exo.performance.network
(www.xpnet.com)

Application Virtualization 2008-2009

Assessing the Architectural and Performance Characteristics of Four Leading Windows Application Virtualization Solutions

Introduction – Why This Report?

As IT organizations consider their future configuration management plans, many are evaluating application virtualization solutions as a means for reducing software installation complexity. The capability to capture an entire Windows-based application suite into a single, virtualized package, and then to distribute this image to users with minimal installation footprint, makes application virtualization an attractive option for organizations seeking to minimize their security surface area and eliminate legacy compatibility issues.

Currently, IT shops can choose from a selection of four well-rounded Windows-based application virtualization solutions: Microsoft Application Virtualization (App-V) 4.5; VMware ThinApp 4.0.1; Citrix XenApp 5; and Symantec Software Virtualization Solution (SVS) Pro 2.1. All of these platforms share various common traits, including the capability to stream virtualized images from a network file share or web server. However, when you look more closely, you find that these solutions often take very different approaches to the task of virtualizing an application's installation image.

For example, some platforms (App-V, SVS, XenApp) employ a set of kernel mode drivers and supporting services to manage the virtualization process, while others (ThinApp) embed their virtual environments directly into the application package. This choice of virtualization model directly affects how and where a virtualized application may be deployed, and can also greatly affect the packaged application's performance and runtime characteristics. In fact, when you strip-away the various layers of configuration tools and management interfaces, it is the core virtualization model that ultimately determines whether or not a given platform meets a customer's requirements for flexibility, compatibility and performance.

Hence our motivation in developing this first of its kind Special Report on application virtualization: The need to thoroughly explore the architectural underpinnings of the leading application virtualization solutions and to formally quantify their impact on application resource consumption and performance.

Our goals for this project were relatively straightforward. First, we would analyze and document the core architectural characteristics of each solution. Then, using a combination of test scripts and monitoring agents, we would assess each solution's execution overhead by measuring the time required to complete a set of common business productivity tasks and correlating this with various system and process metrics (CPU utilization, memory consumption, etc.). It was our hope that, by thoroughly documenting our experience testing and evaluating application virtualization solutions, we could paint a picture of the pros and cons of each architecture while establishing their suitability to the task of virtualizing applications across a range of deployment scenarios.

Note: All test scenarios were constructed using the DMS Clarity Studio workload simulation and performance testing framework. This multi-process platform analysis solution allows IT professionals to conduct ad-hoc scalability and performance testing against a range of client/server and office productivity scenarios. DMS Clarity Studio is available for non-commercial use at no cost to the end-user. For more information, visit the exo.performance.network web site at www.xpnet.com.

What We Found

After analyzing the architectural and performance characteristics of the five leading Windows application virtualization solutions, several clear messages emerged. First, solutions that employed a kernel mode driver or service (App-V, SVS, XenApp) introduced significant additional computational overhead, and this translated into poorer overall performance when executing common productivity tasks. By contrast, solutions that used an embedded virtualization model (ThinApp) generally delivered better overall application throughput, with overhead levels that were usually lower than those of their kernel-mode-dependent contemporaries.

Reliance on a kernel-mode interface also limited our flexibility in testing such solutions. For example, we could not easily deploy virtualized applications into a highly-secure, locked-down computing environment. Nor could we move them freely from Windows version to Windows version – each host OS required a compatible implementation of the kernel mode components (typically packaged as a “client agent”) as well as packages specifically encoded for that particular flavor of Windows (i.e. one-to-one deployment). As of this writing, only ThinApp supports deployment onto 64-bit versions of Windows XP, Server and Vista. And while this will no doubt change in the future as Microsoft updates App-V and Symantec and Citrix embrace 64-bitness, during our testing, this lack of 64-bit host OS support severely limited our ability to evaluate performance across multiple generations of Windows software and hardware.

Conclusions

With overall linear execution efficiency on average twice that of the nearest competitor, it was a self-contained (i.e. agent-less) virtualization solution – VMware ThinApp – that delivered Microsoft Office 2007 performance levels most closely approximating a non-virtualized, native installation. ThinApp’s low average CPU overhead, combined with a conservative memory footprint and excellent OfficeBench test script throughput, yielded the best overall runtime profile as measured by the number of additional CPU cycles (vs. a native installation) required to complete each script pass.

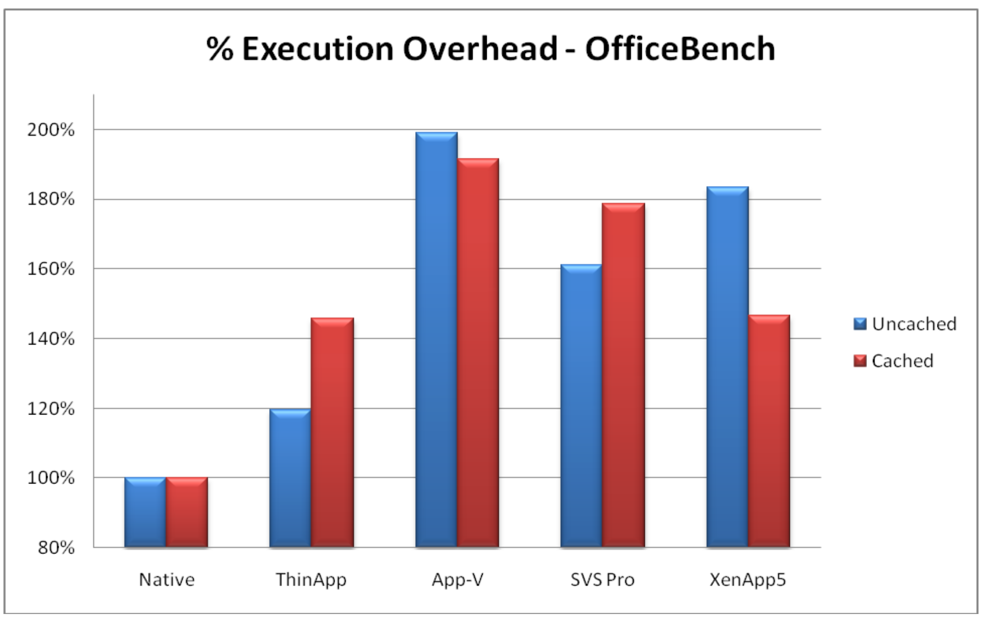

Figure 1.1 – Execution Overhead by Platform (Lower is Better)

Key takeaways:

- Application virtualization solutions that use an embedded virtualization model (ThinApp) deliver the best application throughput. Only ThinApp delivers the combination of excellent raw performance plus low overall CPU utilization, making it the better solution for organizations seeking to minimize the performance “hit” typically associated with virtualization technology.

- By contrast, solutions that employ a kernel-mode driver or service (App-V, SVS, XenApp) introduce additional layers of software complexity – including significantly higher kernel-mode activity – which translate into runtime overhead that slows the application and/or places an additional burden on the CPU. These agents also consume a considerable amount of memory, both directly – as part of the agent’s process – and indirectly, through expansion of the application’s working set.

- Agent-based solutions also introduce a new and potentially catastrophic single point of failure (kernel mode execution) that IT organizations must factor into the testing and certification of their desktop computing stacks. Functional limitations, such as the lack of support for locked-down environments and/or inability to run on specific Windows versions (x64), further complicate the application virtualization equation, forcing IT shops to invest additional resources into designing infrastructure around these planning and deployment hurdles.

Bottom Line: Customers wishing to maximize desktop performance while minimizing configuration and deployment headaches will want to give strong consideration to agent-less solutions like VMware ThinApp. ThinApp’s advantages in the areas of application compatibility, flexibility and performance make it the logical choice for IT shops seeking to leverage application virtualization to augment their existing configuration management and deployment strategy.

Methodology – How We Tested

Our primary goal in preparing this report on application virtualization was to accurately quantify the performance characteristics of the five leading vendor platforms. To achieve this goal, the DMS research team conducted comprehensive benchmark testing across all five virtualization solutions, using DMS Clarity Suite OfficeBench and Microsoft Office 2007 Enterprise.

OfficeBench is a classic linear test script that uses OLE Automation to drive Microsoft Word, Excel, PowerPoint and Internet Explorer through a series of simulated business productivity tasks. It is both platform and version independent: OfficeBench can run unmodified against six major versions of Windows (2000, XP, 2003, Vista, 2008, Windows 7) and four major versions of Office (2000, XP, 2003, 2007). As such, it is the ideal testing tool for evaluating complex virtualization environments and for conducting multi-VM server sizing and scalability studies for VDI deployments.

Note: All tests were executed against a common hardware test bed (Dell XPS M1710 w/2.0GHz Core 2 Duo CPU, 2GB of PC5300 DDR-2 SDRAM, 7200RPM disk) running Windows XP with Service Pack 3 and Office 2007 Enterprise with Service Pack 1. Individual test bed customizations are detailed in the following sections. For native baseline testing the team performed a local installation of Office and ran OfficeBench directly against this configuration.

In an effort to incorporate network and server-related factors as part of the methodology – including package streaming performance and deployment overhead – the research team installed and configured the corresponding server components for each solution. For those solutions that did not require a dedicated server (VMware ThinApp), the team used a network share point on the App-V Domain Controller as a package distribution point. All server images were hosted on a Dell PowerEdge 2950 with 8 CPU cores (Dual Xeon), 8GB of RAM and a 1.2TB SATA RAID 5. Client and server were connected over a dedicated 1GBps Ethernet link.

Microsoft App-V

Microsoft Application Virtualization (App-V) 4.5 is an agent-based virtualization solution with strong ties to Windows Server and Active Directory. During lab testing, the research team first configured a Windows Server 2008 Domain Controller, then installed the Microsoft System Center Application Virtualization Management Server. An Office 2007 virtualized application package was then created on a separate workstation, using the App-V Sequencer utility to capture the installation process and optimize the startup blocks. Next, the team copied the package to the server and provisioned it for access by a test domain user group. Finally, the package was deployed to the App-V test client via the unencrypted variant of the RTSP protocol, at which point the team used a virtualized instance of CMD.EXE to access the App-V virtual environment and launch OfficeBench.

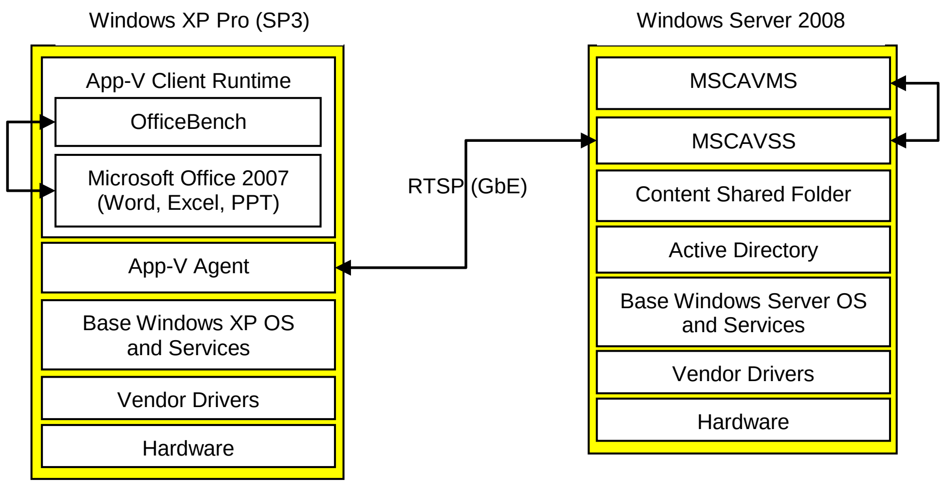

Figure 2.1 – The App-V Test Bed

The actual test process consisted of booting the system from a cold start and then executing a single pass of the OfficeBench test script in two phases:

- The first phase involved executing OfficeBench against an un-cached copy of the Office 2007 package. This, in turn, prompted the App-V agent to begin streaming the package contents from server to client, beginning with those blocks that were highlighted during package sequencing. OfficeBench was executed against this configuration in order to provide data on streaming-related network overhead as well as to establish raw startup times for initial client access to a newly deployed application.

- The second phase involved executing OfficeBench against a fully cached copy of the Office 2007 package, using the App-V agent's built-in caching mechanism to store the package contents locally on the test client. OfficeBench was executed against this configuration in order to establish the raw performance characteristics of App-V in an "offline" scenario where no additional streaming overhead was involved.

In each case, the DMS Clarity Tracker agent was configured to run in parallel with the DMS Clarity Studio OfficeBench workload and to collect system and process metrics while the test script was executing. All test results and metrics data were then uploaded to a centralized web repository (the DMS Clarity Analysis Portal) for later analysis.

VMware ThinApp

VMware ThinApp 4.0.1 is an agent-less application virtualization solution that features an embedded virtual operating system (VOS) that is deployed as part of the application package. During lab testing, the research team re-used the App-V Domain Controller server as the host for an SMB file share containing ThinApp packages. The team first created the Office 2007 package by using the ThinApp Virtualization Studio application to capture the installation process on a separate system. They then uploaded the package to the file share and accessed it using a UNC path. Finally, the team executed the test scenario by loading a virtualized instance of CMD.EXE from the package and then launching OfficeBench from within the ThinApp VOS.

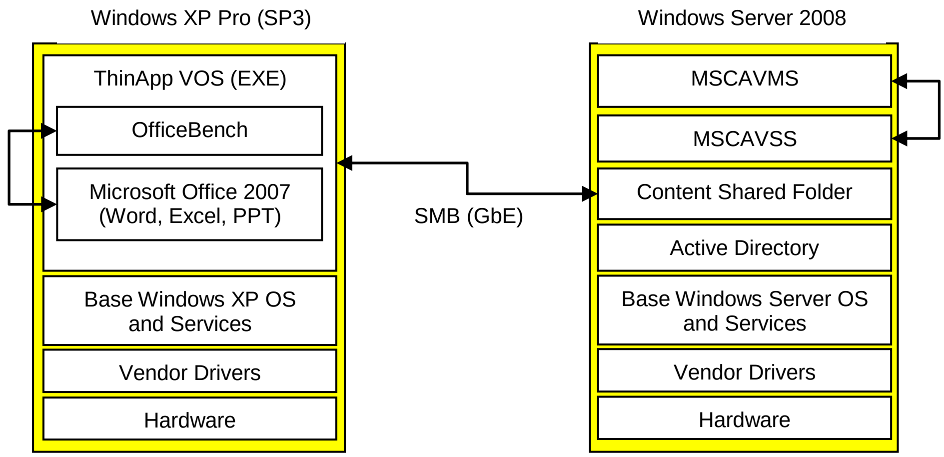

Figure 2.2 – The ThinApp Test Bed

The actual test process consisted of booting the system from a cold start and then executing a single pass of the OfficeBench test script in two phases:

- The first phase involved executing OfficeBench directly against the un-cached network file share. This, in turn, forced the ThinApp VOS to stream the package to the client from the source file, allowing the team to record data on streaming-relate network overhead and to establish the raw startup times for initial client access to a newly deployed application.

- The second phase involved executing OfficeBench against a fully cached copy of the Office 2007 package, using the Offline Folders feature off Windows to provide a persistent, client-side copy of the package. OfficeBench was executed against this configuration in order to establish the raw performance characteristics of ThinApp in an "offline" scenario where no additional streaming overhead was involved.

In each case, the DMS Clarity Tracker agent was configured to run in parallel with the DMS Clarity Studio OfficeBench workload and to collect system and process metrics while the test script was executing. All test results and metrics data were then uploaded to a centralized web repository (the DMS Clarity Analysis Portal) for later analysis.

Symantec SVS Pro

Symantec Software Virtualization Solution (SVS) Pro 2.1 is an agent-based virtualization platform that uses a kernel mode filter driver to virtualize access to file system and Registry resources. During lab testing, the research team first configured a Windows Server 2003 workgroup server, then installed the SVS Pro Streaming Server. An Office 2007 virtualized application layer was then created on a separate workstation, using the SVS Admin utility to capture the installation process and export the resulting layer to a VSA file. Next, the team copied the VSA to the server and used the SVS Pro Packager utility to convert it into an AppStream-compatible ZIP package and upload it to the server's package repository for provisioning. Finally, the package was deployed to the SVS Pro test client via a streaming variant of the HTTP protocol, at which point the team was able to execute OfficeBench directly against the now active SVS layer.

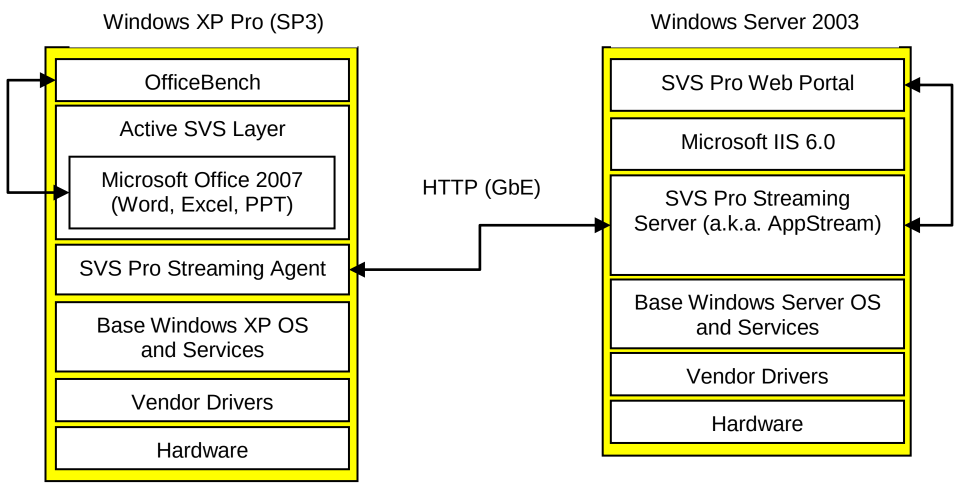

Figure 2.2 – The SVS Pro Test Bed

The actual test process consisted of booting the system from a cold start and then executing a single pass of the OfficeBench test script in two phases:

- The first phase involved executing OfficeBench against an un-cached copy of the Office 2007 package. This, in turn, prompted the SVS Pro streaming agent to begin streaming the package contents from server to client. OfficeBench was executed against this configuration in order to provide data on streaming-related network overhead as well as to establish raw startup times for initial client access to a newly deployed application.

- The second phase involved executing OfficeBench against a fully cached copy of the Office 2007 package, using the SVS Pro agent's built-in caching mechanism to store the package contents locally on the test client. OfficeBench was executed against this configuration in order to establish the raw performance characteristics of SVS Pro in an "offline" scenario where no additional streaming overhead was involved.

In each case, the DMS Clarity Tracker agent was configured to run in parallel with the DMS Clarity Studio OfficeBench workload and to collect system and process metrics while the test script was executing. All test results and metrics data were then uploaded to a centralized web repository (the DMS Clarity Analysis Portal) for later analysis.

Citrix XenApp 5

Citrix XenApp 5 is an agent-based virtualization platform with strong ties to Citrix Presentation Server and related server-based computing technologies. During lab testing, the research team first configured a Windows Server 2008 workgroup server, then installed the XenApp 5 server components. An Office 2007 virtualized application package was then created on a separate workstation, using the XenApp 5 Streaming Profiler utility. Next, the team copied the package to the XenApp5 server and provisioned it for streaming access via the XenApp 5 web portal. Finally, the package was deployed to the XenApp 5 test client via a streaming variant of the HTTP protocol, at which point the team was able to execute OfficeBench by launching it from a virtualized copy of CMD.EXE included with the package.

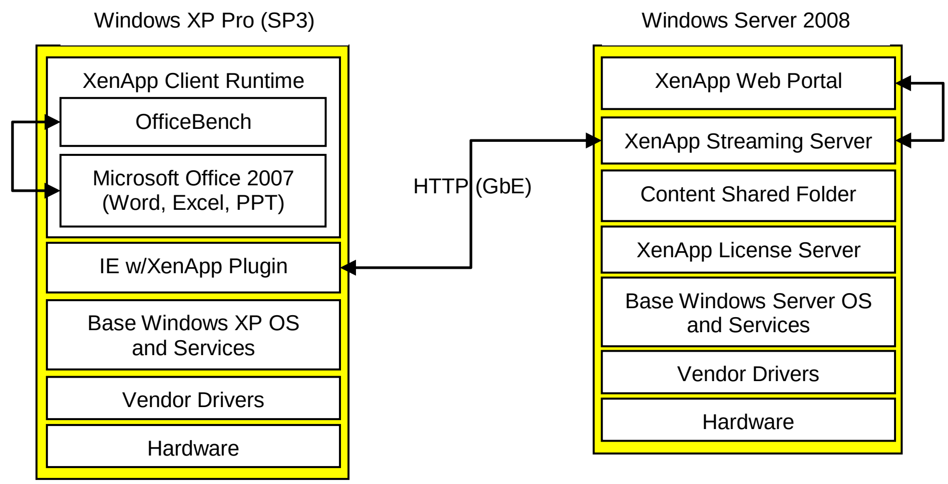

Figure 2.2 – The XenApp 5 Test Bed

The actual test process consisted of booting the system from a cold start and then executing a single pass of the OfficeBench test script in two phases:

- The first phase involved executing OfficeBench against an un-cached copy of the Office 2007 package. This, in turn, prompted the XenApp 5 browser plug-in and agent process to begin streaming the package contents from server to client. OfficeBench was executed against this configuration in order to provide data on streaming-related network overhead as well as to establish raw startup times for initial client access to a newly deployed application.

- The second phase involved executing OfficeBench against a fully cached copy of the Office 2007 package, using the XenApp 5 agent's built-in caching mechanism to store the package contents locally on the test client. OfficeBench was executed against this configuration in order to establish the raw performance characteristics of XenApp 5 in an "offline" scenario where no additional streaming overhead was involved.

In each case, the DMS Clarity Tracker agent was configured to run in parallel with the DMS Clarity Studio OfficeBench workload and to collect system and process metrics while the test script was executing. All test results and metrics data were then uploaded to a centralized web repository (the DMS Clarity Analysis Portal) for later analysis.

Notes on Installation Complexity

During the execution of this project, the research team became intimately familiar with the four participating application virtualization platforms. Over several weeks, the team had the chance to install, configure, tear-down and reinstall the core components of each solution. Along the way, they made numerous observations about the overall installation complexity and number of steps required to implement a working application virtualization environment:

- VMware ThinApp was by far the easiest solution to use and deploy. Featuring an embedded virtualization layer and no server components, ThinApp allowed the team to successfully create and deploy a virtualized Microsoft Office 2007 package in a matter of minutes. ThinApp's "Setup Capture" utility made encoding a new virtualized application a simple, wizard-driven affair, while the capability to simply copy the package to a file share or web server ensured that distributing the resulting package was entirely straightforward.

- Microsoft's App-V was the most time-consuming solution to implement thanks to its reliance on a multi-layered Windows Server back-end. Simply preparing the necessary basic services can take several hours, and on one occasion, the team wasted an entire afternoon debugging client-to-server communications issues with the RTSP protocol (a common complaint against both App-V and its architectural predecessor, SoftGrid). The actual package creation process is quite complex, requiring a high-degree of familiarity with App-V concepts and deployment paradigms.

- Symantec's SVS Pro suffers from a lack of integration between the various moving parts. For example, the AppStream server components require that exported SVS layers that are to be streamed first be converted into compatible ZIP archives via the Packager utility. Unfortunately, the utility continually stripped-out critical Registry keys from the layer, forcing the team to manually edit the layer once it had been streamed down to the client. This, and other glitches, translated into much wasted time as the time struggled to work around the various limitations and issues.

- Like Microsoft App-V, Citrix XenApp 5 requires a great deal of server preparation, including the installation of various prerequisite services and components. Configuring XenApp's licensing and building the initial web-based management site can also be time-consuming and is fraught with potential failure points. Fortunately, the process of encoding and provisioning new virtualized applications is relatively straightforward, though some familiarity with Citrix concepts and deployment paradigms is required.

Test Results

The first thing to notice about the OfficeBench scoring results is how much faster a native installation still is. Virtualization of any kind almost always incurs a performance penalty, and the application-level variety is no exception. That's why it's important for IT shops to consider runtime performance when evaluating application virtualization solutions and to opt for platforms that deliver good throughput plus low overhead.

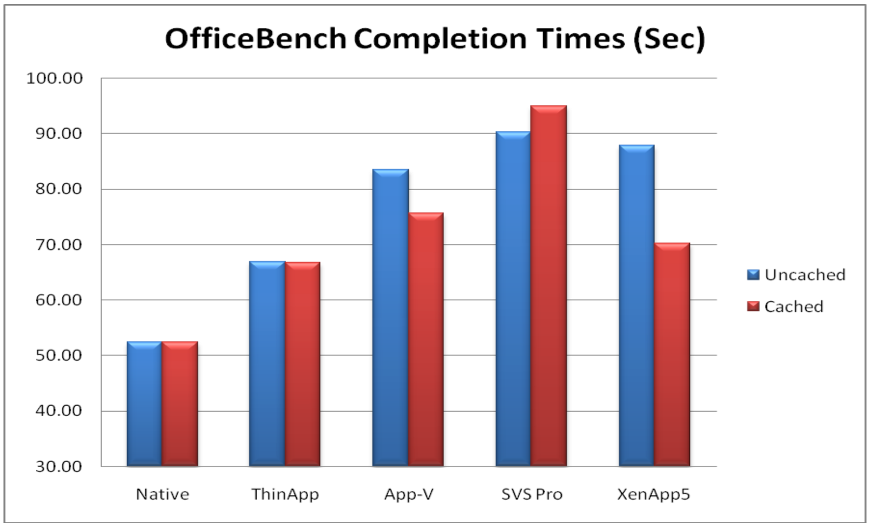

Figure 3.1 – Charting the Virtualization "Tax" (Lower is Better)

When viewed in the context of overall CPU utilization, the Office 2007 benchmark numbers take on additional meaning. For example, ThinApp's OfficeBench results look even more compelling when you factor-in the number of CPU cycles consumed by each platform while executing the test script. Hence our decision to lead this report with a chart on execution efficiency, a metric that combines the two data points to create a more accurate compound metric.

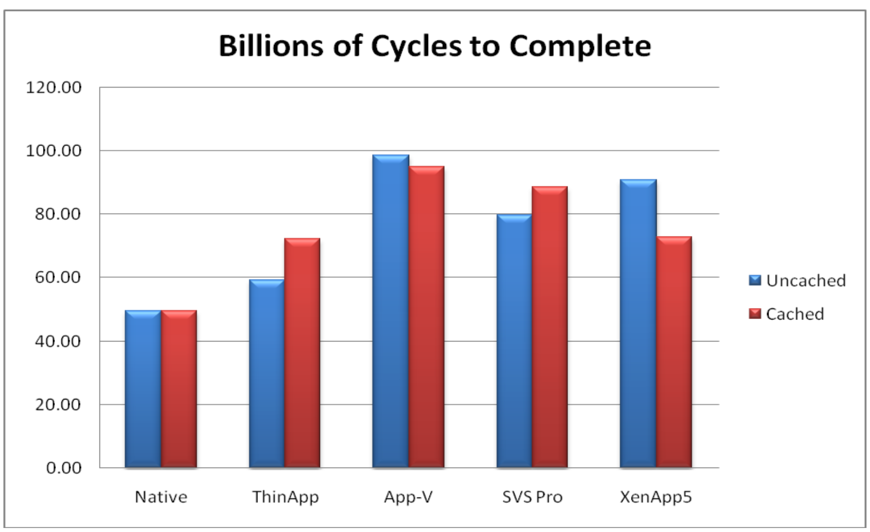

Figure 3.2 – OfficeBench CPU Consumption (Lower is Better)

Another critical metric, memory consumption, shows ThinApp again leading the pack, with a combined Office 2007 (Word + Excel + PowerPoint) working set 26% lower than the next most efficient platform, Citrix XenApp.

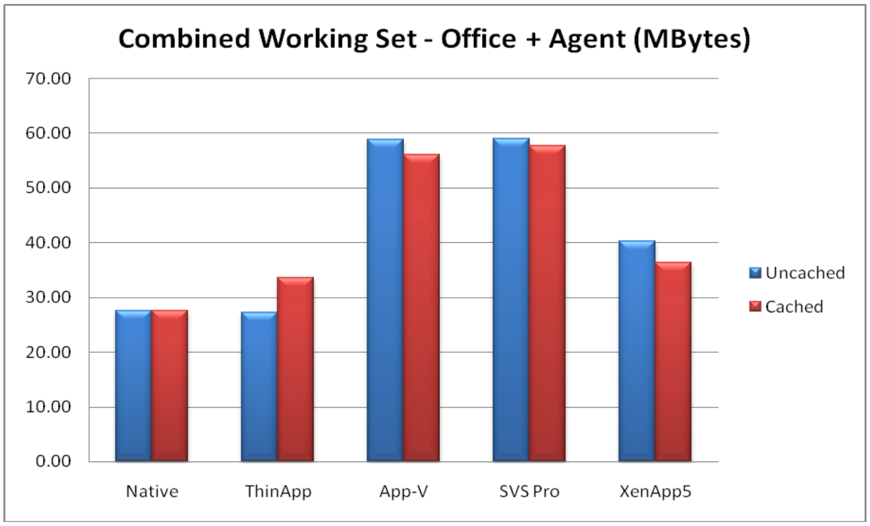

Figure 3.3 – Virtualization-Related Memory Overhead (Lower is Better)

Finally, in terms of network bandwidth utilization (during uncached testing), the team recorded a statistical three-way tie between ThinApp, SVS Pro and XenApp5, with Microsoft's App-V retaining the distinction of being the least efficient solution (53% off the pace) when it comes to streaming data over the wire.

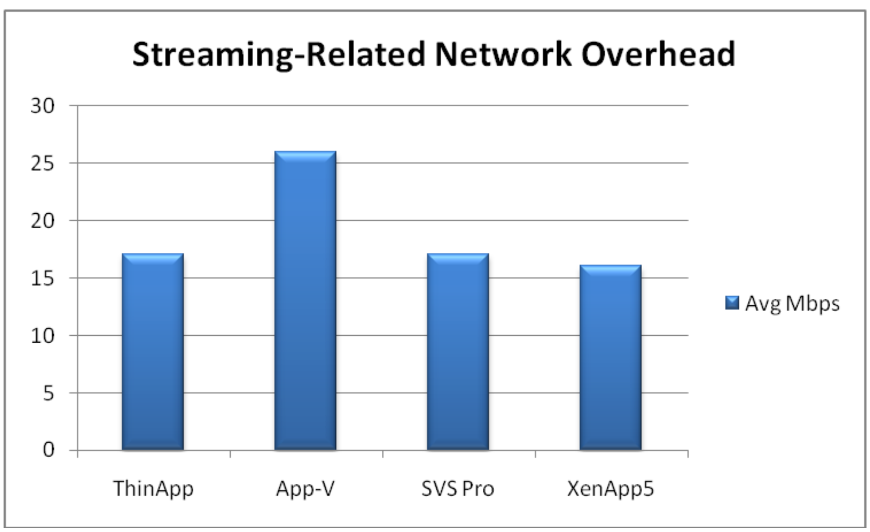

Figure 3.4 – Bandwidth Utilization During Streaming (Lower is Better)

Note: All tests were conducted over a Gigabit Ethernet LAN (NetGear FS108). The test team used the DMS Clarity Tracker agent to collect data on system, process and network metrics during each OfficeBench script pass.

Interpretation

In terms of raw script performance, the clear winner was VMware ThinApp. With an average OfficeBench completion time (uncached + cached / 2) a full 18% faster than its nearest competitor, ThinApp delivered Office 2007 performance levels that most closely resemble a native installation. More importantly, it accomplished this while consuming less memory and using fewer CPU cycles than the other platforms, giving VMware's solution a significant advantage in complex business computing scenarios.

The rest of the field fell prey to their own architectural shortcomings, specifically, their reliance on a separate agent process to manage the virtualization environment. Without exception, this de-coupled model led to higher resource consumption and poorer overall performance than the class-leading ThinApp.

Otherwise, the solutions can be summarized as follows:

- ThinApp – Low CPU and memory overhead levels; fastest OfficeBench execution times; good all-around performer.

- App-V – High CPU, memory and network overhead levels; second slowest OfficeBench execution times; poor all-around performer.

- SVS Pro – Moderate-low CPU but high memory overhead; slowest OfficeBench execution times; poor all-around performer.

- XenApp 5 – Moderate CPU and low memory overhead; slow OfficeBench execution times for un-cached test runs; acceptable performer.

About DMS

Formed in 2003, Devil Mountain Software, Inc., is an independent software development and IT consulting company focusing on Windows-based performance management solutions. The exo.performance.network is a worldwide community of Windows IT professionals and organizations working together to develop the world's first global repository of computer performance-related knowledge and data. For more information about DMS, our products & services or the exo.performance.network, please visit our web site: www.xpnet.com.

Disclaimer

This document is for informational purposes only. The information contained in this document represents the current view of Competitive Systems Analysis, Inc. (CSA), on the issues discussed as of the date of this publication. Because CSA must respond to changing market conditions, it should not be interpreted to be a commitment on the part of CSA, and CSA cannot guarantee the accuracy of any information presented after the date of publication.

This Special Report is for informational purposes only. CSA MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS SUMMARY.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. All other product or company names may be the trademarks of their respective owners.