

Saving money and increasing security of desktops: forget “VDI”

The problem with desktop “virtualization”	2
What is “client virtualization?”	3
Why client virtualization instead of VDI?	4
Where does client virtualization make sense?	5
What client virtualization is not	7
The client virtualization user experience	7
Costs and efforts for getting started with client virtualization	8
The bottom line	8
About this paper’s sponsor: Neocleus	8

Written by Brian Madden
July 2009



In this paper, independent industry expert Brian Madden will look at how companies are trying to deal with the management of all the various desktop and laptop devices that are out there. This is something that's causing a lot of people to think about "VDI." Brian argues that while VDI makes sense in certain use cases, it's never going to be a broadly-adopted for a majority of desktop users.

Instead, Brian believes that client hypervisors will play a bigger role than VDI in the world of desktop virtualization. Read on to learn why.

The problem with desktop "virtualization"

At its most broad (and literal) definition, the "virtualization" of a desktop happens whenever that desktop's logical management has been separated from the physical device. In today's world, many people think "VDI" when they think of "desktop virtualization."

While it's true that VDI is *one type* of desktop virtualization, it's not the *only type* of desktop virtualization. (For those who aren't clear on the terms, "VDI" literally means "Virtual Desktop Infrastructure." But in common use, "VDI" means the "flavor of desktop virtualization where users' desktop computers are replaced with thin client devices, and then they use the thin clients to connect to Windows desktops running as VMs on a server in a datacenter.")

In other words, VDI is server-based computing, just like Citrix or Terminal Server. Really the only difference between VDI and Terminal Server is that VDI has a one-to-one relationship between users and remote hosts, whereas Terminal Server has a many-to-one relationship between users and remote servers.

Server-based computing has several advantages over traditional desktop computing, namely:

- *Management*, since all instances of the OS and applications are in the datacenter, updates and patching are simple.
- *Access*, because users can access their applications from anywhere and from any type of client device.
- *Performance* improvements, because multi-tier "fat" apps run in the datacenter, close to their data sources, and only small screen updates go across the network to the client
- *Security*, since all data stays in the datacenter, there's nothing of value to steal on the client device.

Again, these four advantages are characteristics of server-based computing, and thus they apply to both Terminal Server-based and VDI-based flavors.

Of course any model of computing with great advantages also has disadvantages, and server-based computing is no different. Specific disadvantages include:

- *Online only*, because if the client device is not online, then there's no access to the remote host, and all work stops.

- *Network performance*, since all execution happens in the datacenter, if the network is too choppy or too slow then server-based computing won't work
- *Limited peripheral support* means that certain applications and use cases won't work. (For instance, even if you have USB redirection, and even if your remote protocol and network will support multimedia applications, you still won't be able to load video from a camera connected via USB to a client, since all of that raw video would need to travel across the network from the client to the host.
- *Significant datacenter hardware costs* are usually associated with server-based computing implementations, because you have to buy the servers and storage to run all the user sessions that were previously run locally on users' client devices.
- *A major paradigm shift* can be troubling for users and admins, and server-based computing certainly is a major shift.

The advantages and disadvantages of server-based computing listed here have been known for fifteen years. And the fact that some very real disadvantages dampen the advantages is exactly why server-based computing has remained (and will remain) a niche solution that's only appropriate for a small subset of users or use cases.

Of course the disadvantages of server-based computing have not prevented people from using it in cases where it was not appropriate. A lot of people have been burned by server-based computing because they rolled it out where it wasn't the right fit, but they did so because they had to do something to deal with the crazy management costs and security holes of their existing solution, and server-based computing was the "least worst" choice.

Such is the cost of a world with only two choices: server-based computing or the "old way."

Fortunately we don't live in that "all or nothing" world anymore. We now have access to new technologies that balance the best of both worlds, controlling the costs and security of traditional computing models while still letting our users run graphically-intense software with the ability to work offline.

This can be achieved with "client virtualization."

What is "client virtualization?"

We opened this paper with a discussion of "desktop virtualization" and how it's a very broad term that's larger than Terminal Server, VDI, or server-based computing. We could also say "client virtualization" falls under the larger "desktop virtualization" umbrella.

(While it's somewhat unnatural to define "client virtualization" and "desktop virtualization" as two different things, there are only so many words we can use, so it's inevitable that we'll have to play this game of semantics at some level.)

So what is "client virtualization?" We can define it as running a virtual machine directly on a client device (desktop or laptop). There are two ways this can happen, which are

commonly referred to as “Type 1” and “Type 2” solutions. (Seriously, those two terms are all we could come up with!)

Type 1 client virtualization

In a Type 1 client virtualization environment, the virtualization engine is the main operating system on the client device. In many cases it’s literally installed onto a blank laptop by dropping a disc into the drive and wiping out the machine. This is commonly called a “client hypervisor” since it runs in a similar way to hypervisors like VMware ESX or Microsoft Hyper-V.

When one of these client hypervisors is used, every OS is executed as a “guest,” and the hypervisor has ultimate control over the hardware.

Type 2 client virtualization

The other kind of client virtualization software is commonly referred to as a “Type 2” environment. In a Type 2 environment, an OS (like Windows or Mac OS X) is installed like normal on the client. Then a virtualization application is installed just like a normal application into the OS. This creates VMs that run “on top” of the existing OS.

(By the way, people often refer to Type 2 environments as “Type 2 hypervisors,” although that’s not technically correct since, by definition, a hypervisor is running at the lowest level as in the Type 1 scenario. These “Type 2” things should be called “Type 2 virtualization environments” or “Type 2 platforms” or something. They should definitely NOT be called “hypervisors.”)

Choosing a Type 1 or Type 2 environment

In many ways, Type 1 and Type 2 environments are similar: they both all allow complete virtual machines run locally on a client device that are totally separate from a locally-installed OS. And that’s the key here. With client virtualization, an administrator can completely build, configure, secure, and deploy a complete “corporate” VM that runs locally on an end user’s client device.

In case you’re wondering about the practical differences between Type 1 and Type 2 client virtualization environments, think of it like this: Since Type 1 environments replace the local OS, they’re most often used when the corporation owns the client asset or when the virtual machine guest is the primary OS the user uses. Type 2 environments, conversely, make the most sense in situations where the user only occasionally runs applications from their VM, or perhaps where a user only needs temporary access to a guest VM.

Why client virtualization instead of VDI?

Now that we’ve looked at two different types of “desktop virtualization”—server-based computing (both VDI and Terminal Services) and client virtualization (both Type 1 and Type 2)—let’s look at why you’d choose one over the other.

We already discussed the fact that a lot of people are choosing server-based computing today to get their management and security costs in control, although in a many cases

this is done at the expense of user experience or it requires significant investments in the datacenter.

Client virtualization is interesting because it allows companies to enjoy the same central management and high security of server-based computing / VDI, while also giving their users the ability to work offline and the ability to use any application, while leveraging the existing client devices that are already out in the field.

That said, it's not truly client virtualization "versus" server-based computing, because each type of virtualization solves different challenges. Sure, there may be some people who implemented server-based computing when client virtualization would have been more appropriate, and those people should switch. But in most cases, if a particular application requires server-based computing (based on the advantages listed previously), then you should continue to use server-based computing, even if your client desktops have been virtualized.

Where does client virtualization make sense?

When comparing client virtualization to the traditional way of managing clients (where you install an OS natively on the device), there are several scenarios where client virtualization really shines. Specifically:

- You can leverage existing client hardware that's already "out there"
- The client virtualization engine can completely secure the client device
- You can use a single disk image for all your devices
- Users can run multiple client environments side-by-side
- OS migrations are easy since you can bring the new environment online along side the old environment
- Desktop restoration or recovery becomes almost "automatic"

Let's step through each of these advantages one-by-one:

Works with existing stuff that's "out there"

The single biggest advantage of client virtualization is that you can leverage all of the desktops and laptops you already have out there. This applies to the client devices themselves *and* to the back-end datacenter. Server-based computing projects (whether Terminal Server or VDI) move the entire client environment into the datacenter, which means implementing those technologies has a huge upfront cost as you build out your datacenter to support all your users. With client virtualization, you only need a few servers on the backend to send out disk images. Everything else runs on the client devices you already have.

Security

Because client virtualization moves everything into a VM, it's now easy to encrypt the entire VM disk image. And since this encryption is done outside the VM, it's not even possible to boot the VM unless the proper password for decryption is known.

Client virtualization solutions also include a “kill pill “or” poison pill capability, which means that laptops can be configured so that they need to “check in” every so often. If they ever fail to do so, they automatically “destruct,” wiping out and thrashing the disk images. Or if a user knows that his or her laptop was stolen, an administrator can immediately invalidate it, meaning that if that laptop ever does show up on a network, it’s immediately wiped out. This security checking and secure wiping is performed by a background service VM, so there’s no way that a user can cancel the process in the task manager or anything.

Single image for all client devices

Client virtualization means that administrators only have to create a single disk image for all their client devices, regardless of the make or model. In fact that same disk image can be used locally on client devices and centrally for VDI machines.

Business and personal environments on the same client device

Client virtualization is all about running all client environments as VMs on the client device. This means that it’s relatively easy to run multiple VMs side-by-side on the client device. This is a great solution to help users maintain separate work environments and personal environments on the same laptop—something that’s important now because it’s just not practical to tell users not to install any non-work software or data on their laptops.

That said, we still need to protect our corporate assets—both the client devices and the corporate data. With a client virtualization environment, we can give the user two VMs—one VM that is totally locked-down and controlled that runs the corporate apps, and another wide-open VM for personal use that corporate IT doesn’t have to support.

This ensures that the user can’t do anything in the personal VM which would break or jeopardize the integrity of the corporate VM. We don’t have to worry about viruses or sniffers or anything like that since the personal environment is completely separate from the corporate environment.

In fact we could even use technologies like Network Access Protection to put the two VMs into two different VLANs. A corporate VLAN would only be accessible from a domain-joined VM with the proper security certificates. Everyone else, including the personal VMs, would only have access to a VLAN with Internet access and nothing else.

Side-by-side corporate environments

In addition to running corporate and personal VMs side-by-side, there are many scenarios where we’d want to run multiple corporate VMs side-by-side. For example, if there’s a merger or acquisition, the new company would only need to provide a new corporate disk image to the users. The users could then run the new environment and the old environment at the same time.

OS migration

The same could be said for OS migrations. The corporation could provide new Windows 7 virtual machines for users to start testing while the users still accessed their Windows

XP VMs. Then down the road the company could just disable the old XP images and all users would have to start using the new Windows 7 image.

Desktop restoration or recovery

Since the entire client environment is in a VM, backup is easy. It can be done totally out-of-band, and simple delta files are all that need to move across the network. Restoration is an easy process too, since the virtualization environment insulates the VM from the physical hardware, recovery can be done from any device to any device. A user with a lost laptop or hardware failure can be up and running again on temporary equipment in less than an hour.

What client virtualization is not

While client virtualization has a lot of advantages, it's important to keep it in the proper context when thinking about it. Client virtualization is not a replacement for application virtualization. It does not replace desktop management. (It helps desktop management, but it doesn't replace it.) Client virtualization is not a replacement for backup tools. (Again, it helps the backup process, but it doesn't make it go away.)

Fundamentally all client virtualization does is bring the benefits and advantages of server virtualization to the client. But like server virtualization, each company needs to figure out the best way to leverage it in their own environment.

The client virtualization user experience

Remember that all client virtualization solutions can be broken down into either "Type 1" or "Type 2" products.

Type 2 solutions can be somewhat confusing for users, since they would have "two desktops." (Desktop 1 is the traditionally-installed OS, and Desktop 2 is the guest OS running in a VM on top of the host.)

Type 1 solutions, on the other hand, are completely transparent to the end user. (Well, if they're done right, the presence of the hypervisor is completely transparent to the end user.) In fact using a Type 1 solution is a lot more meaningful than reading a paper explaining it. (Although that said, Type 1 client virtualization solutions can be tricky to demo, because when they're done right, the user doesn't know it. So you boot up a laptop and show the user, and if their reaction is, "yeah? So what?" then you know it's working!)

And that's all there is to it. If you have a Type 1 client hypervisor, the user experience is identical to a locally-installed OS. The laptop boots up and runs Windows, and the user wouldn't even realize it's running via a VM.

The same is true if the client is running multiple VMs side-by-side, except that there would be a special key combination to switch from one VM to the other. But other than that, the experience is transparent to the user.

Costs and efforts for getting started with client virtualization

Performing a full cost analysis or ROI study is beyond the scope of this paper, but we can touch on a few basic things here:

First, the fact that client virtualization can leverage your existing client hardware is key. That means that the initial costs are limited to the cost of the client virtualization software and the time it takes to get it installed.

Then once it's in place, management costs should start to melt away (although the exact amount you'll save will depend on how you managed the old environment and how you're planning to manage the new environment.)

It's also possible that you'll be able to save some costs by not using other software programs. For example, if you currently use (or plan to buy) client security software that encrypts files on the client device, you don't have to use that anymore. The same can be said for backup and recovery software.

The bottom line

Client virtualization is not an ultimate solution that fixes every desktop problem. It's just one piece of the technology puzzle we're assembling to manage desktops and laptops moving forward. The good news is that the technology is real today, and you can put your hands on it and start testing it now. And since client virtualization doesn't fundamentally change the way users do their work, it's easy to implement and test as you get more comfortable with the technology.

About this paper's sponsor: Neocleus

Neocleus believes that the only way to truly meet the demands of today's organizations is to provide solutions that give IT the control they require while also delivering an uncompromised, high performing end user productivity environment. To this end, Neocleus transforms the management and security of client computing with its innovative client virtualization solutions. Using Neocleus, a single PC functions as if it were two completely separate machines. The isolation of 100% secure operating environments provides a platform to deliver solutions that are flexible, high-performing, cost-effective, and integrated with existing systems and processes.

The core of Neocleus' groundbreaking client virtualization technology is based on the Xen hypervisor virtualization technology. Neocleus has extended this technology to the personal computing environment to deliver a Type 1 hypervisor that runs directly on the bare metal of the client hardware. The Neocleus hypervisor is optimized and designed specifically for personal computer use and enables solutions to efficiently operate in 100% secure, isolated, independent environments or virtual machines (VMs). Technology called Neocleus PassThrough allows each VM to interact directly with underlying hardware devices delivering native performance and an uncompromised user experience. A robust virtualization application programming framework offers powerful programming capabilities, accelerating the creation of client computing applications mentioned in this paper.

Neocleus is currently shipping NeoSphere, the first product built upon Neocleus' second generation Type 1 client hypervisor. This platform reduces the cost of desktop ownership and provides IT with new levels of flexibility, efficiency, and security.

For more information about Neocleus visit www.neocleus.com.