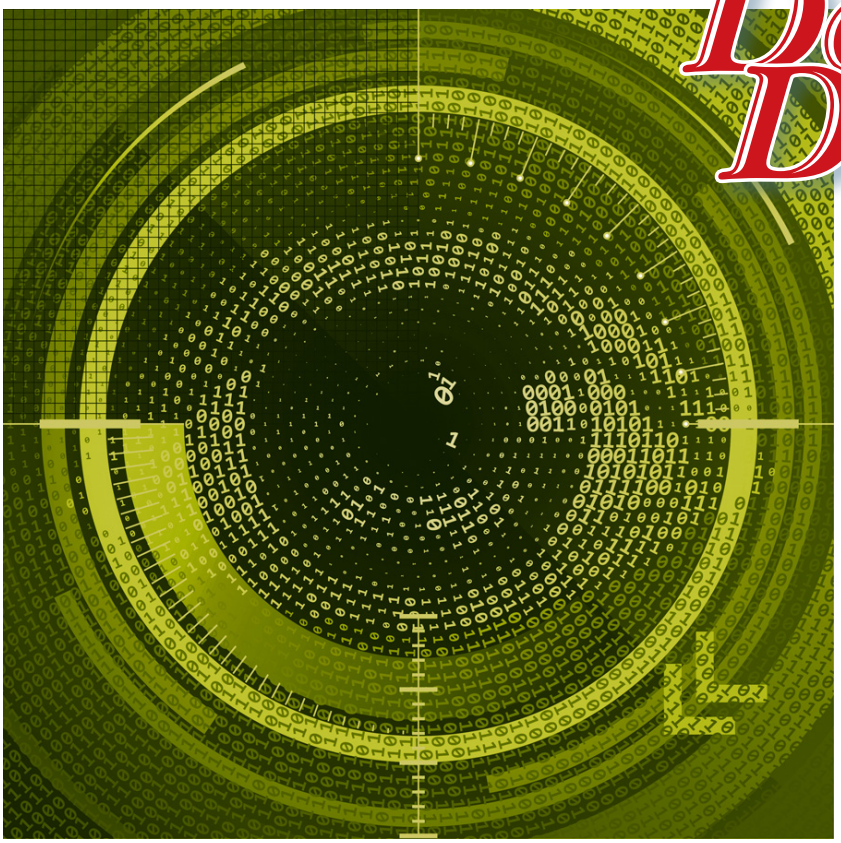
InfoWorld

SPECIAL REPORT

Virtual Desktop Infrastructure

Deep Dive



The new face of thin client computing

Making the leap to Virtual Desktop Infrastructure

When the server maintains a virtual machine for every desktop user, a more PC-like experience results, but weigh your choices carefully before you start

By Paul Venezia

THE UTOPIAN VISION of a fluid and easily managed desktop infrastructure has floated around IT since the dawn of the desktop era. There have been many attempts at corralling the desktop hydra, but none have provided a universal solution. Microsoft Terminal Services, Citrix XenApp, and a host of others have found markets, but the quest for desktop salvation continues.

Enter the newest comer: VDI (Virtual Desktop Infrastructure).

At its simplest, VDI is nothing more than one desktop VM per user, running on top of a hypervisor. As with server virtualization, each desktop VM is assigned RAM, disk, and I/O resources, and a full installation of the OS resides on the virtual disk. The user interacts with the desktop VM using a remote display protocol such as Microsoft's RDP (Remote Desktop Protocol), or Citrix's ICA (Independent Computing Architecture). The client is generally a diskless, power-sipping, thin client system that does little or nothing more than connect to the VDI infrastructure.

The result of this and other server-based desktop computing solutions is that the core of the functionality -- and all the precious company data -- is contained within the datacenter, not spread far and wide among cubicle farms or across many miles to remote sites.

By centralizing desktops, you simplify administration and security and remove the need for basic desktop maintenance such as replacing failing power supplies, hard drives, and so forth. Power consumption is also reduced, and in some cases, cooling costs for dense office spaces drop due to the removal of all those fat client systems and their 350-watt power supplies. The upside can be substantial .

DEVILS IN THE VDI DETAILS

But there are definite downsides to VDI. Some of these issues are found in other server-based desktop computing solutions, but affect VDI implementations as well.

Let's start with the most important issue of all: user acceptance and overall performance. Each VDI instance may be fast and snappy while performing relatively mundane tasks such as word processing, e-mailing, or running spreadsheet formulas, but can suffer mightily when confronted with rich content such as Flash applications, videos, or other multimedia applications. This is generally due to the desktop display transport protocol rather than the performance of the VM, but that only makes the problem harder to solve. Very public problems may arise with user acceptance, which is generally the death knell of any large project.

The solutions to this problem can be costly. Some vendors combine efforts on the server and client side and ship audio and video streams alongside the display protocol, matching them up at the client end and using the client's processing power to render the video. This results in much cleaner playback, but requires a more powerful and expensive thin client to handle the extended workload.

Such solutions work well for standard video playback, but still have issues with Flash video and Flash applications. It's easy to test the performance of VDI or any other server-based desktop computing solution that uses the standard RDP protocol: Just use the Microsoft Remote Desktop Connection client to connect to a server or desktop system and try to view a video on YouTube. It may play reasonably well if you're using a 100Mb or faster LAN, but it's simply a nonstarter over anything less than that. Generally speaking, watching a video across an RDP connection requires 3.5MBps sustained bandwidth.

Bandwidth concerns may dog any form of server-side desktop computing, such as serving desktop sessions across WAN links (a classic scenario for high latency and/or low bandwidth). Printing and proper mapping of USB devices may also be problematic. These issues can be sorted out with the right combination of tools and budget, and aren't exclusive to VDI, but they need to be addressed during VDI planning phases.

Other issues are clearly VDI-specific. The first and most significant: storage requirements. Some VDI implementations require that every desktop system have a virtual disk like any other VM. When you factor in 8GB or 10GB per desktop VM, multiplied by the expected VDI user count, storage quickly becomes an expensive issue. Also, it should be noted that VDI does very little to reduce desktop software administration requirements, because each VM is an island that must be managed like any other desktop system. That means using third-party tools to push updates, install software, and make changes.

And then there's cost. On one hand, VDI may leverage an existing virtualization infrastructure, possibly lowering the initial hardware cost below other server-side desktop computing solutions. But once you've broken through the layers of software licensing required for VDI, you may find that it eventually evens out. Depending on the solution you choose, you could wind up paying more per desktop VM than per traditional fat client system, with the ROI pushed pretty far out. It's vitally important to run all of these numbers before you jump into any VDI implementation. Factor in thin client costs, implementation costs, licensing, and expansion of an existing virtualization infrastructure.

VDI vs. TERMINAL SERVICES

Distinct parallels can be drawn between VDI and traditional Terminal Services solutions. They both place many user desktop sessions on a single server or set of servers and they both use the same protocols to deliver those sessions to the same thin clients. Both offer centralized desktop management tools. But the similarities end there.

A Terminal Services environment places all user sessions on the server OS itself. This means that a single instance of Windows Server 2003 or Windows Server 2008 is installed on a bare-metal server, and all users log into that server instance. Each user session is presented with its own desktop, but runs alongside all the other sessions on that particular server.

The benefit of this scenario is one of scale: Generally speaking, you can probably shoehorn more Terminal Services sessions into a single physical server than VDI sessions, although that depends on the application set. On the flip side, in a VDI environment, it's possible to institute hard and fast resource limits to any user session simply by defining the resources of the desktop VM.

In a VDI implementation, that same server is running a hypervisor, not a full OS, and is hosting some number of desktop VMs. The end effect is that each method places multiple desktop sessions on the same server, but the manageability of those sessions differs significantly.

For instance, with Terminal Services, there's no way to snapshot a session like you can a VM, nor is there a way to migrate active sessions from one physical server to another. This means all users must log off of a Terminal Services server before it can be taken down for maintenance. With VDI, all of the active desktop sessions on that server could be migrated to other servers in the farm without disruption. The server that normally hosts all those VMs could be taken down for maintenance without anyone knowing.

Indeed, as with any advanced virtualization infrastructure, it's possible to completely rebuild the entire underlying physical server structures without taking a single application offline or interrupting a single user.

Further, load-balancing is intrinsic to a properly implemented VDI solution. If one or more desktop VMs are using significant resources on one host system, other VMs on that system may be migrated on the fly to other physical hosts, ensuring that all the desktops have enough resources to go around. In traditional Terminal Services environments this is not possible; a single heavy user session can negatively impact other sessions on the same server without any automated remedy.

But Terminal Services has one big management advantage: Changes made to the host servers are made available to every user session. Application installation is simplified to a degree and global modifications are relatively easy.

With VDI, that's not always the case. Depending on how the desktop VMs are deployed, updates and changes may require manually modifying each VM

individually; using third-party management tools; or firing up a desktop VM template, making the changes, and saving the template. Desktop VMs that are linked to the master template will reflect the changes the next time the user logs on.

Nonetheless, application handling is one of the main benefits of VDI over Terminal Services. Many applications simply refuse to function in Terminal Services environments. Others have limited functionality or may not be supported by their vendor in a Terminal Services environment. This presents a real problem for any Terminal Services-based infrastructure, but it's generally not a problem for VDI.

Why? VDI implements a single dedicated desktop instance per user, indistinguishable from a physical desktop at the OS level. Any application that would run on a fat client should run on a VDI VM (with some exceptions made for video and graphics performance). This is a major reason some shops are pushing VDI over Terminal Services.

PLANNING AND IMPLEMENTATION

Your approach to implementing a VDI solution depends on the individual IT infrastructure, the expected user requirements, and the VDI solution itself.

If all users will be connecting to their sessions via high-speed LAN, for example, then the choice of protocol may not be terribly important. But if there will be remote-site users, or if users will be connecting to their VDI sessions from home or while on the road, the choice of protocol is crucial. In these cases, you will obtain better performance and usability with Citrix's ICA protocol than with RDP or Teradici's PCoIP (though PCoIP, adopted by VMware, is still better than RDP). Yes, RDP will function over these links, but the user experience will be worse than with ICA or PCoIP when bandwidth is low or latency is high.

Beyond the protocol is the layout of the VDI deployment itself. VDI can put a big dent in a storage array if every user session requires its own virtual disk. Luckily, there are solutions to that problem. The best way to reduce VM storage requirements is to use VM templates and generate new desktop instances on the fly as users log in.

A VM template is built from a full desktop OS installation that is prepared with all requisite applications, settings, tweaks, and so forth. Users' VMs can be built either from complete copies of the template's virtual disks or from deltas based on the original template. Using the latter approach is generally best. In either case, when a new user session is connected, a snapshot is taken of that template, and the user logs on to the copy as a desktop instance. When the user logs off, that instance is destroyed; the virtual disk lives only as long as the user is logged in.

Generating users' VMs from templates requires some snappy disk during periods of high log-in activity, such as the morning or during shift changes. To stay ahead of peak demand, templates can also be used to generate a pool of available desktops before users log in.

Naturally, template-driven deployment does not permit the use of static desktops on a general basis. Each desktop instance is present only during the user session and maintains no state otherwise. This requires that the rest of the infrastructure be built to support transient desktops. Functions like roaming profiles, enforced network file storage, and the abundance of group policies that make these things happen must be in place in order for template-driven deployment to be possible.

There are also methods to contain specific user groups within specific VM templates. For instance, a gold template could be created for engineering users, accounting users, call center users, and so forth, with each of these user groups assigned the proper VM instance based on its needs. This allows for granular control over resources and applications; engineering users, for example, will require more RAM and CPU resources than call center users.

From the user's standpoint, of course, the use of VM templates means working in a standardized, locked-down environment. Unless the software the user needs or wants is provided by IT, it won't be available. From IT's standpoint, of course, that's not a bad thing.

There's no cut-and-dried calculation for the physical resources your desktop VMs will require. For a single-app VDI VM, it's best not to descend below the single-CPU, 384MB RAM threshold, and a rule of thumb might be closer to one CPU and 512MB RAM. For general-purpose VMs, trial and error will dictate the right mix of CPU and RAM resources to find the sweet spot between adequate performance and physical server capacity.

In most cases, you'll find that desktop VMs will chew through RAM more than CPU resources, so it's best to use a virtualization platform that can both oversubscribe RAM counts and leverage shared memory segments to get the biggest bang for your VDI buck. Whereas each VM may be granted 512MB, in most instances the actual physical RAM footprint should be much smaller when running on a central hypervisor.

USER ACCEPTANCE AND USE CASES

Sometimes the decision to adopt server-based desktop computing is a no-brainer. Call centers, treatment rooms, point-of-sale, and other fixed-purpose computing environments lend themselves to the server-based desktop computing model.

Limited user interaction is the main reason these environments suit Terminal Services, and it's a good rule of thumb for VDI too. Scenarios in which employees tend to use only one or two applications, and desktop workspaces are interchangeable fit nicely with the concept of a non-persistent VDI implementation.

The typical office user, however, will find the switch more problematic. If users typically customize their desktop environments, install applications, stream music, or listen to CDs on their PCs, they will find the restrictions infuriating. Business functions that require the use of media such as CDs may also be problematic.

With careful planning, you can reap the benefits of non-persistent desktop images and still allow users to enjoy some desktop customization and personalization, but that might not be possible in all cases. Above all, it's important to spend sufficient time and resources on a pilot program that involves at least one representative in every user group you expect to bring into the VDI fold. Otherwise, expect problems.

So it is with every user-facing IT project, but with VDI, it's vitally important to know your users, their requirements, and their expectations. A successful VDI rollout hinges not only on technology, but also on politics.

CLIENT CONSIDERATIONS

The selection of a hardware client for your VDI deployment directly relates to the VDI solution chosen. Some thin client devices won't work with some VDI solutions; others specifically work with a certain protocol only. The most popular protocols, namely RDP and ICA, have broad support across nearly every platform and device, from Windows to Linux and Macs to smartphones.

If you're planning a basic VDI call-center-style implementation, look no further than the lower-end clients. In fact, as long as the client has the requisite client connection software and you're just doing virtual desktops, you probably don't need anything more. If you're expecting to leverage some local applications on the client rather than simply connecting to a VDI back end, then you'll need more robust thin clients with beefier CPU, RAM, and graphics capabilities. However, be aware that if you go with Windows-based thin clients, you're paying not only for the VDI license and the Windows desktop instance license, but also for the Windows thin client license. In essence, you're paying three times for the same user session.

Most thin clients support the popular VDI solutions, so client compatibility isn't generally an issue, but client management can be. Even if the clients lack a spinning disk and a full OS, they still need occasional firmware updates and other management tweaks. Be sure to inspect the vendor's management offerings prior to purchase.

It's also feasible to convert aging full desktop systems to VDI clients. For instance, the open source project Thinstation offers a variety of stateless thin client packages for fat clients. These can be installed on a local hard drive or booted directly from the network. Both methods are feasible, but the latter is far less problematic, because it enables admins to remove the local hard drive, reducing the chances of failure. Leveraging existing hardware is certainly cost effective from the point of view of reducing up-front costs; it just doesn't do much to reduce power consumption at the desktop level.

Zero clients are also available, which are as thin as thin gets, with the CPU integrated right into an LCD display. These do nothing more than plug in to the server-side desktop computing infrastructure, but they sip power and require very little maintenance.

BEYOND THE WALLS: REMOTE VDI

As with Terminal Services deployments, VDI can be used as a VPN of sorts with the use of secured external access. Generally, users run a VDI client on their home PC or company-issued laptop, which connects back to the corporate VDI farm in a full-screen session.

Although the underlying system isn't a thin client, the connection to the VDI farm is identical. This allows these users to function exactly as if they were at their desks, and doesn't require that internal data be transmitted to the remote system. Employees can use any computer anywhere to access internal resources securely, as long as they log off properly.

There are other options for remote VDI use, such as the ability for laptop users to check out their VDI desktop, and then check it back in when they're done. In essence, users retrieve their VMs from the server and run those images on their local systems. When they return to the office, that image is then uploaded back to the server. This may not have broad appeal yet, but it's an attractive feature for a mobile or telecommuting workforce.

BRINGING IT ALL TOGETHER

Although it's obvious that VDI can't solve every desktop problem, neither can Terminal Services or fat clients. Mixing the various technologies can have real benefit.

Terminal Services, VDI, and application streaming can be combined into a three-pronged approach to solving the desktop puzzle. This mix allows IT groups to leverage Terminal Services for the applications that it supports and to use VDI instances for the applications that will not function in Terminal Services environments -- with application streaming for yet other apps. All of this can be presented within the same virtual desktop session. It's expensive and complex, but it offers tremendous flexibility in application delivery and manageability.

VDI is a solid solution for providing server-side desktop computing, but it's not for every environment. The scenarios in which VDI works best are generally the same in which Terminal Services functions best, with the added benefits of greater application compatibility and higher fidelity to the traditional desktop. Remember to pay close attention to the licensing costs and be sure to run a thorough pilot before making the final leap. It's a big project, but if done right, it can reap big dividends.

Paul Venezia is an InfoWorld *senior contributing editor and a veteran consultant.*