



A better way to manage Citrix servers: Centralized block-level disk image streaming

Written by Brian Madden

March 2006

Abstract

This paper covers a technology called disk streaming (sometimes referred to as "software streaming," "network boot" or "diskless boot") from a company called Ardenice, and how you can use it in your Citrix environments to give you much better flexibility and simpler server provisioning and management. In a nutshell, Ardenice has technology that lets your Citrix servers boot from centralized disk image files stored on a file server instead of each server having its own drive. This means that you can add new servers and re-provision existing ones simply by pointing them to a new disk file on the network. It also means that you can reboot Citrix servers at anytime to "reset" them to their gold server image.

At first this sounds really scary, but the technology is pretty amazing and works well. The performance is great too.

In this paper, I intend to take a deep look at how exactly this technology works and how you can apply it to your Citrix or Terminal Server server-based computing environment.

The Technology

In the Ardence world, your computer's disk drive is actually a disk image file sitting on a remote server. (In concept, these disk image files are similar to VMware disk image files.) Ardence calls these "vDisks."

To have your computer use this vDisk instead of its own local disk, you change the boot order preference in the BIOS and configure it so that it boots from the network (or PXE boot). When the computer turns on it boots to the network, grabs an IP address from the DHCP server, and then reads some of the extended DHCP flags to find the bootstrap location. The computer then downloads a very small bootstrap which causes it to contact an Ardence server.

The Ardence server recognizes the booting computer via its MAC address and checks a configuration database to figure out which vDisk file that computer should use. The client computer then mounts the vDisk just like a normal disk, and the boot process continues as normal.

Ardence calls this technology "streaming," although personally I'm not sure that's the best name for it. To me, "streaming" suggests that the disk content is copied down to the client device as it's needed. I guess in some ways that's true. But with Ardence, the client computer is actually mounting a disk volume over the network. The client computer does not need to have any hard drive locally, and the drive is not copied or cached locally.

Before we go any further, I think we need to take a deeper look at some of the technology that Ardence is using here.

At the most basic level, Ardence developed a Windows disk drive device driver. Much like Dell or hp has drivers that enable Windows to recognize their RAID controllers, Ardence has a driver that enables Windows to recognize a remote Ardence vDisk being accessed across a network.

The core of this is their custom developed UDP-based disk drive protocol. It's UDP-based because UDP is packet-based and connectionless, which means less overhead than TCP. (The downside to this is that UDP packet delivery is not guaranteed, but in today's switched networks, packet delivery is virtually guaranteed anyway, and Ardence built some custom logic into their protocol directly that re-requests dropped packets as needed.)

In concept, their protocol is kind of like iSCSI, although the Ardence protocol is much more efficient. Why? The Ardence protocol was developed from the start for use over a network. This is very different than iSCSI, since iSCSI takes a protocol that was developed for local access (SCSI) and wraps a TCP wrapper around it. In iSCSI transfers you'll often find that the protocol header is larger than the payload!

Another fundamental key to the Ardence protocol is that can endure network failures and disconnects/reconnects. This capability is built right into the Ardence disk driver and protocol.

What does this mean? In a typical network boot scenario (where Windows is booting from a network disk instead of a local disk), if you disconnect the network cable while Windows is booting, the system will blue screen. In the Ardence world, you can pull the cable during the

boot process and the process just sits there. The instant you plug the cable back in, the boot process continues. In my lab I removed / reinserted the Ethernet cable half a dozen times during a Windows Server 2003 boot process, and the server started up no problem!

To really dig into the cool stuff, we'll need to look at the Ardence vDisk files that are stored on the network. There are several different ways that a vDisk can be used. The method that I've described so far could be called a "private" disk model, where each client computer is 1-to-1 mapped to an Ardence vDisk file. The Ardence disk driver running on the client computer redirects physical disk block-level read and write requests across the network to the vDisk file, and the vDisk file grows and changes as the client computer is used. Again, this is a lot like a VMware VMDK file.

However, there is another major option that Ardence provides with respect to disk files. Instead of each client computer having a 1-to-1 mapping to each of their own "private" disk files, you can have multiple client computers share a single "public" read-only vDisk file (with proper Microsoft OS licensing). In this case, Ardence configures the disk file as "read only," and all client computers get the same image.

Of course doing this requires some additional intelligence, because as you can imagine, Windows would blue screen if it tried to boot to a read-only disk.

The way Ardence handles this is that they transparently redirect disk write requests to another location. Each client computer that's sharing the same read-only vDisk ends up with a "delta" (or "write cache," as Ardence calls it) file that holds everything that's changed in on that disk since the computer booted up. This write cache can be stored in a specially segmented area in the client computer's RAM, on the client computer's hard drive, or as a separate file on a network file server.

The beauty of using these public read-only disk images is that when you reboot a client computer, the cache is cleared and the computer starts fresh. (What if you don't want the computer to be reset to the base image on reboot? This is what the "private" disks are for that we talked about first.)

How does this apply to Citrix?

Imagine for a moment what this could mean for your Citrix servers. Right now a lot of people reboot their servers nightly. This gives them a chance to bounce the IMA service, clear out the print spooler, and generally prepare the server for the next day's work. But with Ardence, your nightly reboot could actually reset the computer back to its "gold" state. Anything that any user screwed up on that server during the business day is reset back to the original state.

Another great use of this technology in the Citrix world is that you can have "dynamic" silos. Imagine a scenario with about 50 Citrix servers divided into several application silos.

Silo name and purpose	Number of servers in that silo
Microsoft Office	25
Accounting Software	5
Inventory Application	3
SAP	15
Graphics Design Apps	3

In this case, what happens if you need more servers for Office? You have two choices:

- Buy more servers
- Try to figure out which of your other silos is overbuilt, and re-provision a server from there

Either way, once you identify the hardware to use, you have to install Windows, install Citrix, install Office, and then add it to the farm and published application list. Or you have to image your server, change the SID, and add it to the published application list. Either way, this is a labor-intensive process.

Now imagine that all 51 of your servers were using Ardence, and that all of the servers in each silo were sharing that silo’s single read-only “public” vDisk. In this case, your Ardence management tool would list the MAC addresses of each server as well as the specific vDisk that the server was accessing.

If you want to move a server from the SAP silo to the Office silo, all you have to do is make one simple change in the Ardence admin tool and then reboot the server. When the new server booted up, it would mount the Office silo vDisk instead of the SAP silo vDisk.

Boom! You’re done. You wouldn’t have to do anything else at all. You can move servers between silos all you want.

Confused? Let’s look a little more in-depth at this process.

Let’s say you have ten servers in your Citrix farm. We’ll name them Citrix01 through Citrix10. Next, let’s assume that this farm has two silos—one for Office and one for SAP. In this case then you would have two public vDisk image files on a network server—one with Office installed and one with SAP installed. (How do you make these vDisk files? More on this in a bit.)

In the Ardence administration tool, you assign your server MAC addresses to Windows server names and the particular vDisk that they would boot from. (This tool automatically logs the clients as they PXE boot, making it easy to find and identify them. You can even change their names or update their MAC addresses right from within the tool.)

This might look like so:

Server	MAC Address	vDisk File
Citrix01	00-0E-9B-DC-08-57	Office
Citrix02	00-0E-9B-DC-08-62	Office
Citrix03	00-0E-9B-DC-08-64	Office
Citrix04	00-0E-9B-DC-08-65	Office
Citrix05	00-0E-9B-DC-08-78	Office
Citrix06	00-0E-9B-DC-08-61	Office
Citrix07	00-0E-9B-DC-08-63	Office
Citrix08	00-0E-9B-DC-08-80	SAP
Citrix09	00-0E-9B-DC-78-99	SAP
Citrix10	00-0E-9B-DC-08-A5	SAP

To get this environment set up initially, you would also need to make sure that you added all ten Citrix servers to your IMA data store. One

of the great things (in this case) about the IMA data store is that it identifies farm member servers via their NetBIOS name—not via IP address or SID. This means that you can actually add all of your servers to the IMA data store by running a one-time script against the data store to add all the server records. At this point you don't have to worry about assigning them any published applications.

Ok, so now we have a Citrix farm with ten servers added to it. Now you can fire up the Citrix Presentation Server Console and create your published applications. Feel free to publish as many as you want. It doesn't really matter which physical servers you publish them to. What really matters is that you define your published applications as you like them.

Now we can look at what needs to be done when a server boots up. Depending on the physical server's MAC address, the server will boot and mount either the Office or the SAP vDisk. (And of course since these vDisk files are read only, it will also create its cache file somewhere.) A startup script on the Citrix server is necessary to tie this all together.

When the Citrix server boots up, the IMA service will start and connect to the IMA data store that's specified in the mf20.dsn file. After that, we want the server to run a custom startup script. We would create two startup scripts—one that we would add into the Office vDisk file and one that we would add into the SAP vDisk file. Our startup script would do a few things.

- It would query the Windows computer name which would be unique for each server. Ardencc takes care of this for us by tying Windows computer names to MAC addresses.
- It would use MFCOM to contact the IMA data store to remove the server as an available server for any published applications it was previously servicing.
- Again using MFCOM, it would add the server to the available server list for the published applications based on the applications that are installed in that vDisk. In other words, the startup script in the Office vDisk would add this server into the various published application lists for the Office silo, and the script in the SAP vDisk would add itself to the SAP applications.
- If we're using Citrix policies applied to IMA server folders, the script would use MFCOM to move the server object to the appropriate IMA folder for the silo. Again this part of the script would vary depending on the vDisk.
- It will enable logins. (Since we have these startup activities, we would want to create our vDisks so that the servers initially boot up with logins disabled.)

That's it! The beauty of this is that it makes no difference what the IP address or server name is. The server startup script process is what ensures that server is added to the published application list. You can move servers around simply by pointing them to a different vDisk in the Ardencc admin tool. You don't have to "pre-configure" anything—your startup script handles it all.

There are a few other hidden bonuses here. First of all, when you want to add a new server to your farm, this process will take all of 30 seconds. All you would have to do is:

- Run a quick MFCOM command-line script to physically add the new NetBIOS name to the server farm's IMA data store.
- Change the boot order preference on your new server so that it boots to the network instead of to the local disk (if it even has a local disk).
- Open the Ardence admin tool to specify which vDisk (and therefore which silo) you want this new server to belong to based on the new server's MAC address.

Another hidden bonus is this: Imagine if you have a server failure. No longer do you have to have N+1 redundancy in each silo. Now you can have a single "extra" server that is farm-wide. If any server in any silo fails, you just point the extra server to the proper vDisk in the Ardence admin tool, boot it up, and you're all set!

Finally, since this infrastructure makes it so easy to move servers between silos, you can have "dynamic" silos that grow and shrink on demand. Imagine "stealing" one server from each silo at the end of the month to add to the silo that does all of your financial processing or hosts other month-end high-usage applications.

Another cool thing about this architecture is that it of course can be used beyond Citrix servers. You can have as many different vDisk images as you want. (Ardence licensees the product based on physical servers, not virtual disk images.) You could have servers that were Citrix servers by day and enterprise backup servers by night! The Ardence administrative tool lets you specify different vDisk images for servers depending on the time or day that they are booted. So for example, you could have a silo of several servers that are booted up each morning to a Citrix vDisk. Then at night they reboot and mount a backup software vDisk and perform backups of other servers. Then at 6:00AM they reboot again and mount the Citrix vDisk for the next day's work.

The Performance Impact

One of the first things that people question with this architecture is performance. They assume that since physical disk blocks are being transferred across the network instead of across the SCSI cable, the performance must be terrible. In fact this is not the case at all. Consider these numbers.

The Ultra 320 SCSI bus can support up to 320 megabytes per second. However, that's the maximum speed of the data bus itself. In reality, disk read/write speed is limited by the physical speed that the magnetic bits on the spinning platter can be read/written by the drive head. As per documentation from the big three hardware vendors, a 3.5" 15k RPM server hard drive has a transfer rate between 57 and 86 megabytes per second. (This varies depending on where on the disk the data is coming from, since data near the outer edge of the physical platter moves under the read/write head faster than data near the inner edge.) They talk about a "burst" rate of 320MBps, but that's when the data is coming from the drive's cache and not the physical magnetic surface.

Today's networks are 1Gbps (or one thousand megabits per second). To compare the two, we need to convert the disk speed in megabytes to the network speed of megabits, so we take the disk maximum speed of 86 megabytes per second * 8 = 688 megabits per second. Even if we factor in an extra 10% for protocol overhead, you'll see that a 1Gbps network is faster than a 15k RPM disk.

This does not mean that mounting a vDisk across a network will be faster than a local physical disk, because the vDisk is still ultimately stored on a physical disk. It just means that the network will not add a bottleneck to the overall disk access equation. In fact, depending on your scenario, a centralized vDisk might be faster than a local disk. (For example, a centralized vDisk file on a 15k RPM disk versus local disks that are 10k RPM.)

As with all environments, some care will need to be taken to design the proper disk architecture. If you have 100 servers all sharing a single vDisk file on a single disk; that may introduce a bottleneck that you wouldn't have if your 100 servers were all using local disks. However, if your centralized vDisk file were on a RAID 5 volume with 256MB cache configured 100% for disk read operations, and your individual servers' vDisk cache files were stored on their local hard disks, then you would only be reading data from your centralized disk. In this case you could have 200 or more client servers running from the single public vDisk file before performance was worse than having an old-fashioned local disk on each server. (Of course the exact client-to-vDisk ratio depends on your environment, but keep in mind that the central vDisk is only really stressed when the client servers boot up.)

“Personalizing” Individual Servers that share a vDisk

If you've been following along with this process so far, then there is still probably one major question you have. Namely, each Windows server must be unique in your environment. It must have its own name, IP address, and security identifier. On top of that, some applications might require a specific INI file or registry settings. If all of your servers are booting off of the same public read-only vDisk image, then how does this work?

This is where the Ardence technology steps in once again. Think back to the boot process. Remember that the network bootstrap points a booting computer to an Ardence server. The Ardence server has a database of all the client computers. Therefore when the Ardence server receives a client boot message to mount a vDisk, it checks the MAC address of the client computer against its database and can inject the proper computer name. In the case of Windows clients operating in a domain, Ardence also intermediates the communication between the domain controller and the client to maintain the Active Directory credentials between sessions.

Furthermore, Ardence allows you to configure name/value pairs for each client computer in what they call “personalization.” The way this works is that you tie these “personalization” settings to each MAC address and public disk image combination. For instance, you might have a public disk image called “Citrix SAP Server.” You would use the Ardence management tool to specify the MAC addresses of the servers on your network that you want to boot using that image. You would then add your own name/value pairs (these can be whatever you want) for each MAC address. For example, you might configure the

server with MAC address 00-0E-9B-DC-08-57 to have a name of "Citrix IMA Datastore Location" with a value of "SQLServer02." When this server boots, the Ardence software will drop an INI file into Windows that contains these name/value pairs.

So what good are these name/value pairs? It's up to you to do whatever you want with them. For example, maybe you want one read-only public disk image for many Citrix servers, but you want some Citrix servers to access the IMA data store on SQLServer01 and you want some to access a replicated copy on SQLServer02. This is specified in a DSN file called "mf20.dsn" that lives in the Citrix folder on the server. The server's IMA service starts automatically when the server starts and refers to this DSN to see what database it should connect to.

In the Ardence world, you would edit your master public vDisk image and configure the IMA Service to be a "manual" startup instead of an "automatic" startup. To do this, you would configure a system startup script on the vDisk to read the Citrix server value from the Ardence INI file, modify or copy the DSN as needed, and then start the IMA Service.

Creating vDisk files

Ardence's entire technology architecture is based upon the vDisk files that your computers mount over the network. Creating these vDisk files is very straightforward. You build a computer as normal and then install a little Ardence component via an MSI file (in the case of Windows computers). The MSI file does two things:

- It installs the Ardence disk drive device driver so that future systems booting to the image will be able to access it via the Ardence protocol across the network.
- It installs a utility that you can use to "snapshot" the disk drive to create the vDisk image file. This is kind of like taking a ghost snapshot except that somehow Ardence has figured out how to do this live while Windows is running without having to boot into a utility mode or anything.

You use this tool to create the vDisk file on the network and then add the vDisk file into your Ardence configuration database and start assigning the vDisk file to computers. If you need to create any system startup scripts (as mentioned earlier), simply create these scripts on your computer and configure everything (such as disabling ICA logons, etc.) before using the Ardence tool to create the vDisk snapshot.

Once you have your vDisk files on the network, maintaining them is pretty easy too. You can use the Ardence admin tool to make a read/write 1-to-1 instance of a public read-only shared vDisk file. This essentially means that you can boot a computer to a "one off" read/write instance of the vDisk, make your changes, and then set that vdisk back to a read-only shared vDisk file. The Ardence framework can even manage version control for these, so if you start booting your computers to the new vDisk and there is a problem, you can use the admin tool to instantly point them back to the old vDisk. All you have to do to "fix" your broken computers is to reboot them, and the Ardence server will guide them to the old vDisk file.

If you only need to make simple changes to your vDisk image, Ardenance has tools where you can mount the image file as a drive in Windows. You can then use Windows explorer to add, remove, or modify any files as needed.

Using Ardenance with VMware, Softricity, and other “alternate” application management platforms

One thing that’s interesting to me about Ardenance is how it fits into the larger world view of applications. It’s interesting because Ardenance is really a “horizontal” solution that fits well with traditional PCs, VMware desktops, bladed PCs, Softricity-managed applications, and of course Citrix and server-based computing applications. The key here is that Ardenance is virtualizing the physical disk access.

In a world of VMware servers, you can configure your VMs to boot from the network and they can boot and mount Ardenance vDisks with all of the advantages that we discussed previously. (Or you could create an Ardenance vDisk of the VMware host OS and virtualize the disks at that level.)

Ardenance also compliments Softricity. Softricity does a great job of virtualizing and streaming applications. The problem with Softricity is that you still need to have the base Windows OS on a piece of hardware before you can use Softricity. The problem with Ardenance is that it handles the base OS, but you then need to install your applications into your vDisks and still deal with server silos. If you combine the two technologies together then you really have an interesting solution.

From a desktop PC standpoint, one of the main drawbacks to Citrix is that in the quest to bring management back into the datacenter, you end up bringing all application execution into the datacenter. That’s great for security and outside-the-firewall application access, but it’s really not the right choice for corporate inside-the-firewall application usage. With Ardenance, you can let some applications run locally on desktops while still managing them via public shared vDisks, and then of course use Citrix for the specific applications where it makes sense.

The Bottom Line

Ardenance has been around for 25 years, with most of that time focused on the low-level interactions between an OS and the hardware. (In fact, Ardenance is the company that Microsoft chose to write Windows NT Embedded.) Their enterprise products allow you get the benefits of centralized management with local processing, a crucial addition to any Citrix farm, with a price of about \$600 per physical server.

About the Author

Brian Madden is an independent technology analyst, author, and thinker based in Washington DC. He’s written several books and hundreds of articles about Citrix and server-based computing technology. Brian is a three-time Microsoft MVP and the creator of BriForum, an annual independent server-based computing technical conference. He’s also the editor-in-chief of BrianMadden.com, the world’s most popular server-based computing technical website with millions of visitors per year. More information is available at www.brianmadden.com and www.briforum.com.