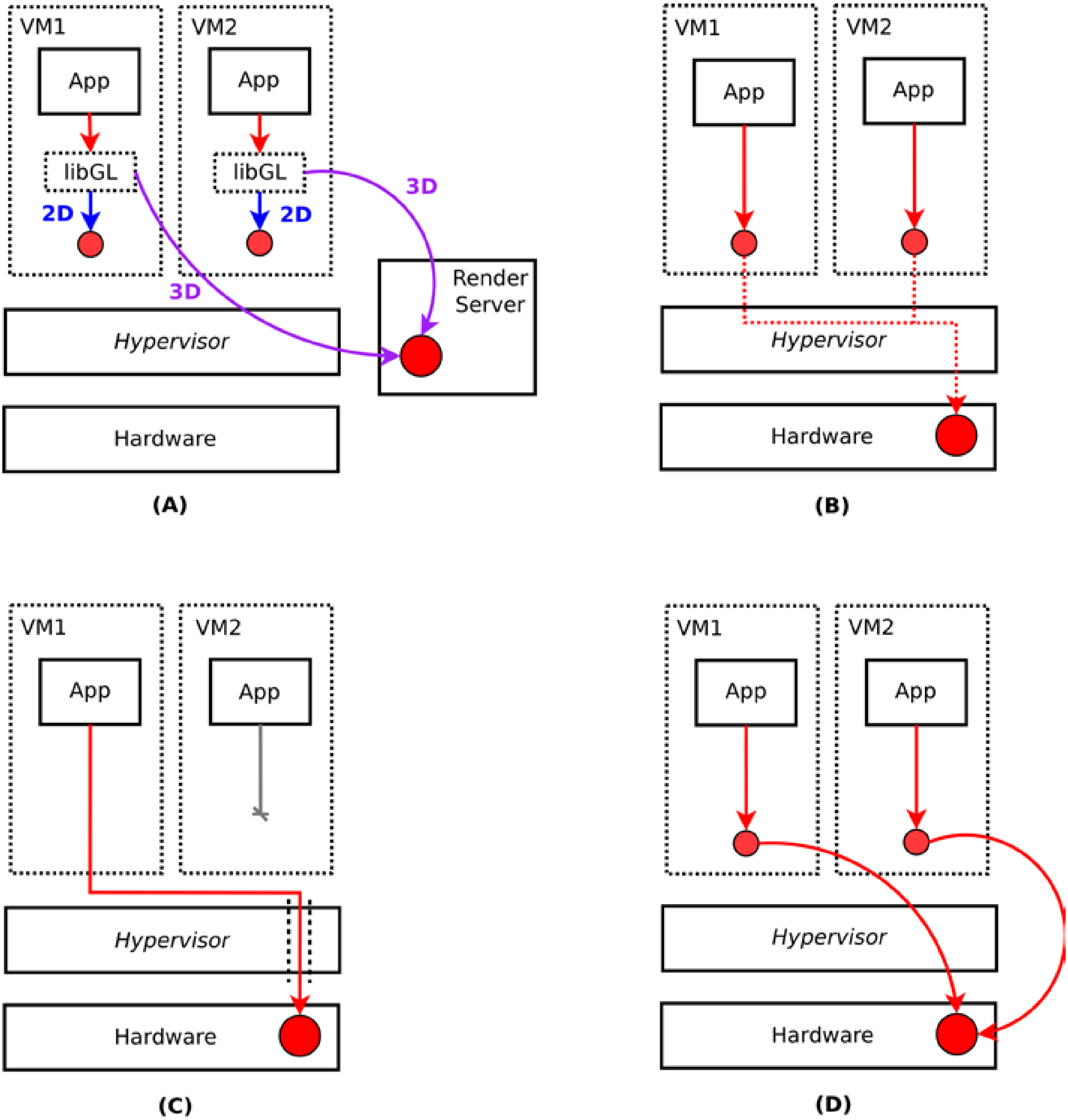

Figure 5: Classification by virtual machine: A - library interposing; B - software GPU sharing; C - GPU pass-through; D - Hardware-assisted GPU virtualization. The "big" red circle represents a physical GPU device, a "small" red circle represents a virtualized or software emulated GPU in the VM (Aphysical on in hardware, virtualized or software emulated in the VM).

Figure 5 summarizes these four ways of classifying virtual machine solutions accessing 3D graphics acceleration.

Classification of solution by remote protocols used

Remote Frame Buffer Protocol (RFB)

The Remote Frame Buffer (RFB) is a simple protocol for remote access to the graphical user interfaces that allows a client to view and control a window system on another computer. Because it works at the frame buffer level, RFB is applicable to all windowing systems and applications. This is section (B) on page 7.

The remote endpoint where the user sits (i.e. the display plus keyboard and/or pointer) is called the RFB client. The RFB server is where changes to the frame buffer originate (i.e. the windowing system and applications). The nature of RFB makes the client stateless. If a client disconnects from a given server and subsequently re-connects to that same server, the state of the user interface is preserved. Furthermore, a different client endpoint can be used to connect to the same RFB server (and two clients can also share the same session active on a RFB server). At the new endpoint, the user will see exactly the same graphical user interface as at the original endpoint

The IETF RFC 6143 [1] document describes the details of the protocol. The popular VNC (Virtual Network Computing) tool for remote visualization and remote desktop is based on RFB. It is relevant to note that the RFB protocol specification defines a weak authentication method [2]. Fortunately, many VNC implementations do include support for enhanced authentication and encryption.

Remote Desktop Protocol (RDP)

Based on the ITU-T T.128 application sharing protocol, the first version of RDP (tagged as version 4.0) was introduced by Microsoft with "Terminal Services", as a part of their product Windows NT Server. Despite the fact that RDP has been mainly developed and promoted by Microsoft, there are several non-Microsoft implementations of RDP that implement only a subset of the functionalities (i.e. FreeRDP [3]).

RDP is a TCP-based protocol and the official implementation includes several interesting features to enhance the remote user experience including audio redirection, port forwarding, and bandwidth usage tuning. By introducing the RemoteFX technology inside its virtualization portfolio [4], Microsoft added virtualized GPU support and host side encoding to RDP.

RDP provides data encryption by default but it does not provide authentication to verify the identity of a terminal server. Security of the communication channel is guaranteed by Transport Layer Security (TLS). A full description of the Microsoft RDP protocol is reported in this MSDN Technical document [5].

Pixel-Compression-over-Internet-Protocol (PCoIP)

Pixel-Compression-over-Internet-Protocol (PCoIP) is a UDP-based proprietary remoting protocol developed and licensed by Teradici [6]. PCoIP has been implemented both in software and in hardware (TERA chipsets). In-silicon PCoIP implementations guarantee a big advantage driving enterprise deployments toward a virtual desktop environment where user workstations are replaced by thin- or zero-clients.

By using PCoIP, images rendered on the server are, as usual, captured as pixels, compressed and encoded and then sent to the client for decryption and decompression. Depending on the image, different codecs are used during the encoding since different compression techniques differ in effectiveness (e.g. a more aggressive and cpu-intensive compression method might be required for high resolution colour images compared to just black and white text).

Due to the nature of the UDP network protocol, the effectiveness of PCoIP for delivering remote graphics strongly relies on the network link. In real scenarios, data packet loss can always occur (probably when crossing the WAN rather than the LAN). If the link Quality of Service is heavily degraded then the PCoIP protocol may encounter difficulties delivering a fluid experience. In low bandwidth environments it utilizes lossy compression where a highly compressed image is quickly delivered, followed by additional data to refine that image (a process called build to perceptually lossless).

Independent Computing Architecture (ICA)

Independent Computing Architecture (ICA) is another display protocol designed and promoted by Citrix.

Unlike frame-buffered protocols like VNC, ICA transmits high-level window display information, much like the X11 protocol, as opposed to purely graphical information. ICA has been designed with the main functionalities of RDP in mind, focusing at the presentation layer of the OSI model (Layer 6). Within the ICA protocol, virtual channels are used to pilot various functionalities, such as client drive mappings, video, keyboard stroke interception, etc. These virtual channels are multiplexed and the traffic is built on a single connection. It also handles data conversion, compression, decompression and bitmap caching. From a security point of view, ICA is compressed and encrypted by default.

Citrix developed another technology on top of ICA called HDX [7]. The aim of this technology is to provide a High Definition User Experience by extending the ICA protocol (doing adaptive tuning and smart adjustments based on application graphics demand). Citrix announced during Synergy 2011 [8] an ARM-based System-on-chip for thin-client implementing the ICA/HDX protocol, positioning this protocol as a true competitor of PCoIP.

Relevant hardware solutions for VDI

This section describes two key hardware technologies currently used in large VDI deployments.

Teradici

Teradici [9] is the inventor and trademark owner of PCoIP. Within its portfolio, Teradici offers a range of hardware solutions to address VDI needs.

Teradici directly commercializes PCIe boards for workstation and server equipped with in HW implementation of PCoIP (TERA and TERA2 chipsets). On the server side (TERA1202, TERA2240, TERA2220 [10]), the Teradici card captures video output directly from the graphic card and stream it through the network. On the client side (TERA1100, TERA2321, TERA2140 [11]), the latest generation chipset can deliver up to four DVI-D displays at 1920x1200, or dual 2560x1600 resolution.

OEMs offer thin clients based on both x86 and ARM platforms. Zero-clients instead are equipped only with a PCoIP chipset that handles entirely video, audio, input encoding/decoding. Several form factors are available for a wide number of OEMs (i.e. Dell, HP, CISCO, Fujitsu, LG, Samsung). It is important to note that the client endpoint can also be an application able to render a PCoIP data stream (i.e. VMware Horizon View Client).

Teradici provides a web-based called PCoIP Management Console, used to carry simple management tasks. This management console is able to collect statistics from the devices, monitors the health on the links, grouping remote hosts and apply simple host-client policies.

In 2008, VMware included Teradici's PCoIP protocol in the VMware View product line. To address multiple desktop environments running inside virtual machines sharing the same bare-metal, Teradici announced the Teradici Hardware Accelerator APEX 2800 offload card [12] in 2011. The Teradici APEX 2800 is a PCI-E standalone board capable of addressing compression and encryption of multiple audio/video fluxes. Teradici claims that this improves the consolidation ratio by up to 1.2-times in typical office workloads, improving up to 2-times in some specific cases.

By constantly monitoring the display activities of all the running virtual machines (VMs) on the server, the card is capable of handling up to 64 different display releasing CPU cycles to the VM. The APEX 2800 card requires specific drivers installed on both VM and the hypervisor. Only VMware ESXi is supported.

NVIDIA GRID Technology

NVIDIA, strong in its position in the game industry, seeks in the future of gaming a new business model called Game-as-a-Service (GaaS). This motivated the development of a new technology called NVIDIA GRID [13]. This technology offers the ability to offload graphics processing in virtualized environments, allowing multiple virtual machines to exploit the full 3D capabilities of the GPU directly, without hypervisor interference.

In order to work, this technology requires both hardware and software components. The hardware has also been engineered to handle the workload of multiple virtualized environments, allowing multiple users to share the GPU resources efficiently.

NVIDIA came out with an interesting classification of "video users" divided into three different user groups: *Designers or Engineers, Power Users and Knowledge Worker* [14]. While Designers or Engineers required full access to a true 3D graphics capable device and its capabilities, Power Users or Knowledge Workers can share GPU resources efficiently taking advantages of idle times in processing not so complex graphic pipelines. Based on this classification, they diversify their product line to match different usage demands.

During the GPU Technology Conference 2013, NVIDIA announced two new Kepler-based [15] boards called NVIDIA GRID K1 and K2. These are designed to enable high quality graphics in virtualized environments:

- The NVIDIA Grid K1 card is equipped with four entry-level Kepler GPUs (768 cores) and 16 GB of DDR3 on-board video memory. Power consumption peak is 130W.
- The NVIDIA Grid K2 card is equipped with two high-end Kepler GPUs (3,072 cores) and 8 GB of DDR5 on-board video memory. Power consumption peak is 225W.

Additional technical details can be found on the NVIDIA website [16]. Noticeable is the presence of a high-performance H.264 encoding engine on board.

In a virtual desktop environment, there are four ways to virtualize or expose GPU in a virtual OS:

- *GPU pass-through:* 1-to-1 dedicated model where one GPU is fully dedicated to a single user;
- *Shared GPU:* the virtualization is purely software;
- *Virtual User Sessions:* multi users in a single OS instance share the same GPU;
- *Virtual GPU (called GRID vGPU):* GPU virtualized at the hardware level made possible by coupling NVIDIA hardware and NVIDIA software.

The NVIDIA GRID vGPU is possible thanks to the combination of hardware and software. On the software side, two components are relevant:

- a NVIDIA driver installed at the hypervisor level (called the NVIDIA vGPU manager) that acts as a traffic-cop by brokering which (and how) the GPU can be accessed by a specific subset of running VMs;
- a NVIDIA driver installed inside the VM that now recognizes a virtual GPU as a NVIDIA GPU and delivers full graphics acceleration.

A software technology at driver level called User-Selectable Machines (USM) allows system administrators to configure the GRID boards to take on the characteristics of specific GPU segments (e.g. standard versus high-end Quadro). The purpose of the NVIDIA vGPU manager is to let every user fit the requested graphic profile.

A key point in deploying VDI solutions that can exploit NVIDIA GRID capabilities is to understand which type of card best suits the specific workload scenario and how many users can fit a single server unit. This is especially true if we consider very demanding 3D CAD/CAE graphical applications. An analysis of the quality of delivered graphics performance is needed before it is possible to quantify the number of users that can efficiently share the GPU resources. Both compute/render resources and total available memory are to be balanced across multiple users, i.e. having too many users share the same NVIDIA GRID cards means less memory in each user's video frame buffer (so less screen resolution). **The whole point of this current work is to design a test platform to build and test with representative user workload to examine just this situation.**

Current debate is focused on measuring the Quality of Experience, a metric that is expected to be correlated to how much difference the user feels (performance, responsiveness, latency, images quality) when performing daily working tasks with a remote graphic solution instead of a classical high-end workstation.

NVIDIA claims that, despite being virtualized, the GRID vGPU technology will deliver up to 95% of the bare-metal performance when the GPU-user ratio is 1:1. Sharing the GPU among users will reduce the performance for obvious reasons (the graphics pipeline is shared). However it has to be considered that rarely do all the users sharing a single GPU at a specific moment in time perform demanding tasks simultaneously so the balance between idle and active users helps to deliver graphics performance at the desired level.

During GTC Japan 2013 [17], NVIDIA released some details about screen resolutions supported versus number of users sharing the same GPU GRID board, reported on Table 1.

CARD	Physical GPU	Frame Buffer [MB]	Virtual Display Heads	Maximum Resolution	Maximum vGPUs	
					per GPU	per Board
GRID K2	2	4096	4	2560x1600	1	2
GRID K2	2	2048	4	2560x1600	2	4
GRID K2	2	1024	2	2560x1600	4	8
GRID K2	2	256	2	1920x1200	8	16
GRID K1	4	1024	2	2560x1600	4	16
GRID K1	4	256	2	1920x1200	8	32

Table 1: VGPU Graphics profiles (source: NVIDIA)

Today Cisco, Dell, IBM, HP Asus and Supermicro manufacture servers certified to host NVIDIA GRID cards in the datacenter environment running 24/7.

Relevant software solutions for VDI

VMware suite

VMware Horizon View is a commercial desktop-virtualization product developed by VMware, Inc. [18]. It leverages VMware vSphere as the virtual desktop host platform; in other words, user desktops are virtual machines running on ESXi hosts. VMware vSphere includes the ESXi hypervisor as well as VMware vCenter Server which is used to configure and manage VMware hosts. Along with many other tools, VMware ThinApp also provides application virtualization.

VMware leverages graphics acceleration in virtual environments using 3 different technologies:

- SVGA/Soft 3D (Software 3D Renderer): a VMware WDDM (Windows Display Driver Model) 1.1-compliant driver is responsible for rendering any 2D or 3D graphics inside the virtual machine in software;
- vDGA (Virtual Dedicated Graphics Acceleration) graphics acceleration is provided directly by the ESXi hypervisor using a dedicated GPU accessible via pass-through;
- vSGA (Virtual Shared Graphics Acceleration): multiple virtual machines share physical GPUs installed locally in the hosts system to provide hardware-accelerated 3D graphics to multiple virtual desktops.

As vSGA allows multiple virtual machines to share hardware GPUs for 3D acceleration rather than requiring a one-to-one relationship like vDGA, the number of concurrent virtual machines running on the same host server has to be carefully determined. In fact, if all GPU hardware resources are already reserved, additional virtual machines will be unable to power on if they are explicitly set to use hardware 3D rendering.

Client connections to VMware View sessions are established by running a VMware View client on a local host. PCoIP or RDP are the two protocols supported. Clients are available for Windows, Mac OS X and several thin-clients. A tool called Ericom AccessNow [19] enables users to connect to their remote desktops directly from any HTML5 compatible browser (but only the RDP protocol is supported).

With respect to NVIDIA GRID features supported by VMware software solutions, vSGA is currently available on View VMware 5.2 (ESXi 5.1) and vDGA will be available in one of the next releases (VMware 5.x and ESXi 5.1). It is important to point out that VMware ESXi supports GPU pass-through and it can be used as a hypervisor together with other vendor solutions such as Citrix. VMware has made no public plans about supporting NVIDIA GRID vGPU.

Citrix Suite

Citrix XenDesktop [21] is a fully integrated desktop virtualization system. It is available in a range of packages designed for several market segments. The Express Edition of XenDesktop is offered as a free. The Enterprise and Platinum Editions of XenDesktop include all of its major virtual desktop delivery models in one integrated package, and also include the capabilities of Citrix XenApp for delivering on-demand applications into virtual or physical desktops.

All versions of XenDesktop include Citrix HDX technologies. XenServer is a commercial product from Citrix that is built from Xen [20] and XCP (Xen Cloud Platform, a bare-metal installable ISO with a full Xen hypervisor virtualization stack). Xen is currently available for the IA-32, x86-64 and ARM computer architectures. Citrix extended the capabilities and the functionalities of XenServer raising it to an enterprise product [22].

In the Citrix world, Citrix XenServer is the preferred hypervisor to host XenDesktop sessions. Both products evolved together to deliver the best quality of experience to the final users, leveraging the ICA protocol and Citrix HDX 3D technology. However other hypervisors can be adopted on the bare-metal side, including Microsoft HyperV or VMware vSphere/EXSi.

Alternatively to Citrix XenServer, Citrix XenApp is an application virtualization solution that allows enterprise applications to be centralized and delivered as an on-demand service. XenApp delivers applications running on a central server using its HDX technologies to securely transmit the application’s graphic interface to the endpoint device.

With respect to of NVIDIA GRID features supported by Citrix software solutions, GPU pass-though is fully supported by XenDesktop 5.6 FP1 (coupled with XenServer 6 or VMware EXS 5.1). Citrix will officially supports NVIDIA vGPU GRID in XenDesktop 7 using XenServer, available by the end of this year [23]. XenApp 6.5 FP2 (renamed XenDesktop 7 RDS) already fully support GPU acceleration and Citrix HDX 3D Pro.

NVIDIA GRID Visual Computing Appliance

NVIDIA GRID Visual Computing Appliance [24] is a one-box system with its own pre-configured stack software to expose application virtualization (or application on-demand) for a limited number of users. Based on the current specifications, the aim of this appliance is to pack eight high-end graphical workstations for 8 concurrent Power Users or Engineers/Designers in a small rack space.

The box consists of a dual Intel® Xeon® Processors (16 cores), 256 GByte of RAM, up to 8 high-end NVIDIA Kepler GPUs, two 10 Gigabit network interfaces and some internal storage capabilities to hold VM and application data all packed into a 4U rack format. Storage hardware has to be provided externally as separate infrastructure on which to store the private data of any user authorized to run on the system.

The software stack has been rewritten by NVIDIA leveraging all aspect of their GRID technology, from the hypervisor technology to the remoting protocol (based Citrix XenServer solution). From a system administrator perspective, the appliance works out-of-the-box by simply entering a network domain and network address. A template image based on Windows 7 is preloaded in the machine and, after customizations, every client connects to the machine will start a session based on this template image. A license server is also provided.

NVIDIA already engaged with ISVs to certify professional graphic suites to work without any additional effort at the installation phase. The initial set of software ready and certified to work on the appliance are in the area of media and entertainment, architecture and engineering construction (i.e. CAD), filmmaking and visual effects.

This solution is clearly not as scalable as the combination of NVIDIA GPU GRID cards equipped inside OEM servers. In fact, it has been explicitly designed targeting SMEs that do not want to care about configuring and maintaining a virtual software environment or simply manage a on-demand business often using external contractors working in-house or remotely. The choice to not integrate any storage sub-system to store users data helps maintenance in a way that the entire appliance is replaceable all-in-one in case of malfunctioning without losing any sensible data.

Microsoft Remote Desktop Suite

Microsoft RemoteFX [25] is a Microsoft brand acquired from Calista Technologies that covers a set of technologies that enhance the visual experience of the Microsoft developed remote display protocol Remote Desktop Protocol (RDP). RemoteFX is not a new standalone product from Microsoft. Rather, it describes a set of RDP technologies - most prominently graphics virtualization and the use of advanced codes - that are being added starting from Windows Server 2008 [26].

In Windows Server 2008 R2, the utility Terminal Services (TS) has been improved and renamed to Remote Desktop Services (RDS). RDS is the delivery vehicle of Microsoft RemoteApp programs and VDI solutions. In the latest windows server 2012 version, when a GPU is present in the server, it can be used to hardware accelerate the graphics via the RemoteFX vGPU feature. When no GPU is present in the server, a synthetic software-emulated GPU is used to render content. Supporting virtualizing a GPU in the virtual environment provides an accelerated DirectX graphics experience for 3D or other graphics-intensive applications. However, Microsoft states that only certain OpenGL applications are supported.

RemoteFX vGPU is the ability for a virtualized instance to access a physical GPU enabling full 3D hardware-acceleration for any application running inside the VM. The RemoteFX vGPU requires Hyper-V. RemoteFX and NVIDIA GRID software together enable multiple virtual desktops to share a single GPU on a Hyper-V server, saving the significant cost of deploying a graphics card to every user.

Like VMware and Citrix, a tool called RemoteApp [27] makes it possible to deliver remote applications to users instead of full desktops leveraging advanced features like Full Single Sign-On and seem-less disconnect and reconnect.

NICE DCV

Originated in IBM Research in 2004 then acquired by NICE in summer 2010, NICE DCV (acronyms of Desktop Cloud Visualization, [28]) is a display-agnostic remote 3D visualization technology that enables users to connect to OpenGL applications running remotely in a data center. By coupling DCV technology with a virtual environment, a virtual desktop experience fully graphic accelerated is delivered to the end users.

NICE DCV works on Windows and Linux platform, leveraging 3D acceleration by library interposing. DCV 'injects' itself between application and the real OpenGL library and is 'perceived' by the application as the system-wide OpenGL library. 3D rendered images are streamed to a remote client using DCV proprietary protocol, 2D images using a common RFB protocol. For this reason

DCV requires a VNC server installed on the remote server. It supports GPU sharing across multiple users and multiple OS. Probably it is the first product on the market to allow the sharing of physical GPUs between multiple Windows VMs while maintaining full OpenGL acceleration and workstation-class performance).

NICE DCV operates in two modes:

- Display Isolation Mode: virtual X11 desktops on the application host for remote viewing. From an end station, you can run an application on a remote desktop using hardware-accelerated 3D rendering. All virtual sessions remain separate and independent of one-another (Linux hosts only)
- Native Display Mode: the application host sends compressed images to the remote DCV end stations. It can be used in a single connection with one end station or in a collaborative configuration with multiple end stations sharing the same session (Linux and Windows)

Another interesting operational mode allows to designate rendering hosts. Each render host can receive OpenGL commands from the applications running on the application servers and, after the render process, send the 3D image updates (in the form of pixel data) to each connected end station using their DCV proprietary protocol. This is a powerful capability because it enables any virtual machine to act as full 3D-accelerated application host even if the virtual hardware emulated by the hypervisor (any hypervisor) does not provide OpenGL acceleration.

DCV protocol capabilities include toggling frames per second and modifying the quality of the image depending on the quality of the connection and the user's level of interaction. In a DCV session, the application user interface is transmitted to an end station using RealVNC Visualization Edition (RealVNC VE, [29]). Other VNC products (whether from RealVNC or another vendor) are not supported. RealVNC VE is an enhanced version of VNC Enterprise Edition designed exclusively for DCV. Because of this strong dependency DCV's available operation modes depend on RealVNC configuration.

NICE DCV is perfectly integrated with NICE EnginFrame [30], a web-based HPC portal, which acts as session manager and broker. At the time of writing, there are no public indicators that NICE DCV is planning to directly exploit the new NVIDIA GRID capabilities.

Comparison Exercise

The goal of this section is to perform a graphical comparison exercise by highlighting interesting features of various hardware plus software solutions already introduced in the previous sections. At the end, by considering all the factors, a subset of preferred solutions is chosen.

The comparison has been carried out by looking at public material on the Internet, from official documentation to third-party whitepapers, from recorded presentations of official events to professional blogs. Many resources have been considered, isolating the technical information.

We show four different comparison tables each one targeting a specific aspect: shared GPU resources in a virtual environment, remoting protocols, graphic capabilities (OpenGL and DirectX) and supported OSes.

Comparison by shared GPU resources in a virtual environment

	Teradici	NVIDIA VCA	VMware suite	Citrix suite	Microsoft suite	NICE DCV	VirtualGL
T2.1 - Does it support sharing the GPU between users?	N	N	Y	Y	Y	Y	Y
T2.2 - Does GPU sharing allow hardware-accelerated 3D?	-	-	Y	Y	Y	Y	Y
T2.3 - Does it work in a virtualized environment?	Y[1]	-	Y	Y	Y	Y	Y[2]
T2.4 - Does it support GPU pass-through mode?	-	Y	Y	Y	Y	Y	-
T2.5 - Does it support the new VGX technology?	N	N	N	Y[3]	N	N	-
T2.6 - Does it exploit NVIDIA GRID H.264 hardware encoder?	N	?	N	N	N	N	-

Table 2: Comparison by shared gpu resources in a virtual environment

[1] Yes but only in the VMware world using APEX cards.

[2] In principle should work if the VM has pass-through or GRID capabilities

[3] Formally announced but not yet publicly released.

Comparison by Remoting Protocols

	Teradici (PCoIP)	NVIDIA VCA	VMware suite (PCoIP)	Citrix suite (HDX)	Microsoft suite (RDP)	NICE DCV (DCV+VNC)	VirtualGL (VNC)
T4.1 - Is the RD protocol used proprietary?	Y	Y	Y	Y	Y	Y	N
T4.2 - Does it support VNC client (N, 2D, 3D)?	2D	N	2D	2D	N	3D[4]	3D[5]
T4.3 - Is the desktop session accessible through a web interface?	?	Y	Y	Y	?	Y	Y
T4.4 - Is a remote session accessible through a thin-client?	Y	Y	Y	Y	Y	Y	Y
T4.5 - Is remote session accessible through a zero-client?	Y	N	Y	N	N	N	N
T4.6 - Is remote session accessible through an application?	Y	Y	Y	Y	Y	Y	Y
T4.7 - Is there the possibility of compensating in case of low bandwidth network link?	Y	Y	Y	Y	Y	Y	Y
T4.8 - Is there the possibility of tuning the video quality?	Y	Y	Y	Y	Y	Y	Y
T4.9 - Does it implement a self-tuning proprietary transmission protocol?	Y	Y	Y	Y	Y	Y	N
T4.10 - Does it allow sharing the same remote desktop/remote view?	N	N	Y	Y	?	Y	Y
T4.11 - Does it support multi-screen mode?	Y	Y	Y	Y	Y	Y	N
T4.12 - Does it support audio?	-	Y	Y	Y	Y	N	-
T4.13 - Does it support connections to local printers?	-	?	Y	Y	Y	Y	-
T4.14 - Does it support connections to local USB drives?	-	?	Y	Y	Y	Y	-
T4.15 - Does it offload remoting protocol encoding to a dedicated hardware?	Y	N	Y[6]	N	N	N	N

Table 3: Comparison by remoting protocols

[4] Yes but only in the VMware world using APEX cards.
[5] In principle should work if the VM has pass-through or GRID capabilities
[6] Formally announced but not yet publicly released.

Comparison by OpenGL/DirectX support

	Teradici	NVIDIA VCA	VMware vSGA	VMware vDGA	Citrix suite (pass through)	Microsoft suite	NICE DCV	VirtualGL
T3.1 - Does it support OpenGL 2.x acceleration?	Y	Y	Y	Y	Y	Y	Y	
T3.2 - Does it support OpenGL 3.x acceleration?	Y	Y	N	Y	Y	N	Y	Y
T3.3 - Does it support OpenGL 4.x acceleration?	Y	Y	N	Y	Y	N	Y	Y
T3.4 - Does it support DirectX 9 acceleration?	Y	Y	Y	Y	Y	Y	?	
T3.5 - Does it support DirectX 10 acceleration?	Y	Y	N	Y	Y	Y	Y	?
T3.6 - Does it support DirectX 11 acceleration?	Y	Y	N	Y	Y	Y	Y	?
T3.7 - Is it easy to certify any 3D (OpenGL) ISV application?	Y	?	Y	Y	N	N	N	
T3.8 - Is it easy to certify any 3D (Direct X) ISV application?	Y	?	Y	Y	?	N	-	

Table 4: Comparison by openGL/direct X support

Comparison by supported OS-es

	Teradici	NVIDIA VCA	VMware suite vSGA	VMware suite vDGA	Citrix suite	Microsoft suite	NICE DCV	VirtualGL
T5.1 - Is the server-side Windows compatible?	Y	-[7]	N	N	Y	Y	Y	Y
T5.2 - Is the server-side Linux compatible?	N	-	Y	Y	Y	N	Y	Y
T5.3 - Are Windows VM supported?	Y	-	Y	Y	Y	Y	Y	-
T5.4 - Are Linux VM supported?	N	-	N	Y	N[8]	N	Y	-
T5.5 - Is there a client-side receiver Windows compatible?	Y	Y	Y	Y	Y	Y	Y	Y
T5.6 – Is there a client-side receiver Linux compatible?	Y	Y	Y	Y	Y	N[9]	Y	Y
T5.7 - Is it the source code available?	-	N	N	N	N	N	N	Y

[7] Since NVIDIA VCA is sold as an appliance, there is no direct control over the operative system installed by default.

[8] The support is not official.

[9] There are open-source projects but not officially supported and not able to handle all RemoteFX extensions.

Conclusion – choosing a 3D remote VM POC platform

The investigative work carried out during the writing of this paper underlined that many of the most common software solutions are today able to exploit 3D graphic acceleration using pass-through mode but only one (Citrix with XenServer/XenDesktop) has officially announced that proper GPU hardware virtualization will be supported. However, we are expecting all the major players mentioned in this report to support the vGPU NVIDIA technology in the future.

The differences between products are very few and there is a general lack of realistic benchmark data and reference performance numbers. It is difficult on paper to identify which solutions are better than others because all the solutions are very generic and applicable to a broad range of scenarios. In our case, we are looking to match both software and hardware capabilities that provide high performance cost effective solutions that work with a wide range of applications support full GPU sharing suitable for remote 3D workstations for power users and also remote cloud access to HPC infrastructure. The only way to really access the suitability of a potential solution is to build it and test under a variety of representative test cases, which is what we intend to do for the next stage of this work.

Our investigation has also underlined the possibility of mixing different software components. For example, it is possible to couple Microsoft RemoteFX with Citrix HDX Pro or install NICE DCV inside a virtual machine using a hypervisor other than KVM, e.g. VMware vSphere ESXi, or to use Citrix XenDesktop to remote the OS. On the other hand, there are clear constraints. For example, the locking of hardware solutions based on Teradici to VMware products or the inability to customize the software stack of the NVIDIA Visual Computing Appliance.

When looking at the results of the feature comparison exercise we have chosen two platforms to take forward which seem to best meet the requirements of a wide application base scientific and engineering workload environment

- *NVIDIA GRID GPU (K1 or K2), a hypervisor (to be selected), NICE DCV;*
- *NVIDIA GRID GPU (K1 or K2), Citrix XenServer, Citrix XenDesktop.*

On the server-side, the design of a proper server host will be crucial. Leveraging multiple VMs running simultaneously multiple applications will require a balance between graphics, cpu and storage. We will learn from the experience where and why a component can be a bottleneck for the entire system.

From this point forward, the focus should be mainly on quantifying the capabilities of the remoting protocols. This evaluation has to be focused not only on the Quality of Service (by producing comparable metrics) but also on the Quality of Experience. It is our understanding that the remoting protocols have the bigger impact in delivery of a good user experience. The graphics hardware is important too but since we have a single choice of professional graphic hardware (NVIDIA GPU), we rely on the ability of various hypervisors to access the GPU capabilities and expose them to various VMs, in shared or non-shared mode.

We will now build the target POC platforms and test with a range of use cases this will be written up in a follow-on paper.

Resources and References

[1] T. Richardson, J. Levine, The Remote Framebuffer Protocol, http://tools.ietf.org/html/rfc6143

[2] Dragon research Group website, VNC: Threats and Countermeasures, https://www.dragonresearchgroup.org/insight/vnc-tac.html

[3] FreeRDP, http://www.freerdp.com/

[4] Announced at Microsoft TechEd 2012 Premiere event, http://blogs.msdn.com/b/rds/archive/2012/06/13/overview-videos-for-remote-desktop-services-in-windows-server-2012.aspx

[5] [MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, http://msdn.microsoft.com/en-us/library/cc240445.aspx

[6] PCoIP technology explained, http://www.teradici.com/pcoip-technology.php

[7] Citrix white paper, Citrix HDX technologies, http://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-hdx-technologies.pdf

[8] HDX Ready System on Chip, http://hdx.citrix.com/hdxsoc

[9] Teradici, http://www.teradici.com

[10] PCoIP workstation solutions specifications, http://www.teradici.com/pcoip-solutions/workstation-solutions/specifications.php

[11] PCoIP zero clients specifications, http://www.teradici.com/pcoip-solutions/zero-clients/specifications.php

[12] Teradici PCoIP Hardware Accelerator (APEX 2800), http://www.teradici.com/pcoip-solutions/hardware-accelerator/overview.php

[13] NVIDIA GRID, http://www.nvidia.com/object/enterprise-virtualization.html

[14] SIGGRAPH 2013 presentation, NVIDIA GRID: A True PC Experience for Everyone Anywhere, http://on-demand.gputechconf.com/siggraph/2013/presentation/SG3112-GRID-State-of-the-Art-Virtualized-Graphics.pdf

[15] NVIDIA Kepler architecture, http://www.nvidia.com/object/nvidia-kepler.html

[16] NVIDIA GRID GPU, http://www.nvidia.com/object/grid-boards.html

[17] GPU Technology Conference Japan 2013, http://www.nvidia.com/object/grid-boards.html (japanese)

[18] VMware Horizon View, http://www.VMware.com/products/horizon-view

[19] Ericom AccessNow, http://www.ericom.com/html5_client_VMware_view.asp

[20] Xen project, http://www.xenproject.org/

[21] Citrix XenDesktop, http://www.citrix.com/products/xendesktop/overview.html

[22] XCP/XenServer Feature Matrix, http://wiki.xen.org/wiki/XCP/XenServer_Feature_Matrix

[23] Citrix blog, Preparing for true hardware GPU sharing for VDI with XenServer, XenDesktop and NVIDIA GRID, http://blogs.citrix.com/2013/08/26/preparing-for-true-hardware-gpu-sharing-for-vdi-with-xenserver-xendesktop-and-nvidia-grid/

[24] NVIDIA Visual Computing Appliance (VCA), http://www.nvidia.com/object/visual-computing-appliance.html

[25] Microsoft RemoteFX, http://technet.microsoft.com/en-us/library/ff817578(v=ws.10).aspx

[26] [MS-RDPRFX]: Remote Desktop Protocol: RemoteFX Codec Extension, http://msdn.microsoft.com/en-us/library/ff635423.aspx

[27] Overview of RemoteApp, http://technet.microsoft.com/en-us/library/cc755055.aspx

[28] NICE DCV, http://www.nice-software.com/products/dcv

[29] Real VNC, http://www.realvnc.com/

[30] NICE EnginFrame, http://www.nice-software.com/products/enginframe