



On the road to the virtual desktop

Windows application virtualization and streaming solutions from Microsoft, Symantec, and Thinstall are laying the groundwork for a subscription-based, click 'n' run future. Imagining the possibilities, we put them through a simple SaaS performance test

By Randall C. Kennedy

September 24, 2007

Click 'n' run. It seems like such a simple concept. Surf up to a Web page, select the desired application from a list, and click. Voila! Microsoft Word appears on your desktop. Or Excel, or Adobe Photoshop... you name it.

[Symantec SVS won an InfoWorld Technology of the Year award. [See the slideshow to view all winners in the platforms category.](#)]

In practice, click 'n' run is a nightmare to implement. Setting aside the licensing issues (and they are legion), the fact remains that delivering traditional fat client applications over the Internet is a technological hurdle on the scale of enabling pigs to fly. First, you have to package the code in a way that allows it to run without actually installing anything. That's where virtualization comes in. Then you need to find a way to deliver the bits to the end-user without choking their network connection or leaving them helpless when they no longer have access to the distribution point. Here streaming and caching are the keys.

Previous attempts at Web-based application distribution have focused mostly on kiosk-style computing, using a virtual machine or terminal session running on a back-end server to deliver a "screen-scraped" UI to the remote user. However, with the emergence of application virtualization solutions from Softricity (now part of Microsoft), Altiris (now part of Symantec), and Thinstall, the industry is poised for an explosion of new and potentially revolutionary delivery models.

Application virtualization solutions at a glance

Thinstall has the cleanest virtualization environment, requiring no client agent, but lacks the application management features that SoftGrid and SVS Pro provide. Only SVS Pro currently supports a "click 'n' run" deployment in which users can launch virtualized applications from a Web browser.

Feature	Microsoft SoftGrid 4.2	Symantec SVS Pro 2.1	Thinstall Virtualization Suite 3.2
Windows Vista client support	Yes	Yes	Yes
Click 'n' run (Web deployment)	No	Yes	No
Directory integration/security	Yes	Yes	No
Client-side persistent caching	Yes	Yes	No
User mode only execution	No	No	Yes
Headless applications/services	No	Yes	No
Supports 64-bit Clients (Vista)	No	No	Yes

[Click for larger view.](#)

These three solutions virtualize the interaction between Windows programs and the Windows OS resources they depend on, including the file system and system registry, allowing them to run in isolation from the underlying desktop. (For the differences in how they go about it, see my writeup on [application and desktop virtualization](#) in "Virtualization: Under the hood.") Combine the virtualization capabilities with streaming servers, as two of the vendors have done, and click 'n' run, on-demand application deployment is just a step or two away.

What makes application virtualization so compelling is its immunity to the very issues that torpedoed the kiosk solutions. For starters, virtualized applications are modular. Though streamed by default, virtualized applications can be configured for offline use, either through caching or by simply copying the application image to the client. And though isolated from the local system (because the application's registry access and private file set are virtualized), they can still interact with local resources, seamlessly accessing the PC's storage and print devices, for instance.

But despite having the right general plumbing, none of these first-generation solutions is designed specifically with a subscription-based computing model in mind. Microsoft SoftGrid and Thinstall Virtualization Suite are still aimed at internal enterprise deployment, whereas Symantec SVS Pro -- by virtue of its integration with a streaming server component from partner AppStream -- is closer to the goal of a true click 'n' run format. All Symantec needs to do is work out the optimization kinks.

It's worth noting that, although the focus of SoftGrid is still internal, the 800-pound gorilla in Redmond has everything it needs to deliver a future Office suite via the Web. The future of software distribution is subscription-based click 'n' run; the question is no longer if, but when. The three platforms reviewed here are helping pioneer the transition.

A simple test of speedy delivery

To better understand the behavior of each application virtualization solution, I constructed a rudimentary test bed featuring Microsoft Office 2003. Using the packaging tools from each solution, I created a virtualized Office image and then used each solution's distribution mechanism to deploy the image to a Windows XP-based client session.

During deployment, I monitored the client session to measure peak network traffic, recording the total bytes sent and received per second in Windows Performance Monitor. I also timed the initial launch of multiple Office applications (Word, Excel, PowerPoint, and Access) in an effort to gauge the efficiency of the three delivery methods.

Note: All testing was conducted under Microsoft Virtual Server 2005 R2 (SP1). In the case of SoftGrid, a Windows Server 2003-based Active Directory environment was created to support the SoftGrid server components. For Symantec SVS Pro, I used a non-Active Directory Windows Server 2003 instance. Thinstall didn't require any server components; however, I did use the SoftGrid server to host a cached share point containing Thinstall-packaged executables.

Microsoft SoftGrid 4.2

When I first reviewed SoftGrid in July 2006 ("[SoftGrid opens a stream toward application manageability](#)") I found a product with tremendous promise saddled with an overly complex sequencing process and myriad runtime limitations. It's now a year later and virtually nothing has changed. Version 4.2 is still plagued by UI quirks and omissions, the failure to automatically share the Content folder during the server installation process being the most glaring (unforgivable, really). There has been no real improvement in application compatibility; for example, SoftGrid still can't handle applications that employ headless services (such as systems management agents). In fact, other than support for Windows Vista as a client OS, SoftGrid is effectively unchanged from its previous incarnation.

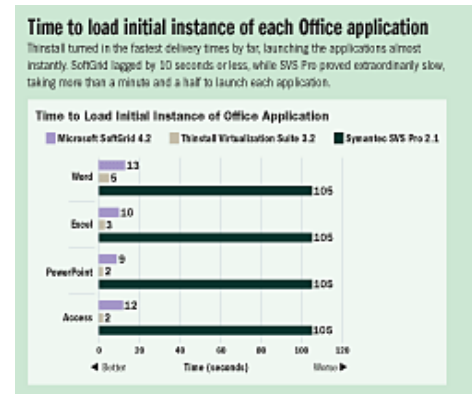
All of which makes the logic behind Microsoft's decision to acquire SoftGrid from its original developer, Softricity, all the more apparent. The Redmond giant has a history of obtaining half-baked products for the purpose of gaining access to a particular core technology -- in this case, the SoftGrid virtualization client and streaming engine. Microsoft needs these code nuggets in order to flesh out its grand scheme for a subscription-based computing model (see my blog post, "[Microsoft's Virtualization Endgame](#)," for details). And barely 12 months since the acquisition, the gutting has begun.

For example, Microsoft recently announced a new packaging tool for SoftGrid, one that allows customers to redistribute SoftGrid-encoded applications using Microsoft's Windows Installer service. This makes it easier for customers to decouple the virtualization client layer from the streaming server layer, effectively making the SoftGrid OSD package format a viable stand-alone delivery platform, much as the company's VHD (Virtual Hard Disk) format functions for virtual servers. It also helps SoftGrid applications play better with existing configuration management environments because the OSD is wrapped within the widely supported Windows Installer (MSI) package format.

MSI compatibility is only the first step. Microsoft will likely do an even more thorough strip job as time goes on, dumping much of the existing management UI, which is awkward at best, in favor of an even more seamless integration with MSCCM (Microsoft System Center Configuration Manager). The company doesn't think much of SoftGrid as a separate product, as evidenced by its decision to release what was essentially the entire Softricity product catalog as a set of free tools (the Microsoft Desktop Optimization Pack for Software Assurance) for customers in the Microsoft Software Assurance program. Clearly, this is a product acquisition in which the sum of the parts is considered greater than the whole.

I tested SoftGrid 4.2 running under Microsoft Virtual Server 2005 R2 (SP1). As with previous versions, the SoftGrid server components require a Microsoft Active Directory environment in order to function. There are a variety of additional requirements, including Microsoft Internet Information Services and Microsoft Management Console 3.0, the latter of which must be [downloaded separately](#). The SoftGrid installer does a good job of flagging any missing pieces, but does not offer to install them from within the setup wizard -- an un-Microsoft behavior that will likely be fixed when the company further integrates SoftGrid with MSCCM.

During benchmark testing, SoftGrid streamed a virtualized Office 2003 package to the desktop with initial startup times of 13, 10, 9, and 12 seconds for Word, Excel, PowerPoint, and Access respectively, all while consuming from 92Mbps to 110Mbps of network bandwidth. Subsequent streaming operations to support accessing additional functionality outside of the initial startup blocks triggered another 50Mbps to 80Mbps of network traffic. Caching the entire Office suite for offline use took just under two minutes while pushing peak bandwidth utilization to 124Mbps.



[Click for larger view.](#)

Overall, SoftGrid remains a solution with tremendous potential, but is marred by unresolved usability and compatibility issues. No doubt the compatibility issues will become moot as Microsoft narrows the product's focus to its own delivery needs while the usability problems will be addressed when SoftGrid is eventually vivisectioned on the System Center chopping block.

Symantec SVS Pro 2.1

Symantec's SVS (Software Virtualization Solution), acquired when the company purchased Altiris in late 2006, has long provided one of the easiest to use and most flexible application virtualization platforms (see my March 2006 review, "[Altiris shakes up Windows configuration management](#)"). Its effortless handling of headless services and other complex application types have made SVS the darling of the IT skunkworks crowd, as evidenced by the popularity of the [Juice online developer community](#). However, in direct comparisons to SoftGrid and even Thinstall, SVS always came up short in the area of application delivery. Complex SVS "layers," such as our Office 2003 test bed, can be quite large (150MB to 300MB or more, depending on configuration), making them impractical to deliver over large, geographically dispersed enterprises.

To address this deficiency, and to bring SVS more on par with SoftGrid, Symantec has partnered with AppStream to add a streaming delivery mechanism for SVS-packaged applications. However, as is often the case with mixed vendor/OEM solutions, the integration of the participating components is less than perfect, leaving the user to navigate disparate UIs rife with redundancy.

For example, to deploy an SVS package via AppStream you begin within the confines of the SVS administration utility. After you've captured an application install you need to export the layer to a VSA file. Next, you copy the VSA file to a system running the AppStream packager where you open the VSA and convert it into an AppStream ZIP package. Once that's done you need to upload the package to the AppStream server and then provision it for distribution. All told, you're forced to navigate across four different UIs (SVS admin, AppStream packager, AppStream upload/import utility, AppStream Web console) spanning three different runtime platforms (Windows, Java, Web). By contrast, with SoftGrid you remain within a single UI and platform (Windows), and importing an application requires just a single step (copying the package to the Content folder).

This spotty integration also extends to the delivery process. When testing SVS Pro against the aforementioned Office 2003 test bed, I clocked AppStream at no less than 1 minute and 45 seconds from initial user request to the appearance of the virtualized application. During this time AppStream was downloading nearly 50MB of data into the local SVS cache and generating just under 80Mbps of network traffic.

Compare this to the 9 to 13 seconds for SoftGrid or the 2 to 5 seconds for Thinstall, and you can see that, at least during the initial program load to cache, SVS Pro is far from a speed demon. However, on the positive side this large initial blob includes the most common code blocks required for all of the packaged executables. So while it might take longer to launch the requested application for the first time, subsequent launch requests -- either to the first application used or any of the other applications in the SVS package layer (in this case Word, Excel, PowerPoint, and Access) -- are serviced almost instantaneously.

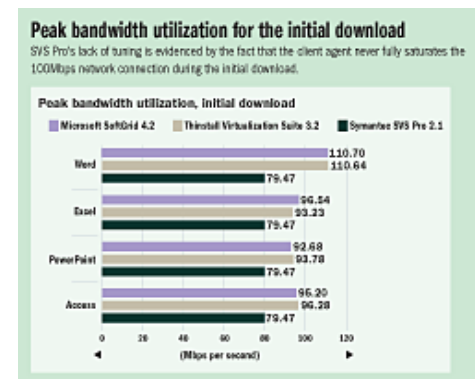
An analysis of subsequent streaming requests (i.e. grabbing additional code to support functions not downloaded in the initial 50MB or more blob) showed a much more granular access pattern. All of which seems to point to an architectural tradeoff when dealing with streaming SVS packages: You need to download more of the package blocks to get the first application started, but these additional blocks help to eliminate the need for subsequent roundtrips to get additional applications loaded. To put this in perspective, you have to add up the initial startup times for Word, Excel, PowerPoint, and Access under SoftGrid (44 seconds) before you've achieved the same degree of functionality that the initial SVS Pro download provides.

One area in which the SVS Pro solution outshines both SoftGrid and Thinstall is in its support for Web-based access to packaged applications. The AppStream Web portal makes it easy for users to surf up and access the desired applications in a kiosk-like, on-demand fashion. In this regard, SVS Pro provides a solution that is closer to the subscription-based delivery model that has been the holy grail of commercial developers for nearly a decade.

ThinstallVirtualization Suite 3.2

Every time I review a version of Thinstall I can't help but remember a joke I heard about something being "a riddle wrapped in an enigma and surrounded by mystery." Thinstall is the riddle. Or is it the enigma? One thing's for sure: The platform's underlying architecture continues to be a mystery to even savvy IT experts. Does it really work without a client agent and without loading anything into kernel mode? What about streaming? What does the company mean by "it delivers the benefits of streaming without the server?"

The answers to the first two questions are yes and yes, while the third is a solid "maybe." It's true that Thinstall requires no client agent or server-side components; everything a Windows PC needs to create the virtual runtime and enable the application loads dynamically from the packaged executable image. (See my February 2007 review, "[Thinstall removes barriers to application virtualization](#).") Furthermore, the company claims that, since Thinstall controls the application's loading sequence, by supplanting the usual program header logic with its own virtual environment bootstrap code, it can directly modulate which bits are passed over the wire to service a particular startup request.



[Click for larger view.](#)

For example, in the case of our Office 2003 test bed, Thinstall's setup capture utility transformed our default installation into a single, large (more than 300MB), compressed file, with multiple shortcut-type redirector files taking the place of the normal Office executables. In practice, launching one of these pseudo-executable shortcuts prompted Thinstall to access the larger code image file and extract only those bits necessary to assemble the virtualized runtime image, a fact evidenced by our test results: When launching Word, Excel, PowerPoint, and Access, Thinstall generated peak network utilization on par with SoftGrid (92Mbps to 106Mbps). It also blew the doors off both SoftGrid and SVS Pro when it came to initial application load times. Whereas the two "streaming" solutions took at least 9 seconds (SoftGrid) and 105 seconds (SVS Pro) to load an application for the first time, Thinstall did it in 2 to 5 seconds, depending on which application was being requested.

Of course, no matter how efficient the download, you don't want your users hammering on the server every time they decide to load a Word document. The lack of client-side caching of code images has been one of the persistent knocks against Thinstall. And though you can work around the issue externally -- for example, under Windows by enabling caching on the share point -- it's still something Thinstall's designers need to address directly.

Another issue they need to address is Thinstall's reliance on SMB-based (Server Message Block) sharing as its only network distribution mechanism. Both SoftGrid and SVS Pro, though not yet fully optimized for Web-based delivery, are at least closer to that goal in that they provide a non-SMB delivery vehicle (streaming), one that should be more palatable to security-conscious administrators.

Thinstall executives claim to be working on a persistent caching mechanism for Thinstall and say they'll address the SMB sharing limitation in the near future. In the meantime, the company has begun talking up its new License Manager product which will hopefully address another of the product's shortcomings: the inability to enable, disable, and monitor access (including license compliance) to Thinstall packaged applications.

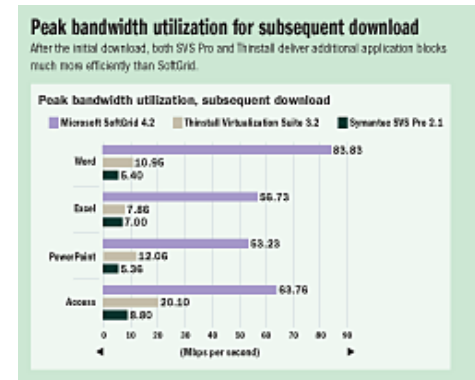
Thinstall License Manager will be released as open source, allowing IT shops to customize the PHP-based (PHP: Hypertext Preprocessor) server components to meet their specific requirements. Until then, potential Thinstall customers will need to weigh the advantages (raw speed plus no client agent or streaming servers required) against the disadvantages (reliance on SMB shares and the absence of an integrated caching mechanism) and hope that the company comes through on its promises to flesh out the Thinstall architecture.

Picking a winner

Neither Thinstall nor SVS Pro nor SoftGrid is fully baked. But while all of them have quirks and limitations, only one of them delivers an experience that's within striking distance of our stated target of a true click 'n' run solution: Symantec SVS Pro 2.1. Despite the awkward integration of the native virtualization component and the OEM delivery mechanism, Symantec's choice of AppStream as its streaming partner yields additional dividends, such as the ability to stream both virtualized and non-virtualized applications from the same platform. SVS Pro is the most complete of the three platforms, and an initial one time delay of a minute or so -- which will disappear after the application is cached -- is a minor nit against an otherwise impressive solution.

SoftGrid, by contrast, is a product in transition. The old bits are getting stale, but that's to be expected for a code base that is in the process of being strip mined. Expect to see pieces of SoftGrid technology popping up all over Microsoft's product line with the end game of delivering an on-demand platform from which the company can distribute most of its wares. In the meantime, the product that was Softricity's crown jewel is now available as a freebie for Microsoft's most loyal customers. These shops will want to take advantage of the opportunity to kick SoftGrid's tires and to prepare for what will doubtless be a major push by Microsoft to seed its version of application virtualization throughout the enterprise computing landscape.

The odd man out in the trio is Thinstall. Thinstall's integrated Virtual Operating System is still a technical marvel. Where else can you find so much muscle crammed into a 300K file header? However, the company's lack of progress in building out the management and delivery mechanisms has left Thinstall looking more and more like a one-trick pony. Those lingering deployment issues -- no caching, SMB-only streaming -- need to be addressed immediately, while the more visionary elements of the Thinstall executive team must start painting a bigger picture before enterprise customers lose interest. Perhaps the best case would be if Thinstall were acquired by a larger player (IBM and HP come to mind) who can help them weave their innovative virtualization tool into a cohesive delivery and management framework to counter those of Microsoft and Symantec.



[Click for larger view.](#)

 [Print this](#)