

# How bare-metal client hypervisors will mean the end of agent-based Windows management

Abstract..... 2

Understanding the “old” (current?) way of managing desktops..... 2

How do companies manage desktops currently?..... 4

A solution? A “bare-metal” client hypervisor ..... 4

What is a bare-metal client hypervisor?..... 5

Isn’t a bare-metal client hypervisor just “offline VDI”? ..... 5

The larger trend of “desktop virtualization” ..... 6

Why use a bare-metal client hypervisor for PC lifecycle management?.. 6

So where does that leave us? ..... 7

How to choose a bare-metal client hypervisor..... 8

The bottom line ..... 9

About this paper’s sponsor: Virtual Computer ..... 9

Written by Brian Madden  
July 2009



## Abstract

---

With all the recent focus on VDI and desktop virtualization, people are starting to talk about “bare metal” or “Type 1” client hypervisors. Promising the ability to run virtual desktop workloads “locally” on client devices, these hypervisors have the potential ability to radically impact our industry.

But client-hypervisors are not just about delivering virtual desktops to local client devices. In fact, one could argue that the true value of a client hypervisor is in the desktop lifecycle management arena, not in the desktop virtualization space.

In this paper, Brian Madden explores the value of bare-metal client hypervisors in the context of PC lifecycle management.

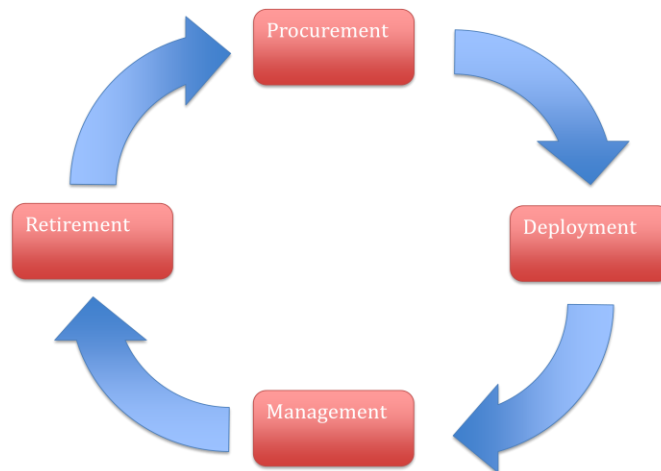
## Understanding the “old” (current?) way of managing desktops.

---

Before we dive into the details of bare metal client hypervisors, we should take a look at what life is like without them. And in fact, let’s take a quick look at the general state of the desktop management industry.

Even though server virtualization is exploding and drastically altering the design of data-centers, at the time of this writing in 2009, desktop virtualization has not really taken hold. Sure, some people are using it here and there in various forms (Citrix server-based computing, for example), but most of these use cases are for specific applications that are deployed in addition to the “traditionally”-managed (or unmanaged) desktops.

In fact, desktop management has not really changed at all over the last fifteen years or so. The “lifecycle management” (LCM) process charts that were popularized in the 1990s still apply today:



(This diagram was created by the author, although it uses the same words that Gartner uses for their LCM diagrams. There’s probably an infinite variation of diagrams like this

available on the web—some with five or six boxes instead of four—but all cover the same basic idea.)

Each box represents a major phase in the circle of life for a desktop. And each of these boxes can be broken down into several sub-elements. (Again, this is not meant to be an exhaustive list, but to instead demonstrate the kinds of elements that make up each phase.) For example:

### Procurement

- Define requirements
- Vendor selection
- RFP process
- Negotiation
- Purchase

### Deployment

- Hardware configuration
- Building the OS image
- Install the OS
- Install the drivers
- Configure the OS
- Install apps for the user
- Configure apps
- Migrate the user's old data
- Physically deliver and setup the device

### Management

- User support
- Training
- Asset tracking and management
- Configuration management
- Break / fix
- Software patching and updating
- New app installation
- Securing the whole thing
- Ensuring the right stuff is backed up

### Retirement

- Remove device from user
- Reclaim any licenses on it
- Change device status in asset tracking database
- Prepare physical device for disposal (data wipe, environmental checks, etc.)

Regardless of the size of the company, and regardless of how advanced or basic a company is, some form of each bullet item *\*must\** take place for each of the 600m or so corporate end-user computers in the world.

## How do companies manage desktops currently?

---

Clearly there are a lot of steps involved in managing the lifecycle of desktop devices within companies today. And any company with more than a few desktops has undoubtedly looked for ways save some costs and effort based on economies of scale arising from having more than one desktop to manage.

The management-saving techniques used here are as varied as the people who use them. For example, some desktop admins may simply use disk “imaging” as a way to deploy new devices, thus saving them the time needed to manually install a lot of similar machines.

On the other end of the spectrum, some software companies make billions of dollars *every year* selling desktop and PC lifecycle management software. Microsoft SMS (now called System Center Configuration Manager) has been offering centralized asset management, user support, and patching capabilities for well over a decade. Altiris (now owned by Symantec) built a multi-billion dollar company writing software to manage these “endpoint” devices at all stages of their lifecycle. (And this continues moving forward. Symantec just announced “Version 7” of their Altiris Client Management Suite in March 2009.)

So while the whole world has a high degree of comfort with these “traditional” desktop management tools, there are a lot of challenges in the PC lifecycle management process that these tools don’t address (or in some cases that they themselves actually *introduce*).

Most of these challenges stem from the fact that Windows was designed to be filled with drivers and to be “locked” to a specific device. This poses a challenge because administrators spend their time maintaining devices instead of users. In essence, “device management” and “Windows management” become one.

As the hardware vendors make changes to their devices, admins have to recreate the images they deploy to those devices.

## A solution? A “bare-metal” client hypervisor

---

There’s a relatively easy way to solve many of the lifecycle management challenges that exist in today’s management products. If you put a hypervisor directly on the end user client device (e.g. laptop or desktop), then you effectively “decouple” the management of the device from the management of Windows and the user.

## What is a bare-metal client hypervisor?

---

Hardware virtualization engines come in two types, called “Type 1” and “Type 2.” (Seriously, those awesome names are really what they’re called!)

A “Type 1” hypervisor is the name given to hypervisors that run natively on a piece of hardware. In other words, the hypervisor is the operating system. This is how all the major server hypervisors (VMware ESX, Xen, Hyper-V) work nowadays. Type 1 hypervisors are typically called “bare metal” hypervisors since they install and run on the “bare metal” of the computer.

A “Type 2” virtualization engine is actually more like an application that you install and run on top of an existing operating system, like Windows. VMware Workstation and the free VMware Server are probably the most widely-known Type 2 virtualization environments, followed closely by Microsoft Virtual PC or VMware Fusion for the Mac. There’s also an open source project from Sun called VirtualBox that’s gaining popularity.

Type 2 virtualization engines that run on laptops and desktops have been around for a long time. While they’re good for certain scenarios (including lab testing and offline VDI), Type 2 environments don’t really help the PC LCM problem since they run on top of another OS. (Which you still have to manage.) So a Type 2 virtualization engine on the client doesn’t “save” you anything; in fact, it probably creates more work for you in that it’s essentially just one more application to manage on top of everything else.

Contrast that to a Type 1 bare-metal hypervisor on the client, which acts as the lowest level operating system. Since it’s not running on top of another OS, Type 1 client hypervisors can be very efficient and don’t require an underlying OS which itself must be patched, secured, etc.

## Isn’t a bare-metal client hypervisor just “offline VDI”?

---

When thinking about bare-metal client hypervisors, a lot of people think these are a form of “VDI.” (Or, more specifically, they think client hypervisors are for “offline VDI.”)

In today’s world, the term “VDI” is typically used to describe a server-based computing (SBC) architecture similar to terminal server, except where users leverage a remote display protocol to connect to a back-end that’s made up of single-user desktop OSes running in VMs instead of a single terminal server.

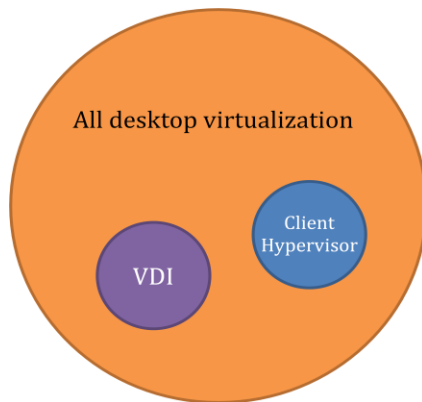
Many vendors offer VDI products, including Citrix, VMware, Quest Software, Symantec, and (later this year) Microsoft.

The latest version of VMware’s VDI product (called “VDI View”) included an experimental capability they call “offline VDI.” This capability combines a Type 2 client virtualization engine, a remote VM-based VDI session, and some synchronization software that lets a user “check out” his remote VM. The VM’s disk image is copied from the remote server to the client machine where it can be booted and used locally, regardless of whether a network connection is present.

Taking that concept one step further, VMware and Citrix have announced plans to build their own bare-metal client hypervisors which will allow them to extend their existing VDI implementations onto bare metal client devices. This had led a lot of people to mentally classify bare-metal client hypervisors as “just” offline VDI. “If I’m not using VDI today,” the thinking goes, “then why should I care about offline VDI / client hypervisors?”

While it’s true that offline VDI does require some form of client virtualization environment, using a client hypervisor for offline VDI and using a client hypervisor for PC lifecycle management are two very different things. (It’s like suggesting cars and trucks are the same thing because they both use internal combustion engines to propel themselves forward.)

## The larger trend of “desktop virtualization”



VDI and offline VDI are both part of the larger industry trend towards “desktop virtualization.”

Taking a step back, consider that “virtualization” is separating the physical from the logical, so “desktop virtualization” is any time you manage the desktop OS separate from the physical client device.

And that’s the key. “Desktop virtualization” has the potential to revolutionize PC lifecycle management, and client hypervisors play a big part of that, whether you use VDI or offline VDI or no VDI at all.

## Why use a bare-metal client hypervisor for PC lifecycle management?

Moving forward, there will be a convergence of “desktop virtualization” and “PC lifecycle management.”

If you look back at the original list of phases that make up the PC lifecycle management process, imagine how many of those disappear in a desktop virtualization environment? So many of the specific sub tasks, like installing the OS, installing drivers, and deploying the OS to the device either disappear or become exceedingly easy.

Everyone reading this can understand the value proposition of being able to create a single disk image that could be used by multiple users. And everyone reading this can understand that use cases exist where you’d want the disk image to run locally on a client device. So how can we combine these two?

Citrix Provisioning Server (Ardence) showed a lot of early promise. With Provisioning Server, you can create a single disk image that’s shared by hundreds or even thousands of users. This works perfectly as long as the client devices that you’re deploying this image to are identical (or at least identical enough to use the same image that’s been “overloaded” with drivers). But what if you have a several different types of client de-

vices? This means that you need to build several different images, and it means that you won't be able to support any random client device.

"No problem," people think, "We'll just throw a hypervisor on the client devices and use Provisioning Server with a VM on the client instead of the bare-metal client." While this is a simple theory, it's not so easy in practice. Existing Type 2 client-based virtualization engines require an underlying base OS. So how do you manage, deploy, maintain, and patch that OS? And if you're taking the time to manage that, then why even bother with the VDI instance? If you think about it, managing a local OS on a client device while also managing a VDI OS is actually the worst of both worlds. Now you're managing two OSes per user instead of one.

This is the exact problem that a bare-metal client hypervisor can solve. In this scenario, there's only one OS to manage. (This is the point at which purists shout, "Hey! The hypervisor is an OS, so you're still managing two OSes." I guess that's technically true, but the bare-metal hypervisor is much easier to manage than a "real" OS, and the general idea is that it would be transparent. Even though a bare-metal hypervisor is technically a piece of software, it can be managed as if it's an extension of hardware. It's the new "HAL," so to speak.)

## So where does that leave us?

---

The client hypervisor turns Windows into your managed resource. It shifts the focus from the desktop (device) to the desktop (OS).

The technology that's used is platform virtualization via a client hypervisor. But in today's world, it's not about reinventing the desktop architecture. (Even though that could happen in the future.) For now it's just about an easier way to deploy and manage all those copies of Windows.

If you start thinking about what a bare-metal client hypervisor can actually *mean* for a client, there are quite a few advantages that are pretty apparent:

- Since the hypervisor emulates generic hardware which is presented to the VM, a single disk image can be used for different (even *very* different) makes, models, and types of client devices.
- Windows updates and patches can be applied out-of-band. (This means the hypervisor and a small management VM can "wake up" and download and apply patches to the Windows VM's disk image.)
- The traditional lifecycle management aspects of Windows (those four boxes from the circle chart a few pages ago) can be managed from the outside of Windows. (Again, this keeps Windows "generic" and "pure" while still allowing centralized management.)
- Type 1 hypervisors can offer better VM performance than Type 2 virtualization engines, since hypervisors are essentially mini OSes designed for virtualization as opposed to a virtualization app running on top of a normal OS that must "play" by normal app rules.
- Eliminating the non-virtualized host operating system (i.e. using a Type 1 hypervisor instead of a Type 2 environment) ensures that malware in the host does not compromise "trusted" virtual desktops.

- Eliminating the host operating system also allows for a true PC lifecycle management model, since a non-virtualized host would need to either be unmanaged or managed through a separate agent-based approach (which kind of defeats the whole purpose).
- A bare-metal client hypervisor can fully isolate virtual machines on the same client device from each other—allowing an unmanaged personal OS to run alongside a secure corporate OS. (This can facilitate a “BYOPC” or “Employee-owned PC” environment.)
- By running the OS on top of a hypervisor layer that “owns” key physical hardware such as hard drive and networking, a client device can be remotely recoverable even in the event of a catastrophic failure of the Windows VM such as a bad patch or a “blue screen of death.” (Because in this case, the management VM would still be alive and could “roll back” the change.)
- OS migrations can be rolled out in a few clicks (even allowing the “old” and the “new” environments to run side-by-side).
- Since all I/O activity would run through the hypervisor, central admin policies can control *all* types of I/O. (USB filtering, network policies, etc.)
- Since all end user and guest VM disk reads and writes are to virtual hard disks, numerous techniques (copy-on-write, snapshots, etc.) provide a more elegant and less user-intrusive approach for user data backup than traditional “in-band” backup agents.
- The hypervisor layer can serve as a “trusted base” where the boot process can be measured, an encrypted file system can be deployed, etc. without the chance for something inside the Windows guest to compromise it.

Clearly, bare-metal client hypervisors are more than another “agent” running in the Windows client.

## How to choose a bare-metal client hypervisor

---

If this sounds good to you, the next step is to start researching and playing with a bare-metal client hypervisor for your environment. Customers ultimately “win” when multiple vendors offer competing products, and fortunately several vendors (large and small) are offering or planning to offer client hypervisor-based solutions.

What’s most interesting about the emerging client hypervisor market is that the different vendors use client hypervisors to solve completely different use cases. So instead of trying to list out all the vendors and what their products do or don’t do, let’s look at a series of questions you can use to evaluate which vendor’s solution would best fit your PC lifecycle management needs.

- Is there a centralized management console that can be used to create and deploy virtual desktops to the client hypervisor?
- Can the management console be deployed easily on a stand-alone basis to easily get started, or is predicated on having deployed a vendor’s server virtualization and/or VDI solution?
- If you do wish to integrate into an existing enterprise management tool set, are APIs for virtual machine / client hypervisor management functions available?

- What is the mechanism for patching and updating virtual desktops centrally? Can they be updated on a one-to-many basis? One at a time? Self-managed by the user?
- Does the solution provide full hardware abstraction or are certain I/O functions passed through directly from the OS to the physical hardware, necessitating hardware-specific drivers in Windows? If hardware-specific drivers are required in the OS, does the hypervisor management tool automate this?

## The bottom line

---

Remember that each vendor making a client hypervisors today has a different reason and angle for their products, so be sure to know the business reasons why you want to use one before you start evaluating your options.

And while VDI and desktop virtualization technologies are still emerging, using client hypervisors to ease the burden of traditional PC lifecycle management is a compelling use case that's real today.

## About this paper's sponsor: Virtual Computer

---

*Virtual Computer, Inc. is redefining PC lifecycle management by making it as easy to manage a thousand PCs as it is to manage one. NxTop™, the company's flagship PC management product, combines a bare-metal client virtualization platform with a powerful central management system to dramatically reduce PC management costs, while improving reliability, security, and the end-user experience. NxTop uses advanced virtualization technology to isolate the main components of a PC: the hardware, operating system, applications, and user data, allowing each to be managed independently. Founded in 2007, Virtual Computer is privately held and headquartered in Westford, MA. For more information visit <http://www.virtualcomputer.com>.*