VMware ThinApp

A Guide for Enterprise VMware ThinApp Deployments

REFERENCE ARCHITECTURE

LEGACY CONTENT: This paper contains valuable information, although some details are different for the current release of the product.

Contents

About This Guide 1

Introduction 1

Goals 1

Reference Architecture Components 2

ThinApp Packaging Framework 2

Source Folder 2

UAT Folder 2

Production Folder 2

Application Packaging Considerations 2

Determining the Execution Mode 3

Update Mechanism 3

Application Dependencies 4

Active Directory Integration 4

Active Directory Group Policy Management 4

Active Directory-Based Access Control 5

Active Directory Script- based Application Registration 5

Storage Layer Design Considerations 5

Benefits of DFS 5

Utilizing DFS with VMware ThinApp 5

Network Layer Design Considerations 6

Using Compression with Streaming Execution Mode 6

VMware View Integration 6

Desktop Persistence 6

View Composer Considerations 7

Integration of VMware View Composer and ThinApp 8

Reference Architecture Design 8

Use Case Descriptions and Diagrams 8

VMware ThinApp Test Bed 12

Virtual Infrastructure 12

Physical Server Configuration 12

Physical Network Details 15

Virtual Desktop Configuration 15

Application Configuration 15

Storage Configuration 15

Workload Description 16

User Simulation Workload 16

Unit Test Workload 16
Validation Testing 16
Summary of Unit Testing Procedure 16
Unit Testing Results 17
Summary of Scalability Testing Procedure 18
Scalability Testing Results 18
Conclusion 21
About the Author 21
Acknowledgements 21
Resources 21

About This Guide

This document provides the design considerations and architecture that customers and partners need for enterprise implementations of VMware® ThinApp, for both physical and VMware View desktop environments. Please consult both this document and the *VMware View Reference Architecture* when implementing VMware ThinApp in a VMware View environment.

Introduction

VMware ThinApp is an agentless application virtualization solution that allows IT organizations to provide applications to end users without managing the complexity of application conflicts and prerequisites or operating system dependencies. VMware virtual machine technology decouples the operating system from hardware. Similarly, VMware ThinApp technology decouples the application from the operating system, for flexibility, portability, and isolation. VMware ThinApp plugs directly into existing IT tools and processes, enabling corporate IT organizations and ISVs to deliver encapsulated application containers across a variety of operating systems without complex configuration and installation requirements.

VMware ThinApp integrates natively with Active Directory as well as with many other third-party solutions for desktop management. All the functions discussed in this document utilize native features and functions of the generally available VMware ThinApp technology. VMware ThinApp also has partnered with specific vendors to deliver customized and integrated functionality for deployment, discovery, inventory, and license utilization. This integration can further increase operational and administrative efficiencies for organizations utilizing application virtualization.

By abstracting applications from the underlying operating systems, application virtualization augments both traditional and virtual desktop solutions. Many customers have deployed application virtualization to their physical devices as a first step toward a transition to virtual desktops as hardware refresh cycles occur or operating system migrations are mandated. VMware ThinApp technology delivers the same benefits for physical desktops, virtual desktops, and terminal services-based platforms. The use of existing infrastructure for distribution, update, and registration of virtualized applications to end users allows customers to leverage the benefits of application virtualization rapidly across the enterprise.

Goals

The goal of this guide is to provide a scalable model for deployment of virtualized applications in the enterprise, addressing multiple use cases for physical and virtual desktops, with design considerations of multiple components for each use case.

The architecture uses common components and standardization to reduce the complexity of implementation and design. Many of the infrastructure components used to validate the reference architecture are interchangeable, so you can use components from your vendor of choice to incorporate unique features that enhance the value of the overall solution. This VMware ThinApp reference architecture specifically focuses on integration with Active Directory, the use of Distributed File System technology for file shares, and View Composer considerations. A more comprehensive discussion of the VMware View solution is described in the *VMware View Reference Architecture*.

All desktop instances cited in this document and the accompanying validation testing were virtual machines. Given the focus on the architectural design rather than the WAN performance, we did not measure the display latency for the VMware View desktops. The metrics gathered reflect the performance of the virtualized application inside the virtual machines, not the display from the View client at the endpoint. A more comprehensive discussion of WAN design for the VMware View solution is described in the *VMware View WAN Reference Architecture*.

Reference Architecture Components

This document contains design guidance for multiple use cases and recommends specific configuration approaches. The first section reviews the basic structure of the packaging process and considerations for virtualizing applications for distribution. Subsequent sections discuss considerations specific to VMware ThinApp configurations with multiple layers of infrastructure.

ThinApp Packaging Framework

VMware ThinApp packages are created as part of the Setup Capture process. During this process, the application files and registry are combined with administrative settings that are embedded into the final package for distribution. As applications are updated and configurations are changed, the packages are rebuilt to embed the changes into the package. The Source, UAT, and Production folders, described below, provide a commonly used framework for structuring the process of application packaging.

Source Folder

The Source Folder contains the project directories created by the VMware ThinApp Setup Capture process. The project directories contain what can be considered the 'source code' of these application packages and administrative settings. Maintain these project directories in a location that is backed up regularly and has access control mechanisms that allow only administrative access.

UAT Folder

The Unit Acceptance Testing Folder provides an intermediary location to perform testing and QA procedures. To ensure quality testing, this location should represent the final end-user community as closely as possible.

Production Folder

The Production Folder is a read-only file-share made available to end users for VMware ThinApp packages that are run in streaming execution mode. End users run the virtualized applications directly from this location. Since the Production Folder is critical and user-facing, it should be monitored for availability and performance. DFS or other technologies that provide high availability are highly recommended. The VMware ThinApp Reference Architecture implements Microsoft DFS, which is available to most customers without additional cost and provides an adequate solution with common components. Figure 1 illustrates the role of each folder and basic flow of the packaging process.

Application Packaging Considerations

Before virtualizing applications with the Setup Capture process, IT organizations should determine some specifics of the scenario into which the package will be deployed. For a comprehensive discussion of these considerations and their implementation, see the *VMware ThinApp Deployment Guide* and *VMware ThinApp User's Manual*.

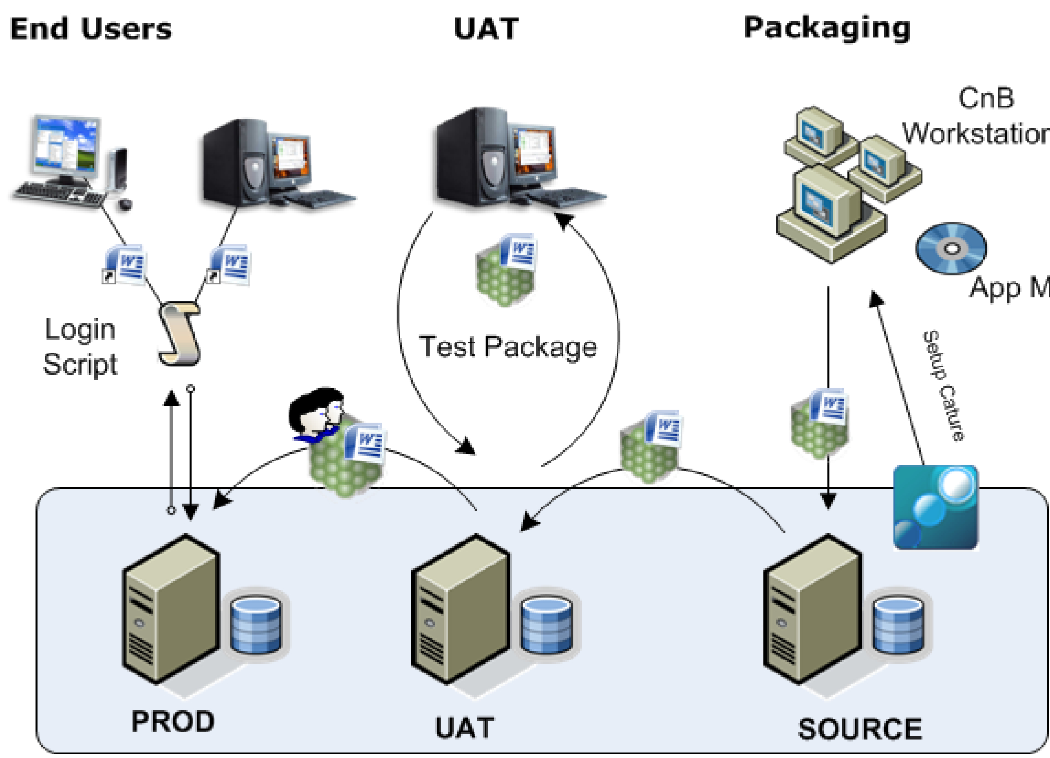

Figure 1. ThinApp Packaging Folder Framework

Determining the Execution Mode

VMware ThinApp allows IT organizations to determine whether to use *streaming* or *deployed* execution mode or to adopt a hybrid approach that lets them manage a standard set of applications centrally while distributing others in deployed mode. The same virtualized application packages can be used for either execution mode.

- *Streaming Execution* mode allows the application to be stored centrally and accessed by multiple users. This one-to-many model provides centralized deployment and update of an application *package* to multiple end users for execution via a Windows desktop shortcut. The user launches the application from the central network location where the application resides and streams data as needed while the application is in use.
- *Deployed Execution* mode distributes the virtualized application packages to the end user's system, on the local file system or on a USB device. In this distributed model, each client device receives and executes the *package* locally and therefore can run the application regardless of network connectivity. End-user devices that are occasionally or always offline require deployed execution mode.

Update Mechanism

There are two primary methods for updating virtualized applications:

- The Side-by-Side (aka Integer) update method for application packages can be used either for streaming or for deployed execution mode. There is no requirement for application downtime. This method places the new application package in the same directory as the original application package and increments the filename extension to an integer number. Subsequent updates can be placed in the directories with extensions .2, .3, etc. For example, placing Mozilla Firefox.1 in the same directory with Mozilla Firefox would automatically update users to execute the Mozilla Firefox.1 package.

- Application Sync is a functionality that can be embedded into VMware ThinApp application packages. Application Sync provides updates to unmanaged machines that connect over networks that are known to have some degree of latency. AppSync provides a mechanism for a differential transfer over HTTP to the endpoint, so it is only used for application packages in deployed execution mode. Application Sync can optionally utilize a UNC point to retrieve updates from locations within a corporate environment. When an application starts, Application Sync can query a Web server or file share to see if an updated version of the package is available.

Application Dependencies

Application Link is a VMware ThinApp feature that allows the administrator to build relationships between packages, creating modular packages that link together instead of larger packages that are more difficult to distribute and update. Administrators can use Application Link to create relationships between local or remote application packages containing components or dependencies. Administrator can configure the originating application package to look first in a local directory and then in a remote directory if the required or optional component cannot be found. To provide a logical link to the remote directory you can use a domain-based DFS namespace. Further detail is provided in the Storage Layer Design Considerations section.

Application Link provides the following functions:

- Links runtime components, such as .NET, JRE, or ODBC drivers, with dependent applications. For example, you can link .NET to an application even if the local machine for the application does not allow for the installation of .NET or already has a different version of .NET.
- Allows administrators to deploy application-specific components and plug-ins separately from the base application. For example, you might separate Adobe Flash Player or Adobe Reader from a base Firefox application and link the components.

Active Directory Integration

VMware ThinApp supports native Active Directory integration to manage configuration of user data locations, dynamic application changes, registration of application packages, and access control.

Active Directory Group Policy Management

Folder Redirection group policies can be used to redirect storage of a user's application settings and user data. The locations can be redirected to a user's roaming profile or home directory or View Composer User Data Disk. **Note: Folders within virtualized applications cannot be redirected to UNC-style path locations.** (see KB article 1013933)

- Application Data
  This location houses the *sandbox*, located in the user's profile by default, to store runtime changes to the virtualized applications registry, folders, and files. This location houses the information that gives users a persistent application experience for thing like toolbar settings and customized options within the applications.
- My Documents
  Redirection of this location to a user's home drive or other configured location ensures that user data is well managed. Virtualized and natively installed applications default to saving documents in this location.

Active Directory-Based Access Control

Administrators can use Active Directory groups to control access to virtualized applications. The *setup capture* process allows administrators to enumerate and assign Active Directory groups. Those assignments are embedded into the package during the build process. This process populates the PermittedGroups parameter in the Package.ini and restricts usage of a package to a specific set of Active Directory users. Administrators can optionally customize the error message to users if they are not allowed to launch the application. For a desktop that is offline, the PermittedGroups function utilizes cached credentials to determine whether the user has permission to launch the application.

Active Directory Script- based Application Registration

Registration of virtualized applications to end users creates shortcuts on the desktop, file-type associations, and entries in the Add/Remove programs applet of the Control Panel. The Thinreg utility helps to automate the registration process. Thinreg.exe can be run from a login script, a local script, or a command line. Since the registration process can enumerate which users have access to application packages, the ThinReg process can be run against an entire directory of application packages; however, it only registers the applications to which the user is entitled. Two common methods of implementation are described briefly below.

- Login Script based
  Implementation of the Thinreg executable can be incorporated into an existing login script with standard methods such as .bat, WSH, KIX, or vbScript. See example below;
  %logonserver%\netlogon\thinreg.exe \\company.com\applications*.exe /Q
- Local Script via Registry Run Key
  IT organizations can choose to implement the application registration process locally on the workstations instead of incorporating it into the login script. The Run key of the registry can call the Thinreg.exe file to perform the necessary functions on login. Placing the Thinreg.exe in the Windows directory simplifies the execution of the script and requires nothing more than the executable to function.

Storage Layer Design Considerations

When users launch applications in streaming mode, they must have read access to the storage location of the application package that the Windows shortcut references. The storage location that hosts the Prod folder should be made highly available so that downtime of a host or storage device does not impact the environment. Since end users only read the package from the central location, the disk usage pattern is entirely read-oriented, so the storage should be configured for optimal read performance. The use of any number of SAN, DFS, or file replication technologies is sufficient to make the file share highly available and redundant.

Benefits of DFS

Many IT organizations utilize Microsoft DFS or Microsoft Windows Server 2003 or 2008 to host file services. DFS offers substantial benefits in terms of lower administrative overhead for folder management, built-in replication, and the convenience of centrally administered DFS domain namespaces. For detailed Microsoft DFS configuration and implementation guidance see:
http://www.microsoft.com/windowsserver2003/technologies/storage/dfs/default.mspx

Utilizing DFS with VMware ThinApp

- Domain-based Namespaces
  Administrators can leverage the capabilities of DFS to create a single logical reference to a folder that houses virtualized applications, for example, \\companycom\Applications. Use domain-based namespaces for DFS links, which are placed into the Active Directory schema. Domain-based namespaces are then natively replicated throughout the Active Directory infrastructure, providing fault tolerance for the DFS namespace.
- DFS Targets and Replication
  DFS Namespaces can be configured to point to multiple locations, called Targets, which house the file shares. Multiple targets provide the needed redundancy for enterprise deployments. Additionally, customers can configure replication to branch offices and determine failover links to remote targets based on Active Directory site design. The DFS targets will be read-only shares so configure the replication links appropriately.
- DFS Namespaces for AppLink Locations
  When administrators package applications they can specify UNC based locations for required or optional application dependencies. Administrators can configure the originating application package to look first in a local directory and then in a remote directory if the required or optional component cannot be found. The use of a DFS domain namespace such as \\company.com\Applinks could be utilized in the packaging process to direct users to the nearerst DFS target for the required component. An example line from a package.ini follows, the first location is local, the second refers to the DFS link:

  ```
  RequiredAppLinks=c:\AppLinks\*.exe;\\company.com\AppLinks\*.exe
  ```

Network Layer Design Considerations

The path through the network between the client device and the Applications folder should be highly available and robust. Consider both aggregate bandwidth and latency when implementing file share with applications launched in streaming mode. These applications use the standard SMB protocol to transfer the blocks of data to the end-user system for execution. The amount of network traffic varies based on the application and usage pattern, consider only low-latency LAN environments for streaming execution mode. See the *VMware ThinApp Streaming Information Guide*.

Using Compression with Streaming Execution Mode

VMware ThinApp provides the option of using compression to reduce the storage footprint of the application packages. Applications launched in streaming mode request only the files and registry necessary to perform the specific application function, which means that the package is never streamed in its entirety. With compression enabled, the quantity of data transferred across the network is greatly reduced (by about 50%), but there is slightly higher memory usage for an application and slightly longer launch times when larger blocks sizes are used. This presents the question of whether it is more beneficial to reduce the amount of network utilization or ensure optimal end-user performance.

VMware View Integration

Desktop Persistence

Persistent desktops are assigned to individual users and remain so assigned until an administrator makes a change. Properly entitled users may also check their desktops out for offline use. Persistent desktops are best for users who want to customize their desktops by installing additional applications and storing local data. Non-persistent desktops are allocated to users temporarily and used only for the current session. Once the user has logged off a non-persistent desktop, it goes back into the desktop pool where it originated and becomes available for the next user. Non-persistent desktops should be used where each user session requires a clean machine or in highly controlled environments where there is no need to store customization on the virtual desktop.

VMware ThinApp application packages can be used with either persistent or non-persistent desktops. Administrators can utilize folder redirection for the application sandbox, which lets end users save their customized application settings, such as toolbars and options settings, to a static location. This allows for persistent application settings regardless of whether the desktop is persistent or non-persistent.

Administrators can also choose whether to deploy virtualized applications into the desktop instance or provide users access to the application on a file share through the automated ThinReg registration process. Deployment into the desktop instance provides more predictable performance because all execution is local, but it requires application packages to be updated on the desktops individually and packages will have a storage footprint in the VM. Providing applications via streaming mode makes greater demands on the network and can affect performance, but it allows applications to be updated with greater efficiency due to the one-to-many model of streaming mode, and there is no disk footprint within the VM.

View Composer Considerations

VMware View Composer provides the capability to create desktop images rapidly from a parent, or standard, virtual machine image. In addition, VMware View Composer reduces the total amount of storage required to deploy virtual desktop images (see Figure 2). By leveraging linked clones, it introduces a streamlined process for upgrading patches across multiple desktops by simply applying the patch to the parent image and recomposing the linked clones.

VMware View Composer also provides the ability to separate user data and profile settings, allowing software updates and patches to be applied to the parent image and inherited by the linked clones. After a linked clone has been updated, the user's personal settings from the user data disk are also applied to the updated image.

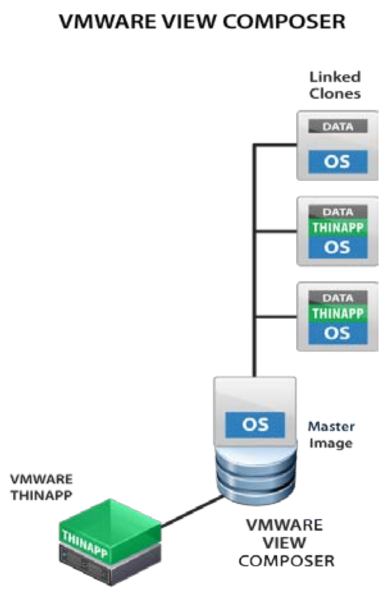

Figure 2. Parent Image with Linked Clones

Integration of VMware View Composer and ThinApp

There are two aspects of the integration between VMware View Composer and ThinApp, the intersection of updates for the operating system and applications, and the use of the User Data Disk for storage of the Application sandbox.

- There are benefits to deploying applications packages into the View Composer base image. View Composer provides a streamlined mechanism to refresh and recompose the base operating system, which can also be used as an update vehicle for ThinApp application packages. This operation would allow for a mass update of application packages along with a desktop recompose operation. The only caveat of this approach is that the administrator will be combining operating system recompose events with application updates. Administrators should compare the frequency of application updates with that of OS updates and determine the appropriate location for ThinApp packages. A compromise can be achieved by using streaming mode for frequently updated applications while deploying less frequently updated applications into the operating system image.
- ThinApp packages can also be located on the User Data Disks, however this requires another means to update those packages such as a software distribution solution and this would need to occur for each individual user data disk.
- When used in conjunction with View Composer, administrators have the option of utilizing User Data Disks to logically separate application and user storage from the operating system. By default, VMware ThinApp places the application sandbox in whatever location the %AppData% variable resolves to in the operating system. User Data Disks automatically receive the dynamic changes of the applications and user-specific settings stored in the sandbox and maintain these through a refresh of the operating system. Administrators can choose to redirect My Documents to a storage location or allow that user data to also be housed in the User Data Disk.

Reference Architecture Design

The reference architecture design provides use cases which are designed to represent a number of common scenarios for configuration. The use cases outline the different of configuration items in a table format and then illustrate the environment in a diagram.

Use Case Descriptions and Diagrams

For the VMware ThinApp Reference Architecture, we provide three use cases that are representative of many customer scenarios:

- Use Case 1— Mobile Users
- Use Case 2 — Remote Users
- Use Case 3 — Corporate Users

Use Case 1: Mobile Users Configuration Table

User Type	Mobile Users
Link Type	Public Internet, Corp VPN or Offline
Desktop Configuration	Corporate Laptop, Home PC, Offline View Persistent Full Clone (Experimental), VM running on the Client Virtualized Platform (Future)
Application Registration	Registry Run Key or User-initiated script on USB key or local device (Home PC's)

Method	
Folder Redirection	None
Deployment Method	Deployed Mode (USB, Local File System)
Update Method	AppSync
Compression	Not Enabled

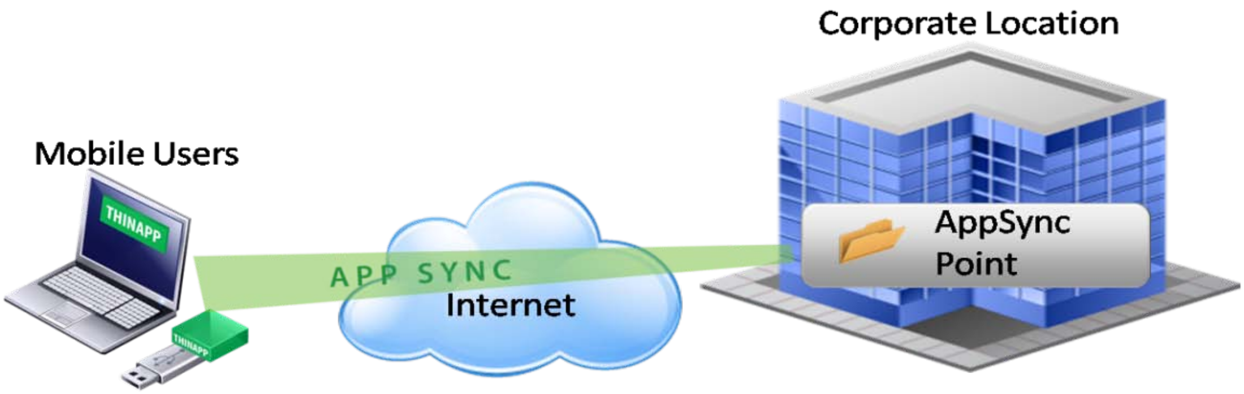

Figure 3. Mobile Users (Use Case 1)

Use Case 2: Remote Users Configuration Table

User Type	Remote users connecting to VMware View desktops
Link Type	Corporate VPN or WAN
Desktop Configuration	View Non-persistent Desktop without User Data Disk
Application Registration Method	Login Script
Folder Redirection	Application Data is redirected to users roaming profile or home drive My Documents is redirected to users home drives
Deployment Method	Deployed Mode for MS Office Suite Streaming Mode for LOB and ancillary applications
Update Method	Side-by-Side for Streaming Mode Applications View Composer for Deployed Mode Applications
Compression	Enabled for Applications using Streaming mode

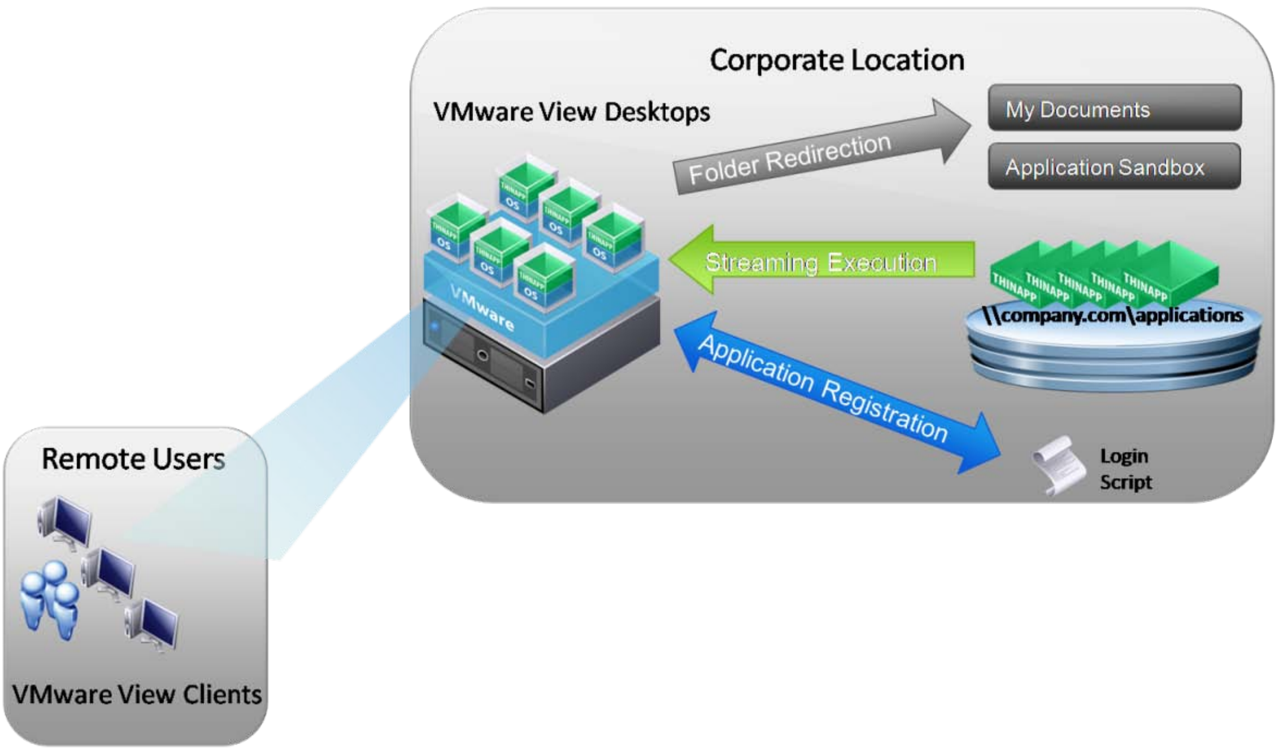

Figure 4. Remote Users (Use Case 2)

Use Case 3: Corporate Users Configuration Table

User Type	Corporate Knowledge Workers with PCs and/or VMware View Desktops
Link Type	Corporate LAN or WAN
Desktop Configuration	Persistent Desktop with User Data Disk or Physical PC
Application Registration Method	Login Script
Folder Redirection	View Based Desktops Application Data automatically redirected via View Composer User Data Disks My Documents automatically redirected via View Composer User Data Disks Physical PC's Application Data redirected to users home drives My Documents redirected to users home drives
Deployment Method	Deployed Mode for MS Office Suite Streaming Execution Mode for LOB and ancillary applications
Update Method	Side-by-Side for Streaming Mode Applications View Composer for Deployed Mode Applications
Compression	Enabled for Applications using Streaming mode

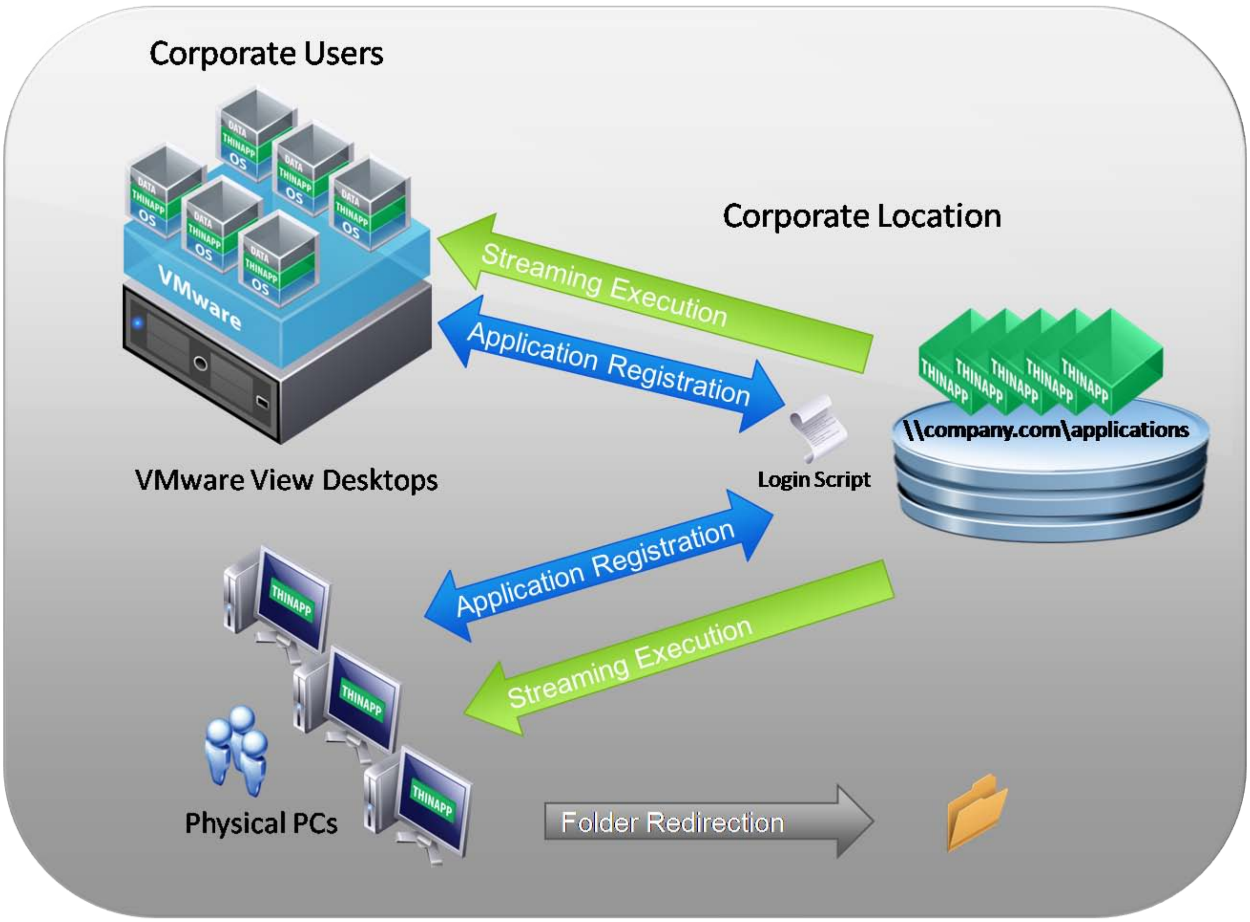

Figure 5. Corporate Users (Use Case 3)

VMware ThinApp Test Bed

Virtual Infrastructure

The virtual infrastructure used for this reference architecture included one chassis of 16 blades to host the virtual desktops and two blades in a separate chassis for the infrastructure components.

Physical Server Configuration

The design of the virtual desktops uses the building block approach described in the *VMware View Reference Architecture*. The virtual desktops are hosted by two 8-node VMware ESX 3.5 U2 clusters. Each 8-node cluster is designed to host 500 virtual desktops. Both clusters were configured as HA clusters and managed by a single VMware Virtual Center 2.5 U4 server.

A separate blade chassis was used to host the common infrastructure components needed for an enterprise desktop environment, such as Active Directory, DNS, DHCP, DFS, and SQL. This chassis also hosted VMware View Manager and vCenter with View Composer. The VMware vCenter server database and View Composer database were hosted on a single Microsoft SQL 2005 server. Each desktop infrastructure service was implemented as a virtual machine running Windows 2003 SP3, with the exception of the DFS Servers, which ran Windows 2003 R2.

See the following diagram and tables for detailed information regarding the testing environment.

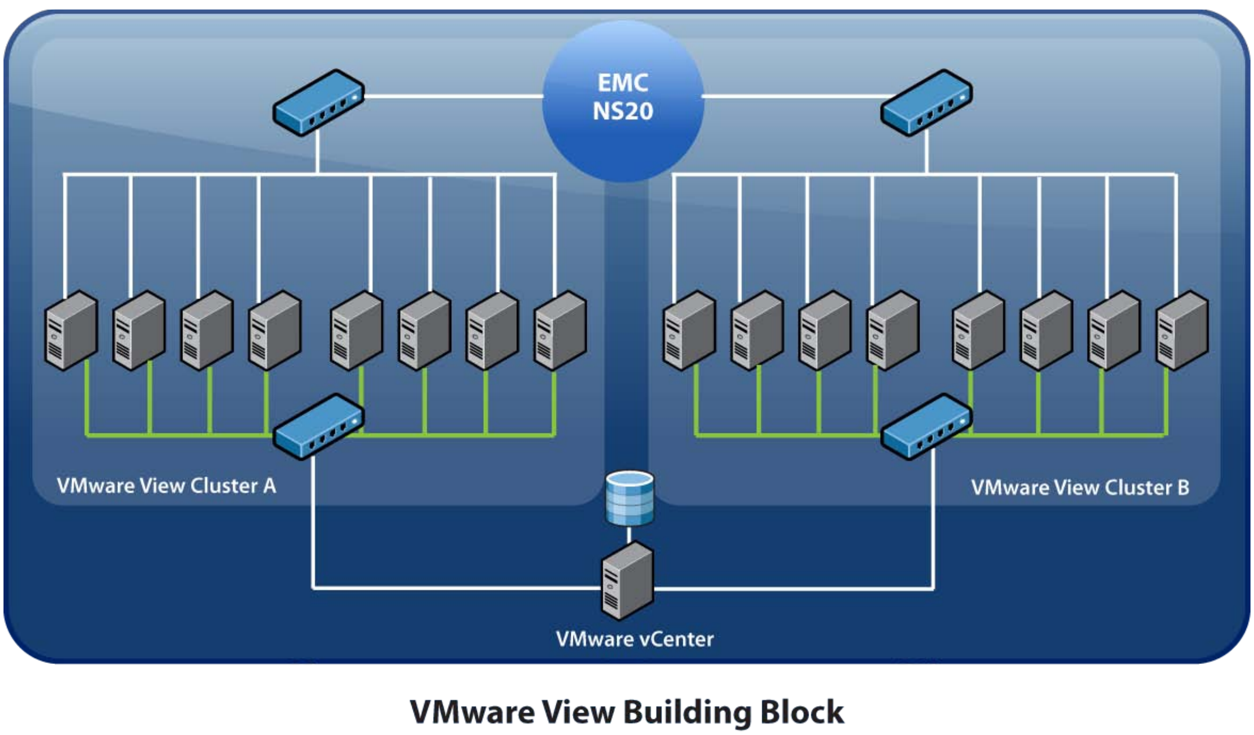

Figure 6. Physical Testing Environment based on VMware View Building Block

Infrastructure Servers	
QTY	**Description**
1	16 Slot Blade Chassis

4	Blade Servers - ESX 3.5 Update 2 4 – Infrastructure Services AD, DNS, DHCP, DFS
2	Quad Core 2.66 GHz Processors
32GB	RAM
1	x 56GB SAS Drive
4	Broadcom Gigabit Ethernet Adapters
4	4 Port Gigabit Uplink Modules
1	VMware vCenter Virtual Machine with View Composer Windows 2003 Server SP3 2 – vCPU 4GB – RAM 20GB Virtual Disk
1	VMware View Manager Windows 2003 Server SP3 2 – vCPU 4GB – RAM 20GB Virtual Disk
1	Microsoft SQL 2005 Server Windows 2003 Server SP3 2 – vCPU 4GB – RAM 20GB Virtual Disk
1	Microsoft Active Directory, DNS, DHCP Windows 2003 Server SP3 2 – vCPU 4GB – RAM 20GB Virtual Disk
2	Microsoft DFS Targets Windows 2003 Server R2 2 – vCPU 4GB – RAM 20GB Virtual Disk
VMware View Desktop Building Block A	
QTY	**Description**
1	16 Slot Chassis – Cluster A/B
8	Blade Servers - ESX 3.5 Update 2
2	Quad Core 2.66 GHz Processors
32GB	RAM
1	x 56GB SAS Drive
6	Broadcom Gigabit Ethernet Adapters
6	4 Port Gigabit Uplink Modules

VMware View Desktop Building Block B	
QTY	**Description**
8	Blade Servers - ESX 3.5 Update 2
2	Quad Core 2.66 GHz Processors
32GB	RAM
1	56GB SAS Drive
6	Broadcom Gigabit Ethernet Adapters
6	4 Port Gigabit Uplink Modules

NOTE: Two Mirrored 56GB drives are recommended for production.

Physical Network Details

VMware View Pod Core Networking Components	
QTY	**Description**
1	Modular Core Networking Switch
1	10 Gigabit Ethernet Modules
1	Load Balancing Module
Building Block Network Components	
1	48 Port Network Switch
VLAN Configuration	
VLAN ID	**Description**
16	VMware View Desktops – Infrastructure -802.11q Tagged
20	Management – 802.11q Tagged
23	Storage – iSCSI – 802.11q Tagged
24	vMotion – 802.11q Tagged

Virtual Desktop Configuration

Virtual desktops were deployed as persistent linked clones with and without user data disks, based on the use case.

- One virtual CPU
- 1 GB of RAM
- 8GB hard disk
- Microsoft Windows XP guest operating system with Service Pack 3

Application Configuration

- The primary virtualized application set included MS Word2007, MS PowerPoint2007, MS Excel2007, Internet Explorer7, and Adobe Reader9.
- The virtualized application set was used in deployed mode and streaming mode via a drive mapping in the login script. Thinreg was used to register applications locally and remotely.

Storage Configuration

Validation testing for the storage layer used the following storage platforms:

- 500 Virtual Desktops running on iSCSI storage hosted by an EMC NS20
- 2 Virtual Windows Server 2003 R2 servers hosting DFS Targets for Domain-based namespaces with their virtual disks located on an EMC CLARiiON CX4-240 Fibre Channel storage array

Workload Description

User Simulation Workload

For validation testing, virtual desktops were equipped to run a workload that simulates typical user behavior with an application set commonly found and used across a broad array of desktop environments. The workload, which is described below, has a set of randomly executed functions that perform operations on a variety of applications. Several other factors can be implemented to increase the load or adjust the user behavior — for example, configuration options include changing the number of words per minute that are typed and the delay between the times applications are launched.

The workload configuration used for this validation included Microsoft Word, Excel, PowerPoint, Internet Explorer, and Adobe Acrobat. The workload controller opened multiple applications at the same time and minimized and maximized their windows as the workload progressed, randomly switching among applications. Individual application operations that the controller performed randomly included:

- Microsoft Word — Open, minimize, and close the application; write random words and numbers; save modifications.
- Microsoft Excel — Open, minimize, and close the application; write random numbers; insert and delete columns and rows; copy and paste formulas; save modifications.
- Microsoft PowerPoint — Open, minimize, and close the application; conduct a slide show.
- Adobe Acrobat Reader — Open, minimize, and close the application; browse pages in a PDF document.
- Internet Explorer — Open, minimize, and close the application; browse a page.

Based on the think time and words per minute used for this validation, this workload can be compared to that of a high-end task worker or lower-end knowledge worker.

Unit Test Workload

A second workload, 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. This workload was used to derive precise measurements of the effects of compression and latency when running virtualized applications in streaming execution mode.

Validation Testing

The reference architecture validation testing results first measure the effect of compression on network utilization for a single repeatable workload, and second, quantify storage and network utilization for a given number of users running a random workload.

Summary of Unit Testing Procedure

The sections below outline the results for the unit testing of the effect of compression and latency for virtualized applications executed in streaming mode. Two virtual desktops were configured to run the OfficeBench workload, with three iterations utilizing virtualized applications running from a DFS share in streaming execution mode. Thinreg was used to register the applications from the remote drive. No native applications were installed in the VMs.

Unit Testing Results

The effect of VMware ThinApp native compression on applications running in streaming execution mode is shown in the following table:

Test	OfficeBench Completion Time (Seconds)	Total Network Payload (MB)	Application Package Size on File Share (MB)
Uncompressed Iteration 1	50.16	202.146	1,162,999
Uncompressed Iteration 2	49.75	201.591	"
Uncompressed Iteration 3	52.44	200.754	"
Compressed Iteration 1	53.38	112.765	858,943
Compressed Iteration 2	51.61	106.523	"
Compressed Iteration 3	54.33	112.311	"

The results of the testing showed negligible differences in OfficeBench completion time, CPU and Memory Utilization. However, the network transfer showed substantial improvement from compression. The average decrease in the network payload between uncompressed and compressed was 45%. The desktop was rebooted between the tests to empty the disk cache and ensure accurate results. Figure 7 graphs the pattern of network utilization.

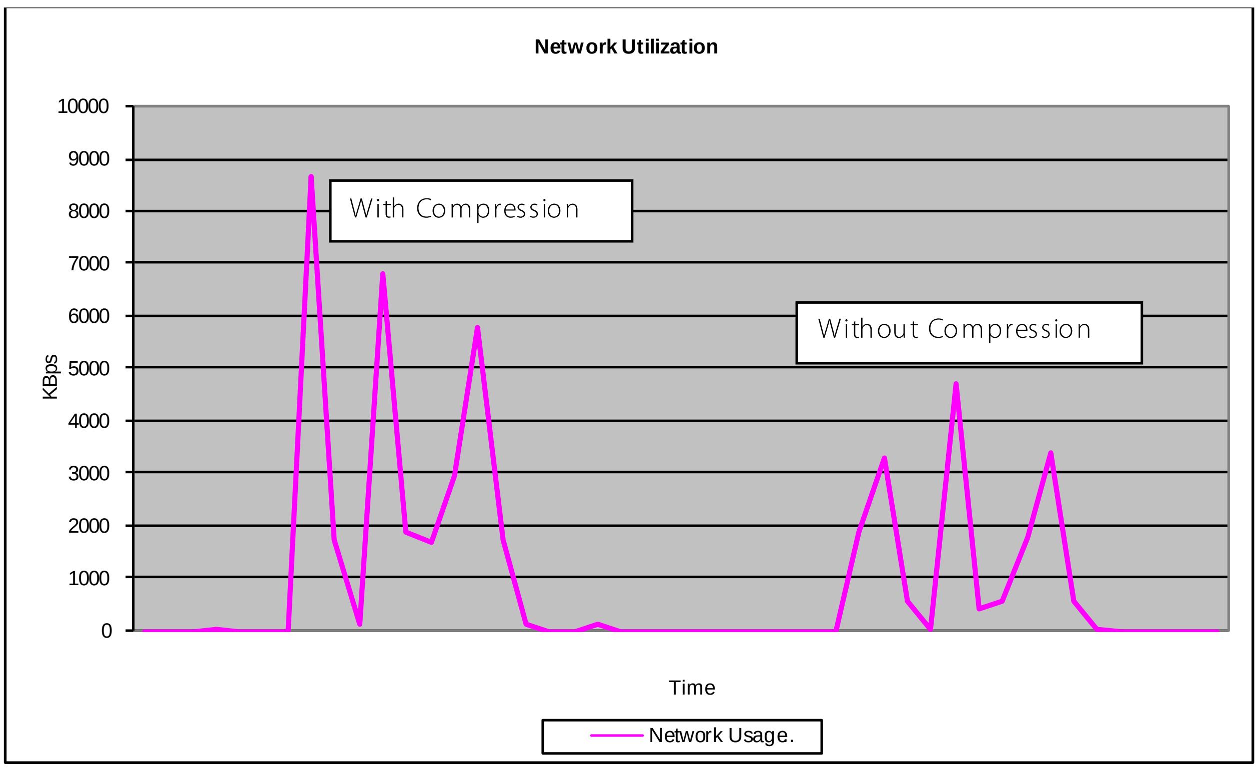

Figure 7. Network Utilization for Compressed and Uncompressed Packages

Summary of Scalability Testing Procedure

The validation testing for scalability made use of the user simulation workload, which provides a random workload to simulate real users. Five hundred virtual desktops were configured to use streaming execution mode and run a series of three iterations. The applications were launched from a pair of Microsoft DFS server's which were the DFS targets for the domain namespace \\company.com\applications. Both DFS servers were located on the same physical ESX host.

Scalability Testing Results

Disk Utilization

Figure 8 illustrates the disk utilization for each of the DFS servers.

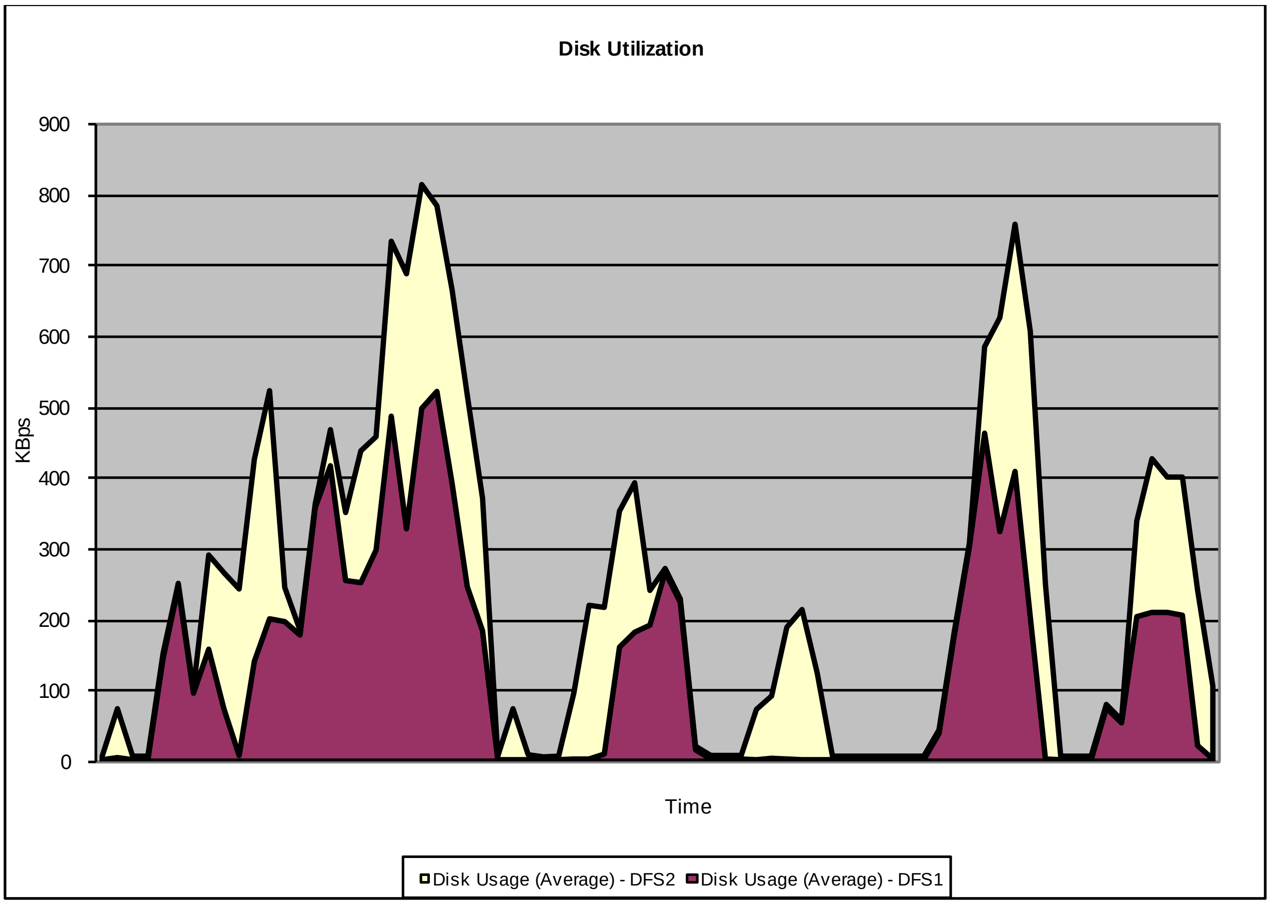

Figure 8. Disk Utilization for Microsoft DFS Virtual Machines

Network Utilization

Figure 9 illustrates the network utilization for the VMware ESX host, which contained both DFS servers.

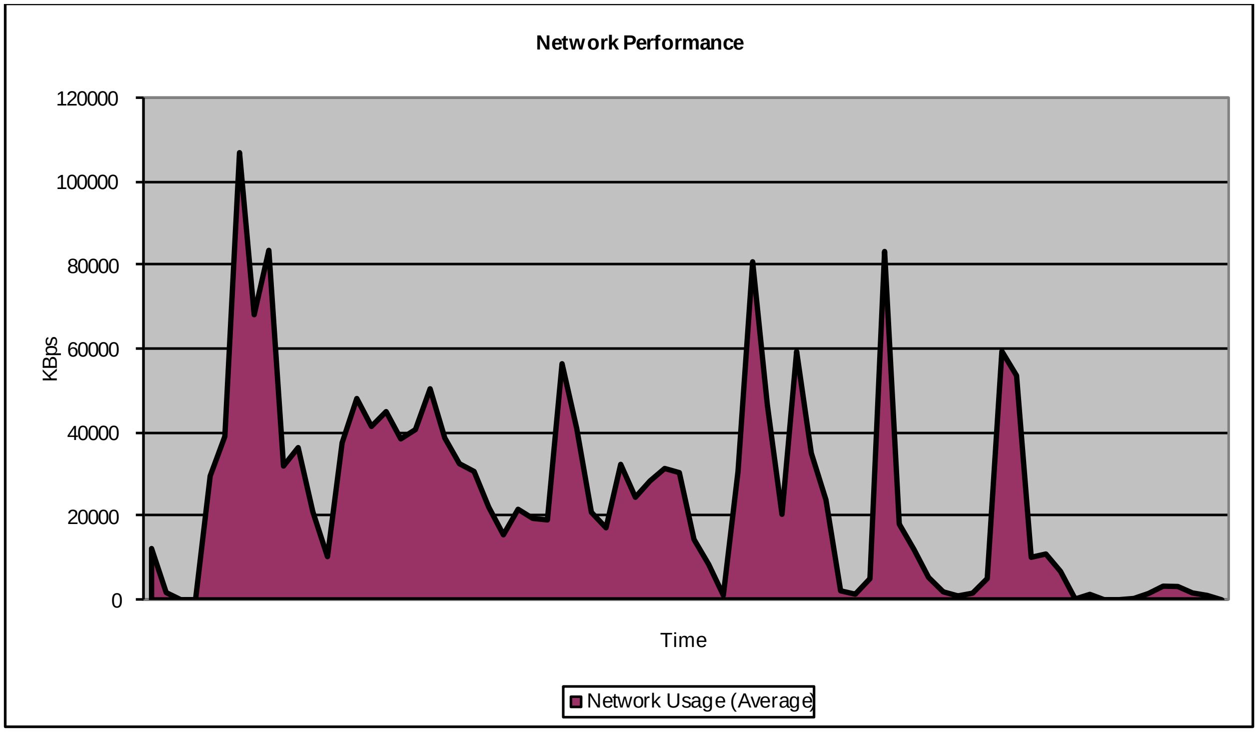

Figure 9. Network Utilization for ESX Host with Microsoft DFS Virtual Machines

CPU Utilization

Figure 10 illustrates the CPU utilization for the Microsoft DFS virtual machines.

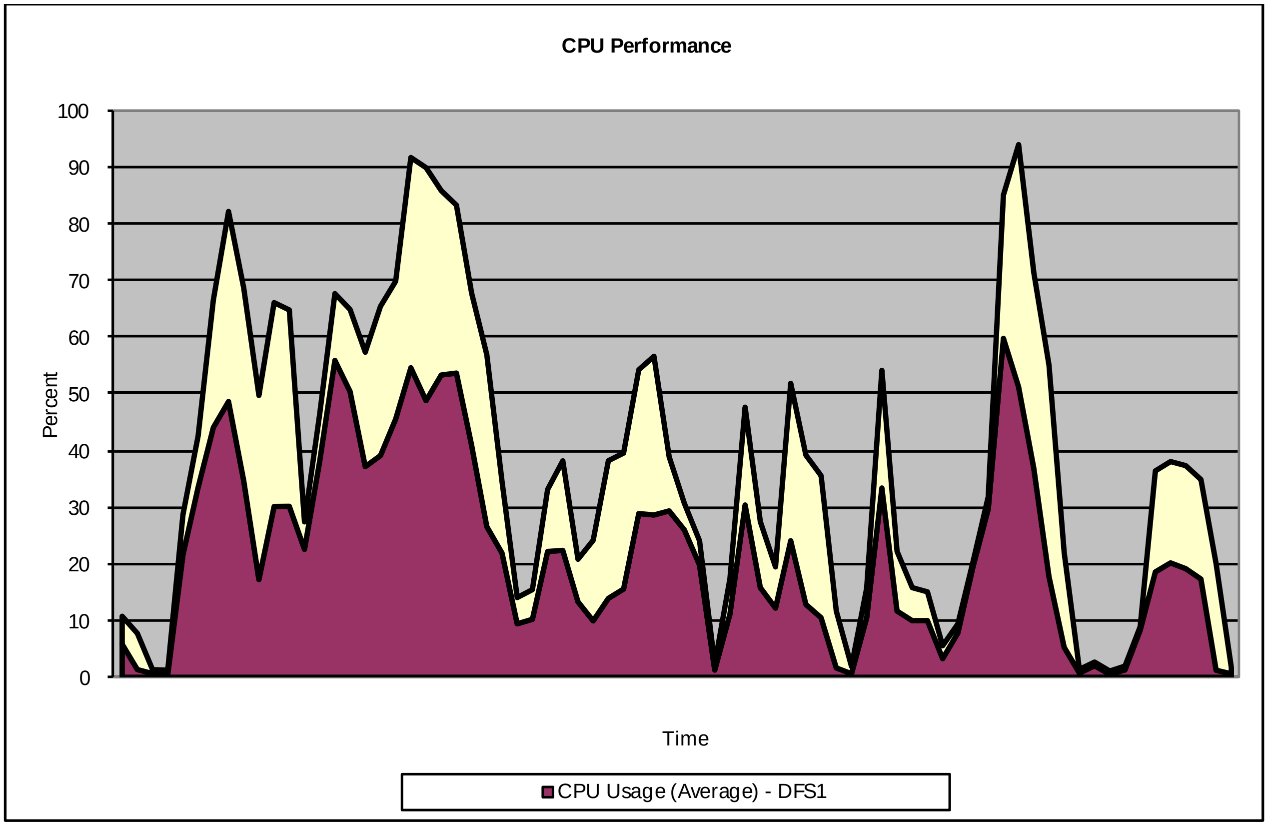

Figure 10. CPU Utilization for Microsoft DFS Virtual Machines

Conclusion

VMware ThinApp greatly simplifies the process of application distribution and management for both physical and virtual desktops. Customers can leverage agentless application virtualization to reduce operational expense incurred by installation reboots, application conflicts, and dependencies. By utilizing the design considerations and use case configurations provided in this document, customers can confidently design and implement VMware ThinApp in their own environments.

The design considerations section provides specific guidance for Active Directory integration, View Composer integration, and the use of DFS to efficiently manage file shares for virtualized applications and their dependencies. The validation results help characterize the network and storage load for a group of 500 virtual machines when utilizing streaming mode for VMware ThinApp application packages. The unit test scenarios show the difference in network load when compression is used for application packages. The following key recommendations emerge from this validation:

- Utilize a structured process and standardized folder layout for application packaging.
- Utilize Active Directory group policy to redirect Application Data and My Document to the appropriate locations depending on the use case.
- Utilize ThinReg through a script based mechanism to automate the registration of ThinApp application packages.
- Deploy virtualized applications via a standard process that relies on logical DFS links that provide redundancy and manageability.
- Determine which applications to run in streaming mode, based on bandwidth needs and application density goals. Utilize compression to reduce the network payload.

About the Author

Aaron Black is a Senior Technical Marketing Manager at VMware, focusing primarily on developing technical content to aid in the evaluation and implementation of VMware ThinApp technology. Aaron's background includes roles as a systems engineer and solutions consultant in the Technical Services organization. Before joining VMware, he worked as a systems engineer with Citrix Systems, lead a technical corporate IT team at Sprint, and designed solutions for customers of Choice Solutions, a platinum reseller of VMware products.

Acknowledgements

The author would like to acknowledge John Dodge, Radhakrishnan Manga, Pak-Shun Lei, Fred Schimscheimer, and Mason Uyeda for their contributions to this reference architecture.

Resources

- VMware ThinApp Deployment Guide
  http://www.vmware.com/resources/techresources/1098
- VMware ThinApp Streaming Information Guide
  http://www.vmware.com/resources/techresources/10027
- *VMware View Manager Administration Guide*
  http://www.vmware.com/pdf/view31_manual.pdf

- *VMware View Composer Design Considerations*
  http://www.vmware.com/files/pdf/view_composer_design_considerations_bp.pdf
- *VMware View Reference Architecture*
  http://www.vmware.com/files/pdf/resources/vmware-view-reference-architecture.pdf

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com