



P r o f e s s i o n a l E x p e r t i s e D i s t i l l e d

VMware View 5 Desktop Virtualization Solutions

A complete guide to planning and designing solutions based on VMware View 5

*Foreword by
Simon Bramfitt, Founder, Entelechy Associates*

Jason Langone

Andre Leibovici

[PACKT] enterprise 
PUBLISHING professional expertise distilled

VMware View 5 Desktop Virtualization Solutions

A complete guide to planning and designing solutions
based on VMware View 5

Jason Langone

Andre Leibovici



BIRMINGHAM - MUMBAI

VMware View 5 Desktop Virtualization Solutions

Copyright © 2012 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: June 2012

Production Reference: 1010612

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84968-112-4

www.packtpub.com

Cover Image by Sandeep Babu (sandyjb@gmail.com)

Credits

Authors

Jason Langone (@langonej)
Andre Leibovici (@andreleibovici)

Reviewer

Raymond van't Hag

Acquisition Editor

Rukshana Khambatta

Lead Technical Editor

Rukshana Khambatta

Technical Editor

Devdutt Kulkarni

Copy Editor

Laxmi Subramanian

Project Coordinator

Leena Purkait

Proofreader

Kelly Hutchinson

Indexer

Rekha Nair

Graphics

Valentina D'Silva
Manu Joseph

Production Coordinator

Arvindkumar Gupta

Cover Work

Arvindkumar Gupta

Foreword

On August 12, 1981, IBM released the IBM Personal Computer. It was a transformative event; the one that in time would far exceed even the most optimistic projections of its future potential. It changed forever how the computer would be viewed, making it truly "Personal".

Much has changed since that day; many of these changes are immediately visible. The computer's user interface has transformed beyond all recognition, from the blocky text of the DOS command prompt to rich windowed interfaces driven by a mouse or touchpad. PCs are faster, smaller, and cheaper than ever before and are capable of doing things that were beyond the imagination of all but the most far reaching science-fiction authors. Other changes are more subtle; the PC is now both ubiquitous and ever-connected. Its transformation in size and appearance has seen it acquire new names to better describe its new form. Becoming the Portable PC first and later the laptop and netbook, and most recently the tablet and smartphone; hiding its nature in ever smaller packages with new interfaces designed to be controlled by gestures and voice. As a result, some would even have it that we are entering a post-PC era.

It is perhaps too soon to say that we have left the PC behind. For all these changes, one thing remains unaltered — today's personal computing experience is still centered on the device. Applications are for the most part installed locally and the decision to walk from device to device or carry one from place to place is based more on the device's size than the user's need for mobility. Either way, the only method to ensure that it is possible to provide service is to rigorously enforce standardization, making sure every PC has all the applications preinstalled on the off chance that someone will need them. This model has worked, after a fashion, almost unchanged for the last 31 years, but of late it is starting to show its age. Now, we need to consider a world shaped by new, deeply destabilizing forces.

The launch of the Apple iPad in 2010 brought with it "Bring Your Own Device" programs and was the stimulus that pushed the "Consumerization of IT" into the limelight, sending shockwaves through what was until recently a smooth running, well-understood service. We are rapidly entering a world where an IT administrator doesn't know from one day to the next how many users he has to support or even where in the world they might be located. Where managing user experience does not mean providing a standard suite of applications across 10,000 identical PCs, each running identically configured copies of Windows, but supporting and insuring a comprehensive user experience (not just the user profile and home directory, but the user's entire working environment, their data, and applications) and making it available in its entirety at any place, any time, and on any device. Welcome to desktop virtualization.

Desktop virtualization is a deceptive term; everybody understands that at its core, server virtualization is all about being able to consolidate server workloads onto fewer physical servers. Desktop virtualization is more subtle and far reaching. It's not about consolidating desktop workloads into fewer physical servers, although, frequently this is part of the picture. Rather, it's about being able to orchestrate the creation of personalized working environments (that is, applications, data, and user profile) and enabling access to them in the manner most appropriate to each individual user's computing and communications environment while at the same time ensuring full compliance with organizational governance, risk, and compliance policies.

Simon Bramfitt

Founder

Entelechy Associates

About the Authors

Jason Langone (@langonej) has been involved in VMware solutions for over seven years and has designed and implemented solutions around the world. Langone has a long history of designing VMware-based tools (for example, V-Moses and ESX Recovery Center), as well as designing robust solutions for enterprise customers. Langone spoke at VMWorld 2006 and won a VMware Vanguard Award in 2007 for Best DR Solution. He is a **VMware Certified Design Expert (VCDX)**, a VMware vExpert, and a **VMware Certified Professional (VCP)**. In addition, Langone has been involved in some of the largest VDI deployments from the Middle East to deployable solutions in the US.

Langone maintains a blog dedicated to virtualization and cloud computing at www.ThinkVIRT.com.

I'd like to acknowledge the following individuals whom I have had the pleasure of working with or who contributed to this book by proofreading, editing, mentoring, commenting, and discussing its content. In no particular order, they are Dwayne Lessner (@dlink7), Simon Bramfitt (@simonbramfitt), Elvedin Trnjanin (@etrnjanin), Andy Murphy (@amurph182), Jordan Harding, Pam Takahama, Tyler Rohrer (@t_rex_vdi), Steve Kaplan (@roidude), and the SPSS team at VMware Federal.

I'd also thank VMware for being the catalyst to many great professional relationships and friendships over the last seven years.

Andre Leibovici (@andreleibovici) is a leading expert in the current area of virtualization and End User Computing and maintains an award-winning and world-recognized blog. For the last 10 years, his passion and dedication around virtualization and End User Computing has helped many organizations while working for VMware Professional Services, EMC Virtualization Team (vSpecialists), and through creating professional blogging resources. His expertise is backed by more than 20 years industry experience managing IT infrastructures for large organizations.

Andre's blog <http://myvirtualcloud.net> is recognized as one of the industry leading technical VDI blogs with more than 1.5 million views every month. Based on his field experience, he developed a number of free tools to help beginners and advanced architects to appropriately size and architect VDI solutions. Those tools include the VMware View Online Calculator, the XenDesktop Online Calculator, and the Display Protocol Online Calculator.

His passion for End User Computing led him to find the APAC Virtualization Podcast and speak at conferences such as the Brazil vForum 2011, Las Vegas VMworld 2011, and the Sydney vForum 2010. Due to his creativity and accomplishments, he received the VMware Virtual Desktop Ingenuity Award 2009 and was recognized as vExpert recipient award for two consecutive years.

Degree qualified, Andre also holds VCP 5, VCAP4-DCA, VCAP4-DCD, VCP4-DT, ITIL V3, EMCCA, EMCDCA, and MCSE certifications. He is currently helping to shape the future of End User Computing by working at VMware as an architect in the Office of the CTO and enjoying his work.

About the Reviewer

Raymond van't Hag has been working for VMware for almost five years and currently holds the role of Sr. Specialist Systems Engineer End User Computing in the Netherlands. Before VMware, he worked for companies such as Dell, Symantec, and IBM. Today he is responsible for supporting larger VMware and ThinApp projects, educating VMware Partners and evangelizing VMware End User Computing strategy via social media, and especially his own blog <http://bright-streams.com>.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Instant Updates on New Packt Books

Get notified! Find out when new books are published by following [@PacktEnterprise](#) on Twitter, or the *Packt Enterprise* Facebook page.

*I would like to thank the support of my beautiful wife and daughter, Whitney and Liliana,
without whom this book would never have been finished.*

-Jason Langone

*For my dear and loving wife Rebecca
...and to Jason Langone, for allowing me to be part of his adventure.*

-Andre Leibovici

Table of Contents

Preface	1
Chapter 1: Components of VMware View	7
Core components of VMware View	8
vCenter Server	8
View Connection Server	10
The types of VMware View Connection Servers	11
View Agent	12
View Client	12
Optional component—VMware View Composer	13
Introduction to View Composer	13
Using vCenter's SQL Express Installation for View Composer	14
Snapshots and linked clones	14
Linked clones	14
Templates	16
Full provisioning versus linked clones	16
Types of disks	16
OS Disk	17
User Data Disk	17
Temp Data Disk	17
Many options of disk types and redirection	18
Thin provisioning versus thick provisioning	19
Optional component—VMware View Transfer Server	21
Checking out	22
Checking in	23
Replication	23
Rollback	24
Summary	25
Chapter 2: Solution Methodology	27
Assessment	28
Questionnaire	29
Assessment worksheet from VMware View 5 Desktop Virtualization Solutions	29

Metric collection	32
Processing the data	36
Use case definition	40
Design overview	41
Storage	42
Isolation at the data store level	43
Networking	44
Compute	44
VMware vSphere and View desktop pool infrastructure	45
Pod architecture	45
Application distribution infrastructure	46
User persona management	46
What is a user persona?	46
Connection infrastructure	47
End devices	48
People	48
Validation	48
VMware View Planner tool (formerly VMware RAWC)	49
Comparing storage platforms	50
Summary	51
Chapter 3: Persistent or Non-Persistent vDesktops	53
Persistent desktops	55
Example scenario	55
Non-persistent desktops	57
Example scenario	58
Other non-persistent notes and considerations	59
Multisite solutions	60
Why is a non-persistent vDesktop best for a multisite?	62
Why distance matters	63
Profiles in the cloud	63
Hybrid: persistent mixed with non-persistent	64
How to choose	65
Summary	66
Chapter 4: End Devices	67
Thick clients	68
Repurposing thick clients	69
Thin clients	70
Teradici PCoIP-powered zero clients	71
Other clients	73
Choosing the proper device	73
A one-cable zero client solution	74

Summary	75
Chapter 5: The PColP Protocol	77
Why lossless quality is important	78
PColP network fundamentals	78
The two types of PColP connections	79
Multimedia redirection	80
The MMR perfect storm	81
Teradici APEX offload card	82
The offload process	84
Defining the offload tiers	85
Design considerations	85
Summary	86
Chapter 6: Sizing the VDI	87
Network considerations	89
Sizing the network	89
Network connection characteristics	91
DHCP considerations	92
Virtual switch considerations	94
Standard versus distributed switches	95
Port binding	95
Port binding and VMware View Composer	97
Compute considerations	97
Working with VMware vSphere maximums	100
Solution example—25,000 seats of VMware View	101
Solution design—physical server requirements	101
Solution design—the pod concept	103
Solution design—pools	112
Solution design—the formulae	113
Summary	113
Chapter 7: Redundancy	115
Physical infrastructure	115
VMware High Availability	115
Do you even need VMware HA?	116
VMware Distributed Resource Scheduling	122
Anti-affinity	123
VMware vCenter Server	124
VMware vCenter Server Heartbeat	125
Why VMware vCenter Server Heartbeat should be used	126
VMware View	126
Replica	126
Load balancing	127
VMware Fault Tolerance	128

Design impact when using VMware FT	129
Parent vDesktop and templates	131
Templates	131
Parent vDesktops with snapshots	132
User personas	132
Summary	135
Chapter 8: Sizing the Storage	137
VMware View Composer	138
VMware vSphere files	147
VMware View specific files	148
Tiered storage	148
Replica disk	149
Internal disk	150
Delta/differential disk	151
Disposable disk	152
Windows paging files	153
Temporary internet files	153
Persistent disk	154
Storage overcommit	156
Storage overcommit level options	158
Storage protocols	160
Maximums and limits	160
64 – to 140 linked clones per datastore (VMFS)	161
250 linked clones per datastore (NFS)	161
32 full – clones desktops per datastore (VMFS)	162
8 hosts per vSphere cluster with View Composer	162
1,000 clones per replica	162
Storage I/O profile	163
Read/write I/O ratio	165
Storage tiering and I/O distribution	169
Disk types	173
Capacity sizing exercises	174
Sizing full clones	174
Scenario 1	174
Scenario 2	175
Sizing linked clones	177
Parent VM	177
Replica	177
Scenario 1	178
Scenario 2	179
vSphere 5.0 video swap	181
Summary	185

Chapter 9: Security	187
The inherent security of VDI	187
Firewalls, zones, and antivirus	188
The fundamentals – firewall rules	189
Virtual enclaves	192
The jailbreak scenario	194
USB redirection and filtering	196
USB filtering at the end device	197
USB filtering via View Connection Server	198
USB filtering via the Windows operating system	198
Smart card authentication	201
Configuring smart card authentication for VMware View	
Connection Servers	206
Preparing the environment for smart card authentication	207
Configuring smart card authentication for VMware View Security Servers	208
Configuring U.S. Department of Defense CAC Authentication	209
Certificate revocation configuration	211
Configure the use of a CRL	211
Configure the use of OCSP	212
Configure the use of both a CRL and OCSP	212
Prohibiting the use of Copy and Paste functions	212
View Connection Server tags	214
Forensics	217
Summary	218
Chapter 10: Migrating from Physical Desktops to Virtual Desktops	219
Migration of the user persona	220
Separating the persona from the operating environment	220
Folder redirection	221
Profiles	222
Cutting over from physical to virtual	226
The use of VMware View User Data Disks	226
Operational considerations with user data	227
Summary	228
Chapter 11: Backing Up the VMware View Infrastructure	229
Backing up the VMware View Connection Server environment	230
Security server considerations	231
Transfer server and ThinApp repository considerations	231
Restoring the VMware View environment	232
Backing up the gold templates	232
Backing up the Parent VM	232

Summary	233
Chapter 12: VMware View 5.1	235
Platform features	235
Content-Based Read Cache (also known as View Storage Accelerator)	236
CBRC storage sizing	238
Host memory sizing	239
Managing CBRC	240
View Composer Array Integration	242
Support 32 (up from 8) hosts in a cluster on NAS	243
Standalone View Composer Server	243
Customizable disposable disk drive letter	244
User experience and client features	245
Management and administration	246
UI enhancements and localization	246
Support of pre-created Active Directory Machine Accounts	248
VMware vCenter and View Composer Advanced Settings	249
Phone home	250
Persona management	250
Security	251
Summary	252
Appendix: Additional Tools	253
VMware RAWC	253
VDI Fox	253
Websites and social media	254
Index	255

Preface

VMware View 5 Desktop Virtualization Solutions is meant as a guide for architects, solution providers, consultants, engineers, and anyone planning to design and implement a solution based on VMware View 5. It will refer to real-world scenarios as they are likely the best teaching examples. It will explain the settings and configurations needed to have a successful solution as well as the reason behind the decisions.

This book is not meant to replace the official administration or installation guides for VMware View published by the great people at VMware. The administration and installation guides are used during the installation and implementation of the solution. The material in this book should be used during the design phase, which is before an implementation is underway.

The driving factors of VDI

Many agencies and organizations are looking at how to deliver desktops as a managed service while increasing end-point security and decreasing associated costs. Popular reasons to implement a VMware View solution include:

- **Security:** VDI removes sensitive data from the end device and improves the ability to manage, secure, patch, and audit large numbers of desktop resources.
- **Windows 7 migrations:** Organizations looking to migrate to Windows 7 are looking at VDI to ease the transition.
- **Technology/Hardware refresh:** The daunting task of replacing outdated PCs during a hardware refresh can incur significant operational costs and reduce productivity. This is an opportunistic time to migrate users to a VDI solution; in addition, existing PCs could be repurposed as thin or thick clients, extending their usable life.

- **Energy reduction:** Some VDI solutions can consume significantly less energy through the use of zero/thin clients and tailored hardware on the backend.
- **Device independence:** VDI can remove the limitations of maintaining a stringent, "Acceptable Client List" for an organization (for example, Dell Latitude 5400S and Mac Books only) and instead allow the end user community to use their preferred device that ultimately connects back into a managed VDI. As long as the device has a support View Client, it is permitted for use within the organization. This is often called, **Bring Your Own Device (BYOD)**.
- **Remote connectivity in times of crisis:** Whether it's H1N1, an erupting volcano, mega-blizzard, or a swarm of locusts, VDI can allow workers to still work when they can't physically get to their work area.

No matter the driving reason, VDI is a technology that has gained a lot of traction across many verticals all over the world. It's also likely that many server virtualization architects will be asked to include a VDI as part of their overall virtualized datacenter solution.

What this book covers

Chapter 1, Components of VMware View, covers the core concepts of VDI as well as the core concepts of the VMware View platform. This chapter also covers VMware vSphere components as they are related to a VMware View solution.

Chapter 2, Solution Methodology, covers a defined methodology, including assessment, use case definition, a VDI hierarchy to establish a common framework of solution design.

Chapter 3, Persistent or Non-Persistent vDesktops, explains one of the most important design points of a VDI solution, desktop persistency. It also provides guidance on making the decision as well as benefits and drawbacks to each approach.

Chapter 4, End Devices, discusses the various end points that can be used to connect into a VMware View VDI. It also provides guidance on selecting the appropriate devices based on the environment and organizational requirements.

Chapter 5, The PCoIP Protocol, explains the protocol behind VMware View, Teradici's PCoIP. It also covers performance tuning, the APEX offload card, and best practices around implementing a solution with PCoIP.

Chapter 6, Sizing the VDI, focuses on sizing the core components of a VMware View solution, including Connection Servers and VMware vCenter Servers. It also discusses designing the solutions with VMware vSphere maximums in mind.

Chapter 7, Redundancy, focuses on building a robust and resilient VDI solution. It also explains how full redundancy can be designed and delivered, as well as design considerations and overall environmental impact.

Chapter 8, Sizing the Storage, focuses on one of the most complex components of VDI design, the underlying storage environment. It also covers both high-level and in-depth technical considerations, and design aspects of the storage system supporting the VDI.

Chapter 9, Security, focuses on hardening of the VDI as well as robust authentication mechanisms. It also discusses security considerations for specific environments, such as government agencies.

Chapter 10, Migrating from Physical Desktops to Virtual Desktops, discusses techniques to successfully migrate a user base from a physical desktop to a virtual desktop. It also focuses on user persona management and abstraction.

Chapter 11, Backing Up the VMware View Infrastructure, focuses on scheduling proper backups of a VMware View environment.

Chapter 12, VMware View 5.1, discusses the new capabilities in VMware View 5.1 along with **Content-Based Read Cache (CBRC)** and additional product highlights.

Appendix, Additional Tools, provides additional tools, online references, and suggested Twitter personalities that may prove helpful in designing a VDI solution.

What you need for this book

As this book is technical in nature, the reader needs to have a basic understanding of the following concepts:

- VMware vSphere
 - Hypervisor basics
 - vMotion
 - Cluster capabilities such as HA, DRS, and DPM

- Active Directory
 - Types of authentication
 - Encryption with certificates
 - Group policy objects
 - Folder redirection
 - Roaming profiles
 - DNS
- Virtual machine basics
 - VMX and VMDK files
 - Snapshots
 - VMware tools
- Networking
 - VLANs
 - DHCP
 - Port types
 - Routing
 - LAN and WAN basics

Who this book is for


The typical readers of this book would have a sound understanding of VMware vSphere fundamentals and would have been involved in the installation or administration of a VMware environment for more than two years.


Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Configure the ODBC connection and use <vCenter Server>/SQLEXP_VIM for the connection string. Replace <vCenter Server> with the appropriate information for your environment."

New terms and important words are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "This information can be found by opening the **Properties** tab from within **Device Manager** with the applicable device highlighted."

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title through the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Components of VMware View

Virtualization, the technology of abstracting the operating systems from the underlying physical server components, has become a cornerstone of the data center architecture. Virtualization allows organizations to run not just one operating system per physical server in the data center, but tens, dozens, or even hundreds, on a single physical server. The benefits of virtualization are many, including a reduction in hardware, power, and cooling costs. In addition, virtualization allows for new techniques of distribution and resilience to be applied, such as **VMware Distributed Resource Scheduler (DRS)** and **VMware High Availability (HA)**. Server virtualization, the virtualization of server operating systems on server hardware, is now a mainstream technology, which is readily accepted, adopted, and implemented in organizations across the world.

Virtual Desktop Infrastructure (VDI), the virtualization of desktop operating systems on server hardware, is another story.

The reason for the slower adoption of the virtual desktops was originally due to many factors, including an immature technology, lack of general understanding of a comprehensive solution, a proven delivery methodology, and a clear understanding of the success criteria of a given virtual desktop project.

Today, many of these hurdles have been removed. The supporting technologies from communication protocols to computing density, platform stability, and desirable end devices, now exist. Design methodologies have been built by some of the largest integrators in the world; yet virtual desktop projects continue to fail, falter, or stall.

This book will provide the architect, the engineer, the project manager, the freelance consultant, or the contractor, with a proven blueprint for success. More importantly, this book will teach the key success criteria to measure the most important design considerations to make and how to tip the probability of the project's success and sign-off in your favor.

Before these concepts can be covered in depth, it is important to understand the components of a **virtual desktop (vDesktop)** solution. The technology in this book focuses on VMware View, which is a market leader in VDI. While some concepts in this book apply specifically to VMware View-based solutions, many of the topics will help a VDI architect of any technology plan and build for success.

Core components of VMware View

This book assumes a familiarity with server virtualization, more specifically, VMware vSphere (sometimes referred to as ESX by industry graybeards). Therefore, this chapter will focus on:

- The VMware vCenter Server
- The types of View Connection Server
- Agent and client software

vCenter Server

VMware vCenter is a required component of a VMware View solution. This is because the View Connection Server interacts with the underlying **Virtual Infrastructure (VI)** through vCenter Web Service (typically over port 443). vCenter is also responsible for the complementary components of a VMware View solution provided by the underlying VMware vSphere, including VMotion and DRS (used to balance the virtual desktop load on the physical hosts). When an end customer purchases VMware View bundles, VMware vCenter is automatically included and does not need to be purchased via a separate **Stock Keeping Unit (SKU)**. In the environments leveraging vSphere for server virtualization, vCenter Server is likely to already exist. To ensure a level set on the capabilities that VMware vCenter Server provides, the key terminologies are listed as follows:

- **VMotion**: It is the ability to live migrate a running virtual machine from one physical server to another with no downtime.
- **DRS**: It is the vCenter Server capability that balances virtual machines across physical servers participating in the same vCenter Server cluster.
- **Cluster**: It is a collection of physical servers that have access to the same networks and shared storage. The physical servers participating in a vCenter cluster have their resources (for example, CPU, memory, and so on) logically pooled for virtual machine consumption.

- **HA:** It is the vCenter Server capability that protects against the failure of a physical server. HA will power up virtual machines that reside on the failed physical server on available physical servers in the same cluster.
- **Folder:** It is a logical grouping of virtual machines, displayed within the vSphere Client.
- **vSphere Client:** It is the client-side software used to connect to vCenter Servers (or physical servers running vSphere) for management, monitoring, configuration, and other related tasks.
- **Resource pool:** It is a logical pool of resources (for example, CPU, memory, and so on). The virtual machines (or the groups of virtual machines) residing in the same resource pool will share a predetermined amount of resources.

Designing a VMware View solution often touches on typical server virtualization design concepts such as the proper cluster design. Owing to this overlap in design concepts between server virtualization and VDI, many server virtualization engineers apply exactly the same principles from one solution to the other.

The first misstep that a VDI architect can take is that VDI is not server virtualization and should not be treated as such. Server virtualization is the virtualization of server operating systems. While it is true that VDI does use some server virtualization (for the connection infrastructure, for example), there are many concepts that are new and critical to understand for success.

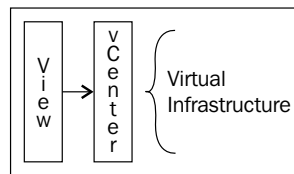
The second misstep a VDI architect can make is in understanding the pure scale of some VDI solutions. For the average server virtualization administrator with no VDI in their environment, he/she may be tasked with managing a dozen physical servers with a few hundred virtual machines. The authors of this book have been involved in VDI solutions involving tens of thousands of vDesktops, well beyond the limits of a traditional VMware vSphere design.

VDI is often performed on a different scale. The concepts of architectural scaling are covered later in this book, but many of the scaling concepts revolve around the limits of VMware vCenter Server. It should be noted that VMware vCenter Server was originally designed to be the central management point for the enterprise server virtualization environments. While VMware continues to work on its ability to scale, designing around VMware vCenter server will be important.

So why do we need VMware vCenter in the first place, for the VDI architect?

VMware vCenter is the gateway for all virtual machine tasks in a VMware View solution. This includes the following tasks:

- The creation of virtual machine folders to organize vDesktops
- The creation of resource pools to segregate physical resources for different groups of vDesktops
- The creation of vDesktops
- The creation of snapshots



VMware vCenter is not used to broker the connection of an end device to a vDesktop. Therefore, an outage of VMware vCenter should not impact inbound connections to already-provisioned vDesktops as it will prevent additional vDesktops from being built, refreshed, or deleted.

Because of vCenter Server's importance in a VDI solution, additional steps are often taken to ensure its availability even beyond the considerations made in a typical server virtualization solution.

Later in this book, there is a question asking whether an incumbent vCenter Server should be used for an organization's VDI or whether a secondary vCenter Server infrastructure should be built.

View Connection Server

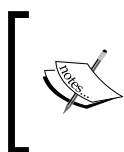
View Connection Server is the primary component of a VMware View solution; if VMware vCenter Server is the gateway for management communication to the virtual infrastructure and the underlying physical servers, the VMware View Connection Server is the gateway that end users pass through to connect to their vDesktop. In classic VDI terms, it is VMware's broker that connects end users with workspaces (physical or virtual). View Connection Server is the central point of management for the VDI solution and is used to manage almost the entire solution infrastructure. However, there will be times when the architect will need to make considerations to vCenter cluster configurations, as discussed later in this book. In addition, there may be times when the VMware View administrator will need access to the vCenter Server.

The types of VMware View Connection Servers

There are several options available when installing the VMware View Connection Server. Therefore, it is important to understand the different types of View Connection Servers and the role they play in a given VDI solution.

Following are the three configurations in which View Connection Server can be installed:

- **Full:** This option installs all the components of View Connection Server, including a fresh **Lightweight Directory Access Protocol (LDAP)** instance.
- **Security:** This option installs only the necessary components for the View Connection portal. View Security Servers do not need to belong to an Active Directory domain (unlike the View Connection Server) as they do not access any authentication components (for example, Active Directory).
- **Replica:** This option creates a replica of an existing View Connection Server instance for load balancing or high availability purposes. The authentication/LDAP configuration is copied from the existing View Connection Server.



Our goal is to design the solutions that are highly available for our end customers. Therefore, all the designs will leverage two or more View Connection Servers (for example, one Full and one Replica).

The following services are installed during a Full installation of View Connection Server:

- VMware View Connection Server
- VMware View Framework Component
- VMware View Message Bus Component
- VMware View Script Host
- VMware View Security Gateway Component
- VMware View Web Component
- VMware VDMDS

VMware VDMDS provides the LDAP directory services.

View Agent

View Agent is a component that is installed on the target desktop, whether physical (seldom) or virtual (almost always). View Agent allows the View Connection Server to establish a connection to the desktop. View Agent also provides the following capabilities:

- **USB redirection:** It is defined as making a USB device — that is connected locally — appear to be connected to vDesktop
- **Single Sign-On (SSO):** It is done by using intelligent credential handling, which requires only one secured and successful authentication login request, as opposed to logging in multiple times (for example, at the connection server, vDesktop, and so on)
- **Virtual printing via ThinPrint technology:** It is the ability to streamline printer driver management through the use of ThinPrint (OEM)
- **PCoIP connectivity:** It is the purpose-built VDI protocol made by Teradici and used by VMware in their VMware View solution
- **Persona management:** It is the ability to manage a user profile across an entire desktop landscape; the technology comes via the **recovery time objective (RTO)** acquisition by VMware
- **View Composer support:** It is the ability to use linked clones and thin provisioning to drastically reduce operational efforts in managing a mid-to-large-scale VMware View environment

View Client

View Client is a component that is installed on the end device (for example, the user's laptop). View Client allows the device to connect to a View Connection Server, which then directs the device to an available desktop resource. Following are the two types of View Clients:

- View Client
- View Client with Local Mode

These separate versions have their own unique installation bits (only one may be installed at a time). View Client provides all of the functionality needed for an online and connected worker. If Local Mode will be leveraged in the solution, View Client with Local Mode should be installed.

VMware View Local Mode is the ability to securely check out a vDesktop to a local device for use in disconnected scenarios (for example, in the middle of the jungle).

There is roughly an 80 MB difference in the installed packages (View Client with Local Mode being larger). For most scenarios, 80 MB of disk space will not make or break the solution as even flash drives are well beyond an 80 MB threshold.

In addition to providing the functionality of being able to connect to a desktop, View Client talks to View Agent to perform the following tasks:

- USB redirection
- Single Sign-On

Optional component—VMware View Composer

The components covered earlier in this chapter belong to the set of mandatory components in a VMware View solution. The major component that is optional in a VMware View solution is View Composer. It should be noted that when some third-party solutions such as Unidesk or storage-based cloning are used in conjunction with VMware View, View Composer is not used. This is because solutions such as Unidesk or storage-based cloning have their own approach for handling mass provisioning of vDesktops.

View Composer is used in the majority of view-based solutions today, but there are very valid scenarios and solutions that do not require the use of View Composer. As this book focuses on VMware View solutions and not VMware View with third-party components, View Composer will be discussed heavily throughout this book.

Introduction to View Composer

View Composer is the component that manages the deployment of linked clones, described later in this chapter, for desktop VMs from a single base snapshot.



View Composer is installed on vCenter Servers only.

View Composer also uses a separate database to store the information regarding mapping, deployment, and so on of the linked-clone desktops. This database can reside on the same database server as the existing vCenter database, assuming that it is a supported platform. However, the database itself must be unique to View Composer. This means that the View Composer database cannot use the existing vCenter Server database (but it could use the same server with a separate database instance).

In addition, a separate **Open Database Connectivity (ODBC)** connection must be set up on the vCenter Servers with the appropriate information for the View Composer database connection.



If View Composer is used, only automatic pool types are supported.
Also, the database instance must be unique to View Composer.

Using vCenter's SQL Express Installation for View Composer

Small **Proof-of-Concept (PoC)** environments may want to leverage the existing SQL Express installation on their VMware vCenter Server. It is possible to leverage the same SQL Express instance as long as a separate database is created. To create a separate database, perform the following steps:

1. Download and install SQL Server Management Studio Express.
2. Connect to the vCenter Server instance of SQL Express.
3. Right-click on the instance name and add a new database (for example, `View_Composer`).
4. Configure the ODBC connection and use `<vCenter Server>/SQLEXP_VIM` for the connection string. Replace `<vCenter Server>` with the appropriate information for your environment.

Snapshots and linked clones

A snapshot saves a point-in-time state of a given virtual machine. Changes beyond the snapshot of the point-in-time are written to a delta disk while the original virtual disk (`.vmdk`) is marked as read-only. This preserves the point-in-time state of the virtual machine until the snapshot is deleted by an administrator. Multiple snapshots of a given virtual machine can be taken, and it is these point-in-time snapshots that are used as the basis for VMware View Composer linked clones.

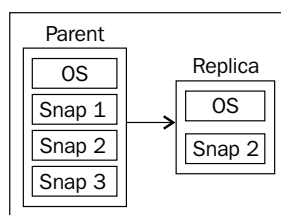
Linked clones

A **linked clone** is a copy of a virtual machine based on a specific snapshot of that virtual machine (known as the **parent**). When a linked clone pool is created, VMware View Composer creates a replica.

The **replica** is the original read-only base virtual machine disk merged with a specific point-in-time snapshot chosen to be the point of deployment for a given VMware View desktop pool. Replicas are always thin provisioned.

A View desktop pool can only point to one specific snapshot at a time but this can be changed easily through techniques discussed later in this book. A virtual machine can have multiple snapshots, thus a single virtual machine with multiple snapshots could be the foundation for all the View desktop pools in an environment. This allows each pool to be based off of their own (or the same) point-in-time snapshot. This is possible because View desktop pools using the linked clone technology do not actually use the base virtual machine snapshots; instead they use a replica (base virtual machine + snapshot).

While linked clones are based off of an original parent VM, each linked clone still has a unique **Media Access Control (MAC)** address and virtual machine **Universally Unique Identifier (UUID)**.



The preceding diagram illustrates a parent virtual machine with three snapshots (Snap1, Snap2, and Snap3). Each snapshot represents a different point in time of the virtual machine. For example, the Snap1 snapshot may have Office 2007 installed; the Snap2 snapshot may have Office 2010 installed; and the Snap3 snapshot may have Office 2010 and Visio 2010 installed. In this example, the Snap2 snapshot was chosen for virtual desktop deployment. Once this snapshot has been selected and the desktop pool has been enabled for provisioning, a replica is created. The replica does not copy the other Snap1 or Snap3 snapshot states.

The use of a replica, which preserves the original parent vDesktop's snapshot state, allows the parent vDesktop to be powered on, patched, or altered without impacting on the state of the vDesktop using the replica. Again, this is because the replica is a copy of a parent vDesktop's snapshot, and not the actual parent vDesktop itself.

Templates

A **template** is a virtual machine that has been marked read-only by converting it into a template. A template is simply a virtual machine, which has had its .vmx configuration file converted into a .vmxt configuration file. Virtual machines are read-only virtual machines that are then used for cloning purposes. A virtual machine created from a virtual machine template is a direct copy of the original template. However, customization specifications can be used to alter certain aspects (for example, SID, hostname, IP address, and so on) of the newly created virtual machines. Customization specifications are created by using the Customization Specification Wizard in the vSphere Client when connected to a vCenter Server.

Full provisioning versus linked clones

VMware View has the ability to use both full clones and linked clones. A **full clone** is a 1:1 independent copy of an existing VM template. This follows the same procedure as deploying a virtual machine from a template in VMware vCenter. A template is selected as the base for the vDesktops and (likely) a customization specification is also chosen.

A vDesktop that has been deployed using full provisioning (for example, think: virtual machine template) does not require access to the original template once it has been built.

A linked clone uses one master VM and then creates a delta disk for each additional VM. The additional VMs have a pointer back to the master VM when they need to talk to their base image (for example, to access the core Windows OS components) but use their delta disk for any unique data for that particular VM or user (for example, . . . \Documents and Settings\). A desktop built using the View Composer technology will have read-only access to the replica VM and read/write access to its delta disk.



Full clones are based off of a VM template, whereas linked clones are based off of a VM snapshot.

Types of disks

There are three types of disks – OS Disk, User Data Disk, and Temp Data Disk.

OS Disk

The OS Disk stores the system data (for example, Windows 7) and provides the base for a functional desktop.

Secondary OS Disk

The secondary OS Disk stores OS data that must be preserved during certain View Composer activities (such as Refresh or Recompose). Each virtual desktop will have a secondary OS Disk. These disks are typically 10 MB in size and are not configurable.

The secondary OS Disk can only be stored on the same data store as the OS Disk.

User Data Disk

A persistent User Data Disk is an optional component of a VMware View virtual desktop. The User Data Disk stores user profile information. By storing this information on a persistent disk, View Composer actions such as Refresh and Recompose will not affect the user profile data. The alternative is to store this information inside the OS Disk, which would cause user profile data to be lost during a Refresh or Recompose task.

The size of the User Data Disk is configurable. In addition, the persistent User Data Disk can be stored on the same data store as the OS Disk or on a separate data store.

Temp Data Disk

A non-persistent Temp Data Disk is an optional component of a VMware View virtual desktop. It is also referred to as the **disposable disk**. The Temp Data Disk stores the OS swap file as well as temporary data that is created during a user session. By storing this information on a non-persistent disk, VMware View will delete all data stored on the disk whenever the virtual desktop is powered off. This can help minimize the growth (in MB) of the virtual desktop as disposable user interaction is discarded and does not become part of the standard OS Disk for each respective user.

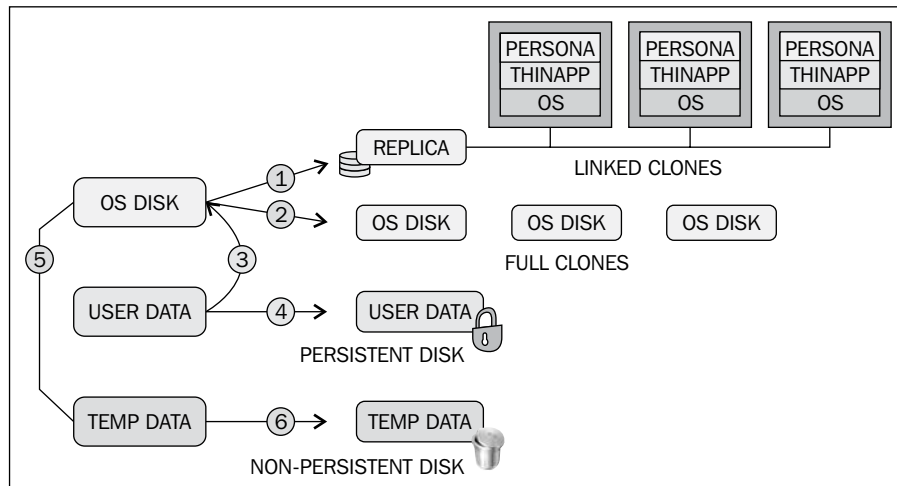
The size of the Temp Data Disk is configurable. The Temp Data Disk can only be stored on the same data store as the OS Disk.

Many options of disk types and redirection

Following is a list of the available options of the disk types and the redirection:

- OS Disk
 - Linked clones (1)
 - Full clones (2)
- User Data
 - OS Disk (3)
 - Persistent disk (4)
- Temp Data
 - OS Disk (5)
 - Non-persistent disk (6)

The following diagram illustrates the preceding disk types and their redirection:



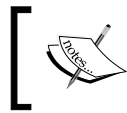
In the preceding diagram, the secondary OS Disk is not illustrated as it is not a configurable option within VMware View.

Thin provisioning versus thick provisioning

When a virtual disk is created using **thin provisioning**, the disk only occupies the actual data size on the disk. For example, a virtual disk (.vmdk) that is 20 GB in size but only has 8 GB of data will occupy only 8 GB on the data store. Two virtual desktops that have a 20 GB virtual disk but only 8 GB of data per disk will occupy 16 GB on the data store.

When a VM that is using thin provisioning needs to write new data to the virtual disk (and thereby increase the size of the virtual disk), it does so in the blocks defined by the data store's block size. The data store's block size is defined prior to being formatted in the **Virtual Machine File System (VMFS)** format. In VMware vSphere 5, the newly created (versus upgraded from vSphere) data stores use a unified block size of 1 MB.

For example, if the VM lives on a 500 GB VMFS-3, which is the data store that was formatted using a 2 MB block size, a 10 MB new write operation will require the write of 5 blocks (10 MB file/2 MB block size), which results in a less efficient use of the overall storage space.



Thin provisioning makes it possible to over-allocate the available storage and can introduce significant issues if not monitored and managed properly.

When a virtual disk is created using **thick provisioning** (default), the disk occupies its entire allocated size on the disk. For example, a virtual disk that is 20 GB in size but only has 8 GB of data will occupy 20 GB on the data store. Two virtual desktops that have a 20 GB virtual disk but only 8 GB of data per disk will occupy 40 GB on the data store.

View Composer uses linked clone technology. A virtual desktop using this technology contains a pointer back to a replica of the original gold snapshot. To clarify, the pointer is not to the original (parent) vDesktop itself, but an exact copy (replica) of the parent vDesktop from a specific point in time. View Composer uses this technology so that each vDesktop doesn't need its own full-sized virtual disk. The pointer uses the replica for read-only access only and writes all changes to a secondary disk, called the delta disk.

Delta disks can be viewed as a change record. Instead of defiling a gold snapshot, all changes (deltas) to the original disk are recorded outside of the gold snapshot, to the delta disk. This leaves the original gold snapshot in pristine condition while still allowing for a usable operating system that accepts changes made by the user and other applications.

Reset, Refresh, Recompose, and Rebalance actions for linked clones

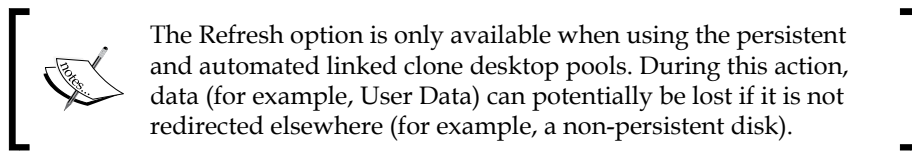
When using linked clones, there are several actions that can be performed on the clones themselves. These actions are as follows:

Reset

The Reset action is equivalent to hitting the **Reset** button on a virtual machine. This is an ungraceful restart of a virtual machine that is equivalent to pulling the power cable out of a desktop and then plugging it back in.

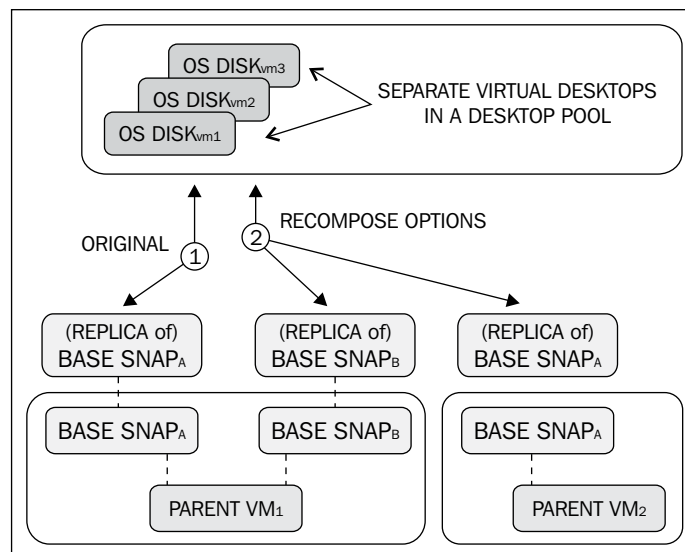
Refresh

The Refresh action is an action that resets the delta disk back to its original state. OS delta disk bloat can happen as a user continues to write changes to his delta disk over time. Data inside the OS delta disk is lost during a Refresh action.



Recompose

The Recompose action is an action that is used to change the snapshot and/or parent VM that is used by the desktop pool:



In the preceding image, number 1 is the original mapping to a base snapshot, number 2 represents the available options during a Recompose action.

Administrators can use the Recompose action in the following scenarios:

- To change the linked clone pool to use a different snapshot (for example, Snapshot_B) of the original parent (for example, ParentVM_1) instead of the current one in use
- To change the linked clone pool to use a snapshot (for example, Snapshot_A) of a different parent (for example, ParentVM_2) instead of the current parent in use



The Recompose option is only available when using the persistent and automated linked clone desktop pools.

Rebalance

The Rebalance action is an action that will evenly distribute desktops across all of the available data stores in the desktop pool. The desktop must be in the ready, error, or customizing state (with no pending tasks or cancellations) to be rebalanced.

A Rebalance action automatically executes the Refresh action on the desktop as well, which resets the OS Disk back to its original state.

During a Rebalance action, the administrator can set whether to rebalance the desktop once the users log off of their desktop or to force all of the active users to log off.

The Rebalance action is also the only supported way to move linked clones to a new data store.

Optional component—VMware View Transfer Server

The VMware View Transfer Server is a new component in VMware View since version 4.5. The transfer server's primary role is to deliver the desktop virtual machines from the data center to the end local device for use in an offline scenario. Running a vDesktop locally on the end device is known as Local Mode in VMware View.

The transfer server also manages the repository for the virtual desktops that will be available for Local Mode and provides synchronization capabilities between the local image and the desktop image residing in the data center.



The transfer server cannot be installed alongside any other VMware View component.

The transfer server's repository can live in:

- A local directory on a local drive
- Shared access via **Universal Naming Convention (UNC)**

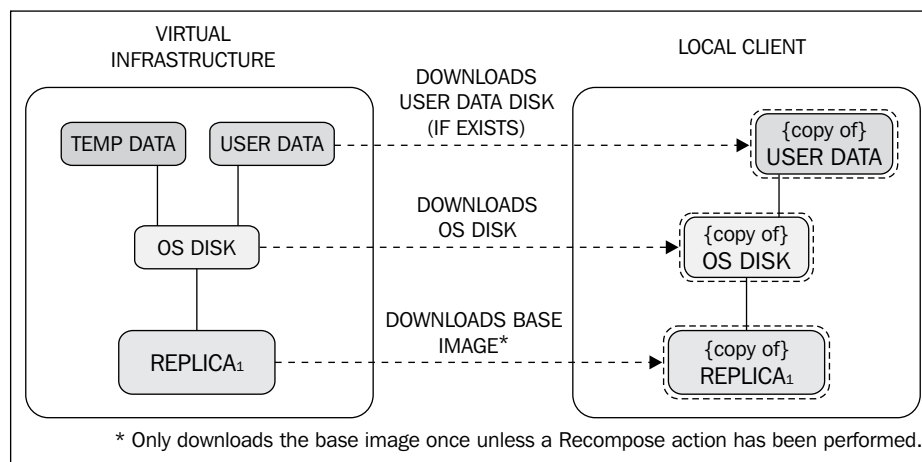
It is important to ensure that the transfer server's repository does not run out of space. For a point of reference, an average checked out Windows XP image will consume approximately 3 GB in the repository.

Additionally, the transfer server must use the following things:

- A static IP address
- The LSI parallel SCSI controller

Checking out

Checking out a virtual desktop is likely the most intensive operation the VMware Transfer Server will perform. The checkout operation involves downloading the entire published virtual desktop image from the View Transfer Server to the local client to be run in Local Mode; the local image is stored in an encrypted format and has a lifetime associated with the image. The checkout operation also places a lock on the virtual desktop to ensure that future inbound requests to the desktop from connecting to the instance running within the virtual infrastructure are prohibited:



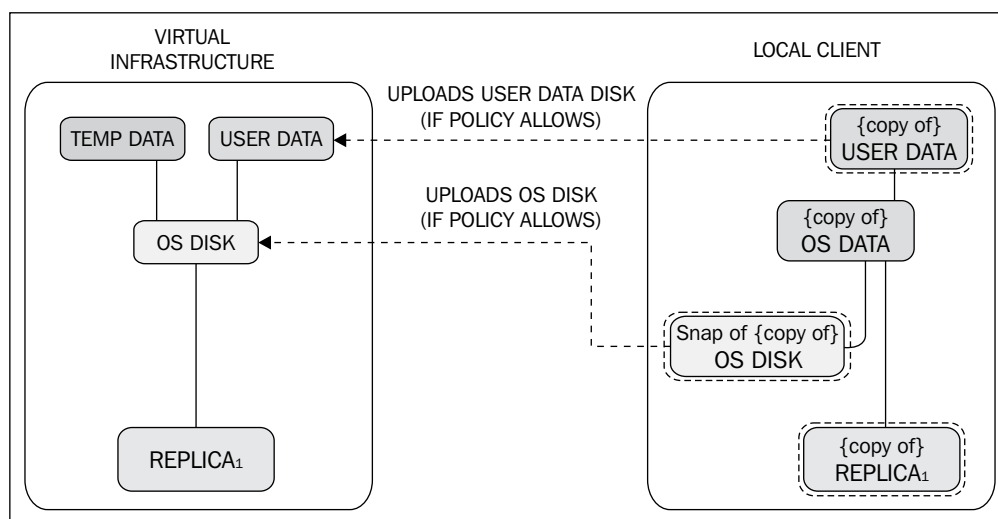
Due to the network bandwidth used (copying gigabytes of data), the checkout process should be performed when the end device is on the same LAN as the VDI. Performing a checkout process over a 3G hotspot could take hours or even an entire day, depending on the size of the vDesktop infrastructure.

Checking in

Checking in to a virtual desktop involves uploading the delta data and configurations to the VMware View vDesktop via the View Transfer Server. This data resides on the OS Disk and the User Data Disk (if it exists). The local copy of the base image is not uploaded as it has not been changed. The lock on the virtual desktop is released within the virtual infrastructure and future inbound connections to the desktop are directed to the virtual machine running within the data center (versus redirecting to Local Mode again).

Replication

Replication is the process of synchronizing a local vDesktop with its data center peer. Replication only transfers the delta data (changes generated by the user). Replication is managed by a replication policy, which configures settings such as frequency and user deferrals:

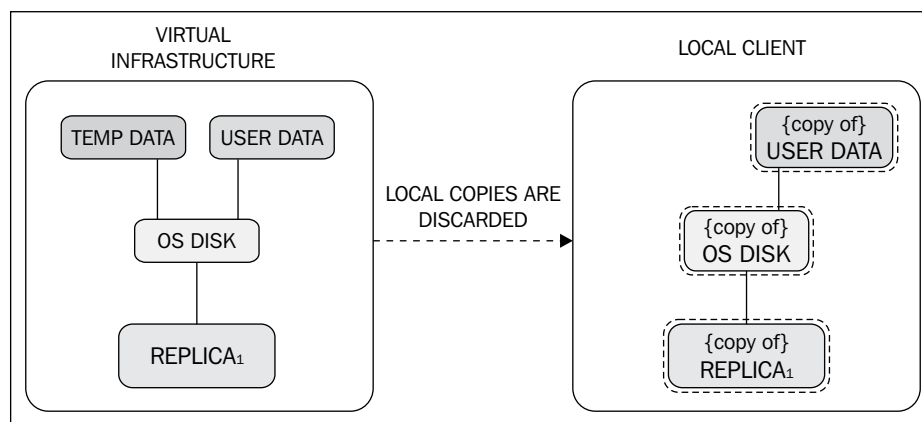


If changed user data from a Local Mode vDesktop is viewed as valuable data, it is important that a replication policy is defined with the View Admin console. Otherwise, if replication is not enforced, valuable data could be lost if an end device has a failure.

One of the benefits of VDI in general is the concept of keeping the data in the data center. Local Mode has several use cases (discussed later in this book) but does bring data back to the edge, even if encrypted. Local Mode should be used only in specific use cases and should not be looked at as a typical VMware View solution.

Rollback

Rollback is one method of removing access to a local desktop that has been checked out (the other method involves removing ownership). It is also possible for an end user to perform a rollback if it is allowed by the policy settings within View:



A rollback performs the following actions:

- If the user is currently logged into the checked out local desktop, the session is terminated and the user is prohibited from logging back into the local desktop. A new check out task must be performed to restore the Local Mode functionality.
- If the user is not logged into the checked out local desktop, all future login requests are sent to the instance running in the data center. A new checkout task must be performed to restore the Local Mode functionality.

Summary

This chapter has provided a solid introduction to the components of a VMware View VDI solution, including VMware vSphere fundamentals. Without an understanding of the basic concepts of the VMware View architecture as well as the underlying VMware vSphere architecture, it will be very difficult to build a proper VMware View solution. For additional reading on either VMware View or VMware vSphere, please refer to the Administration, Evaluation, and Installation guides from VMware on the desired product set.

The next chapter will cover how to collect an organization's inputs to ensure that a VMware View design meets the requirements for success. Collecting the requirements is a phase many VDI architects either skip or do in haste, resulting in a less-than-ideal end product. Once a VDI architect has performed several discovery engagements with an organization, it is certainly possible that he simply asks the likely pitfall questions (for example, "Will you be using a bidirectional audio?"), but until this level of comfort has been reached, performing a full discovery is advised.

Later in this book, the collected inputs of *Chapter 2, Solution Methodology*, will be used to produce a working design for an organization.

2

Solution Methodology

This chapter will focus on the solution approach required to design the successful VMware View environments, including the gathering of inputs, their digestion, and the production of a VMware View design, or project output.

A **Virtual Desktop Infrastructure (VDI)** implementation can have various project drivers, including cost reduction, increased security, or end device flexibility. When participating in a VDI endeavor, it is important to define the project success criteria. For example, if an organization is looking to decrease the login time for roaming physicians, it is important to incorporate a streamlined login process (for example, Single Sign-On) into the overall solution.



A robust VDI solution that doesn't address the organization's key criteria may still be judged as a mediocre success, instead of embraced a significant improvement to the end user work environment.

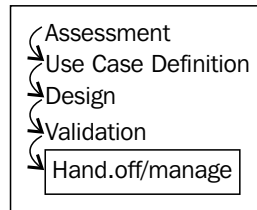


Also, for solution providers and architects that are designing VMware View solutions for multiple organizations, it is important to take a scientific and organized approach for designing View solutions. Designing the solutions for separate organizations in tandem can be very confusing. By taking the same approach to each project, the VDI architect can minimize errors and, more importantly, streamline the amount of effort required to produce a design.

The key phases of designing a VMware View solution are:

- Assessment
- Use case definition
- Design

- Validation
- Handoff/manage



As noted in *Chapter 1, Components of VMware View*, this book focuses on designing the solutions based on VMware View 5. In a full solution, there would be an implementation phase, which would follow the validation phase. These phases, which would include subtopics such as defining the project timeline, key milestones, handoff, and operational readiness would also be incorporated. Nevertheless, these topics are outside the scope of this book.

This book is a guide for architects to design the solution with the highest probability of success.

Assessment

A VMware View solution may replace existing physical desktops and require a migration phase or it may be a completely new solution to an organization. In either case, a proper assessment is required to understand how to scope hardware requirements, storage needs, and additional solutions that may be required (for example, a profile management solution).

An assessment is used to collect the necessary information in order to design the solution properly. An assessment is the process of collecting both technical information as well as organizational information. Typically, there are three components of an assessment:

- Questionnaire
- Metric collection
- Discussion

Questionnaire

A **questionnaire**, whether filled out manually or electronically, is a valuable tool to start collecting the necessary data to put together a solid VMware View design. A questionnaire is also a useful tool to help an organization feel involved in the design, which helps increase the chance of success.

An organization that participates actively in the VDI design will feel more invested in its success. An organization that was handed a VDI solution with little input will likely enter into their solution with scepticism.

The VDI Fox™ tool, produced by the authors of this book, has a questionnaire available from an iPhone or iPad, which can be used in the field in real time. The questionnaire is also available in the next section. It can be printed out and filled in on site, or the VDI Fox tool can be used. For more information about the VDI Fox application, please visit <http://www.redfoxllc.com/>.

Assessment worksheet from VMware View 5 Desktop Virtualization Solutions

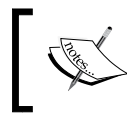
The following table shows the assessment worksheet from VMware View 5 Desktop Virtualization Solutions:

Question	Value	Description
Desktop pool inputs		
How many kiosks (explained after this table) will be supported by this solution?		It provides the total number of vDesktops that will be configured as kiosks with (likely) automatic login.
What is the maximum number of vDesktop users in the environment?		It provides the total number of vDesktops for a persistent vDesktop solution.
What is the maximum number of concurrent users that will connect at any given time?		It provides the total number of vDesktops for a non-persistent vDesktop solution.
What are the desired desktop operating systems for the VDI?		It determines a baseline of memory and vCPU required for the vDesktops.
What is the full size of the existing desktop images?		It determines how much disk space is required.
How often do you want the vDesktops refreshed?		It determines the desktop pool settings (refresh immediately, refresh after <i>n</i> minutes, refresh at <i>n</i> percent disk bloat, and so on).

Question	Value	Description
Number of point-in-time snapshots to maintain the base image?		It determines how much disk space is required.
Full highly available solution?		It determines whether redundant View Connection Servers, vCenter Servers, and so on, will be used.
Will remote workers be supported?		It determines whether View Security servers are required.
Will users be classified based on where they are connecting from?		It determines whether View Connection Server tags are required.
Should external users connect to a different VDI than internal users?		It determines whether View Connection Server tags should be used.
Will smart cards, CAC cards, proximity badges, and so on be used for authentication?		It determines the settings of View Client, View Agent, and the necessary certificates.
Do you adhere to FIPS 140-2 compliance?		It determines the advanced settings of the View environment.
Do you use two-factor RSA authentication?		It is used to configure advanced authentication options.
Are your remote users primarily in a connected state?		It determines whether the solution requires VMware View Local Mode.
Are USB drives allowed?		It determines the advanced View configuration settings.
Does the environment have shift workers?		It determines the desktop pool settings.
What is the average work day of an employee in hours?		It determines the desktop pool settings.
Storage inputs		
What is the amount of available space in the desktop images?		It determines the thin size of the image (full - space free = thin size).
How many desktop base images exist in the environment today?		It is used as an input for the number of desktop pools in the solution.
Landscape inputs		
How are desktops patched today?		It determines the patching solution for vDesktops.
What is your existing server infrastructure?		It is used to understand the technical landscape.

Question	Value	Description
What is your existing storage infrastructure?		It is used to understand the technical landscape.
What is your switching infrastructure?		It is used to understand the technical landscape.
Which group manages your virtual server environment?		It is used to understand the political landscape.
Which group manages your physical desktop environment?		It is used to understand the political landscape.
Who do you foresee managing your VDI?		It is used to understand the political landscape.
How technical are the people that will manage the solution once implemented?		It is used to understand the level of operational readiness that should be performed.
Current Microsoft desktop licensing agreement?		It determines whether the necessary Microsoft licensing exists.
Current asset tracking solution for physical desktops?		It determines the existing technology landscape.
Will new end devices be purchased for this solution?		It determines whether new devices (for example, zero clients) can be used or the existing hardware will be repurposed.
Network inputs		
Will CD or DVD burners be used?		It is used to understand the PCoIP bandwidth considerations.
Will microphone jack headsets be used?		It is used to understand the PCoIP bandwidth considerations.
Will Dragon Naturally Speaking or Dragon Medical be used?		It is used to understand the PCoIP bandwidth and storage IOPS considerations.
Will USB handsets, Dictaphones, and so on be used?		It is used to understand the PCoIP bandwidth considerations.
Will videos be streamed on a regular basis?		It is used to understand the PCoIP bandwidth considerations.
Will a VOIP solution be required?		It is used to understand the PCoIP bandwidth and end device considerations.
How far from the VDI will the majority of the users reside?		It is used to understand the network requirements and topology.
Will the solution reside on a sensitive or classified network?		It is used to understand the advanced configurations of the overall solution.

Question	Value	Description
How many VLANs are currently used by your physical desktop networking environment?		It is used to understand the the existing network layout.
Will you be supporting the remote offices?		It determines whether a remote office solution needs to be incorporated.
What is the current number of Dynamic Host Configuration Protocol (DHCP) scopes (and size) available for desktops?		It determines the existing network topology.
Profile management inputs		
What are you using today for profile management?		It determines whether profile management is required, and/or if there is a solution already in place.
Where do the users' home directories live?		It determines the existing profile management solution.
Success criteria		
Three months after the solution is live, how will you know you've chosen the proper solution?		It elicits the success criteria for the VDI project.
What is your primary motivation for implementing a VDI solution?		It elicits the success criteria for the VDI project.



A **kiosk** is used to identify an end device that will be configured to automatically connect to a vDesktop in the VMware View environment.

Metric collection

Once the questionnaire has been completed, it is time to begin collecting real-world data from the organization's existing physical landscape. By completing the questionnaire before collecting usage data, the metric collection phase can focus on the areas of concern.

For example, if during the questionnaire it was revealed that the VMware View solution would be supporting surgeons who work 10 hour days and use bidirectional audio (for example, Dragon Medical) often throughout the day, it would be beneficial to collect the data such as CPU, memory, and network usage of these surgeons, even if they are a very small subset of the overall end user population.

Assessments can be performed on an existing physical desktop infrastructure and on an existing VDI solution (for an organization looking to migrate from Citrix XenDesktop to VMware View, for example).

There are various ways of collecting the assessment data, including the use of Liquidware Labs Stratusphere Fit™. Fit uses the software running on the desktop that reports back to a central location (Stratusphere Hub). Some of the key metrics collected are as follows:

- CPU and memory usage (total and per application)
- Average login delay
- Average **Input/Output Operations Per Second (IOPS)**
- Network latency
- Peripherals

In addition to metric and inventory collection, Fit can be used to rate users on their suitability for a vDesktop solution.



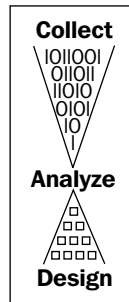
There are plenty of successful VDI solutions implemented that have never undergone a metric collection phase; however, these are likely implemented by very seasoned VDI professionals. If available, it is encouraged to execute the metric collection phase as it will drastically increase the project's chance of success, as well as give the data to support both the good and bad results as the project progresses.

The goals of the metric collection phase are as follows:

- Establishing a baseline average user
- Establishing the top ten applications used in the environment
- Identifying the current points
- Identifying the potential pitfalls for VDI
- Classifying the users into use cases

Fit can also be used to organize the future vDesktop users into use types, which translate into desktop pools within the View infrastructure.

For example, if Stratusphere Fit determines that the vast majority of users are allocated 2 GB of RAM on their physical desktop (and use 80 percent of the allocated memory), but a small contingent of users have 4 GB of memory (and use 80 percent of their allocated memory), then two different linked clone desktop pools (at a minimum) are likely needed. The first pool would have a base desktop image with 2 GB of memory, whereas the second pool would have a base desktop image with 4 GB of memory:



The preceding diagram demonstrates that the data must first be collected and then analyzed. After analysis the design can begin.

To properly determine how much physical hardware to procure as well as how to logistically divide the user population into the desktop pools, it is important to collect the information based on the real-world end user activities. Some of the key metrics to collect include the average and peak numbers as follows:

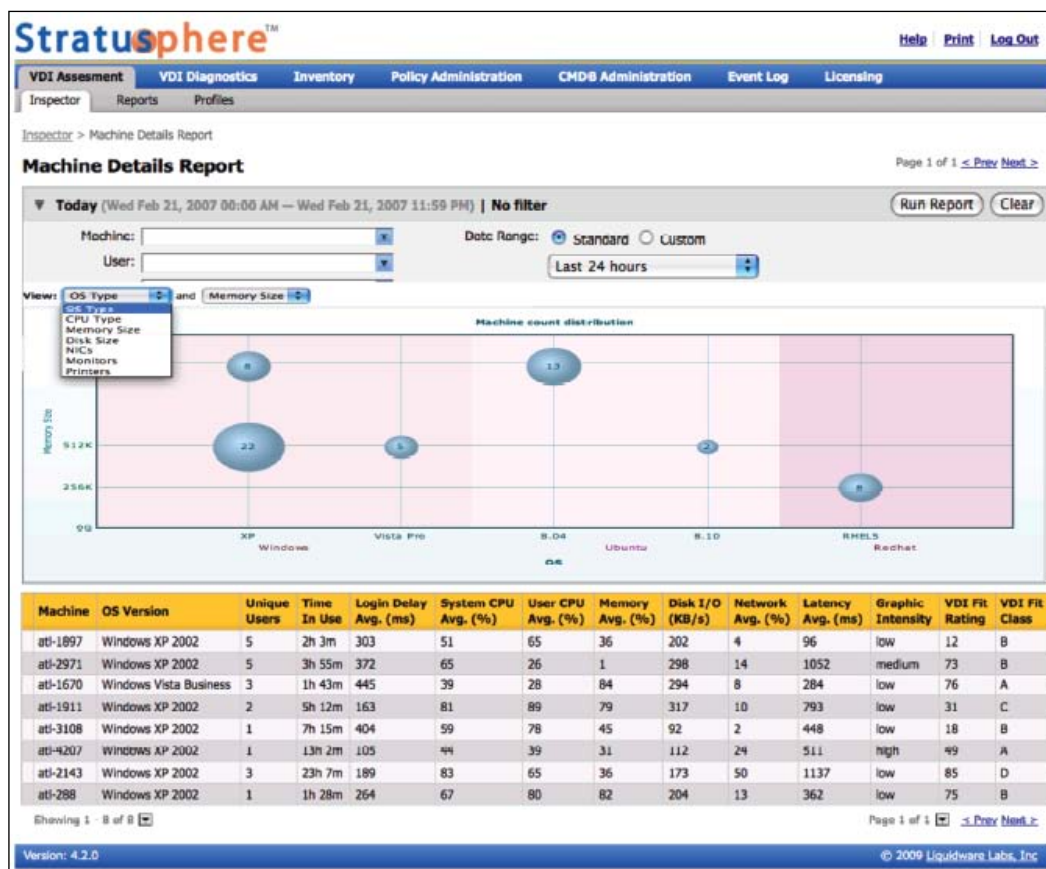
- Memory usage
- CPU usage
- CPU ready/wait time
- Network throughput
- Network latency
- Disk throughput (MBps)
- Disk activity (IOPS)
- Disk read/write percentage
- Most frequently used applications
- Number of monitors
- Unique peripheral requirements
- Unused applications
- Graphics intensity

With most collection utilities, an agent must be installed on the desktops to be surveyed. The most efficient way to install such an agent is typically through an existing mechanism such as group policy objects, Microsoft System Center Configuration Manager, logon scripts, or a variety of other methods.



While data collecting tools are often forbidden on the sensitive or classified networks, Liquidware Labs has taken the necessary steps and certifications to be able to run them on many sensitive networks.

Fit from Liquidware Labs generates unique reports that help a VDI architect properly understand the existing physical desktop environment so that a robust VMware View solution can be designed and implemented:



You can find the preceding screenshot at http://www.liquidwarelabs.com/images/Bubble_Graphs.png.

It is recommended to collect data between two and four weeks to ensure that enough time has passed to capture the most typical business cycles. If the organization that is being assessed has known peak work periods (for example, end of quarter closeouts), it may be helpful to collect the data during this time. However, it is advised not to let the collection of outlier data slow down the overall project's process. The careful analysis is important, but just as prolonged capacity planning assessments can bring a server virtualization project to its knees, so can an overly prolonged VDI assessment.

Processing the data

Once the end user data has been collected, it is important to properly comprehend and analyze the information that has been gleaned from the customer's environment, as shown in the following list:

- Memory usage (in MB)
 - **Input:** It is the total memory consumed (both active and peak) by the end users. The summation of this metric does not yield the total memory required for the VDI solution.
 - **Output:** It helps determine how much total memory is required by the physical environment. The formula is as follows:

MB required for vDesktops + MB required for supporting infrastructure + MB overhead of hypervisor + MB for swap space = Total MB required.

- CPU usage (in GHz)
 - **Input:** It is the total CPU usage consumed (both active and peak) by the end users. In a VDI environment, total CPU usage is likely less important than the users per core.
 - **Output:** This helps determine the total CPU usage required by the physical environment and is used to determine which users may require multiple vCPUs.

Why users per core?



Calculating users per core is more important than trying to determine the total number of MHz needed by a given user. Two key CPU-related metrics are CPU Ready and CPU Wait. **CPU Ready** is the time in milliseconds that a machine spends waiting for a CPU to become available. If the design implements too many vDesktops per core, the end users will experience a high CPU Ready time and their vDesktops will be sluggish as the vDesktops wait for one another to free up the CPU time for their own processing needs. **CPU Wait** is the time in milliseconds that a CPU spends waiting for an I/O to complete. A high CPU Wait is usually indicative of either a disk or a network bottleneck. For example, a particular vDesktop may be running a process that reads and writes the large amount of data to the local filesystem. If the underlying storage is unable to meet the performance needs of the vDesktop's process, a high CPU Wait time may be realized.

- Network throughput (in KBps)
 - **Input:** It is used to determine potential bottlenecks in both an on-site and/or a remote solution. As the VDI solutions are often implemented in a central location, the network throughput can be used to determine the required bandwidth between a remote office and the location of the VDI.
 - **Output:** For example, an environment with minimal network throughput between the end users at a remote site and the centrally located VDI may choose to implement VDI at the remote office and manage it from a central location (one solution to limited network throughput). It is also important to understand any unusual network bandwidth requirements of the VDI users.
- Network latency (in milliseconds)
 - **Input:** It is used to determine the potential bottlenecks in both an on-site and/or a remote solution. As both the **Microsoft Remote Desktop Protocol (RDP)** and VMware's PCoIP protocol are latency sensitive, it's important to understand the round trip time for data transmissions between the end users and the location of VDI.
 - **Output:** For example, an environment with high network latency may choose to implement policies optimizing the PCoIP protocol at both the end devices and within the vDesktop. The policies may limit advanced graphics capabilities and use compression to maximize an end user's experience.

- Disk throughput (in MBps)
 - **Input:** It is used to determine potential bottlenecks in the underlying storage environment supporting VDI from a bandwidth perspective.
 - **Output:** For example, an environment with a class of users requiring large amounts of disk throughput (for example, recording audio) may need to have their vDesktops reside on a separate data store from other vDesktops users. This not only suggests and promotes the tiering of the storage platform but also prevents low disk throughput users from being impacted by high throughput users.
- Disk activity (in IOPS)
 - **Input:** It is used to determine potential bottlenecks in the underlying storage environment supporting VDI from a performance perspective.
 - **Output:** Disk IOPS is one of the most important metrics of a VDI design and, as such, a chapter later in this book is dedicated to the fundamental concepts of storage in a VDI solution. Collecting disk IOPS consumed by end users before a migration to a VMware View solution will provide an idea about the overall number of disk IOPS that need to be supported in the solution. However, as discussed later in this book, storage design plays a paramount role in the overall success of a VMware View solution.
- Disk read/write percentage
 - **Input:** It is used to help determine a suitable storage environment for the VDI.
 - **Output:** More on designing a suitable storage environment for the VDI can be found later in this book.
- Most frequently used applications
 - **Input:** A list of common applications used on a daily basis by the majority of users in a given user class.
 - **Output:** Understanding the most frequently used applications can help determine a candidate list for a given application virtualization (AppVirt) solution, for example, VMware ThinApp. The most frequently used application list may also be helpful in determining what applications will be included in a desktop pool's parent image.

- Number of monitors
 - **Input:** It is used to determine how many monitors a given user is using.
 - **Output:** As supporting additional monitors will change the memory settings of the vDesktop (and if using linked clones, all vDesktops in a given pool), it may be useful to separate multi-monitor users from single-monitor users. With a growing number of environments, two monitors is the standard and four monitors is abnormal. In this case, separating the four monitor users may prove beneficial:
- Unique peripheral requirements
 - **Input:** It is used to determine potential compatibility issues with a VMware View solution.
 - **Output:** For example, a class of users may require the use of webcams for collaboration. At the time of writing, the webcams were not supported by VMware View 5.
- Unused applications
 - **Input:** It is used to determine applications that are never or seldom used in a desktop environment.
 - **Output:** For example, an environment may have several applications that are included in the current gold image. However, it may likely make sense not to include such applications in the base image for the VMware View vDesktop deployments.
- Graphics intensity
 - **Input:** It is used to determine potential bottlenecks in the underlying storage environment supporting the VDI from a bandwidth perspective.
 - **Output:** For example, it is used to determine which users may need additional hardware (for example, blade PC) to provide the level of graphics required by an end user. In some environments, the capabilities of a vDesktop will not be able to provide the level of graphics capabilities required by a class of users. In these environments, a solution that integrates a blade PC solution into a VDI (for example, a solution from ClearCube) would be ideal.

Use case definition

Once sufficient data has been collected during the assessment phase, the data can then be analyzed to determine the specific use cases.

A **use case** is a collection of connection, performance, peripheral, and other characteristics for a group of vDesktop users. It is important to use the data collected in the assessment phase as well as observations (if possible) of the users in action to determine the primary use cases for a given environment.

For example, through a diligent assessment phase, the following may be determined:

- Total of 300 vDesktop users
- 50 of which require 2 monitors
- 25 of which have high performance needs that are fulfilled with 2 vCPUs and 6 GB of RAM
- 200 of which have average needs that are fulfilled with 2 vCPUs and 1.5 GB of RAM
- 25 of which require 4 monitors

If the design attempts to support all of the preceding requirements in a single desktop pool, the desktop pool would have support for 4 monitors and vDesktops with 6 GB of RAM. This type of design would waste a lot of physical resources (for example, 4 GB of additional memory for 175 vDesktops, totaling 700 GB of waste).

Therefore, a proper design would be as follows:

- A desktop pool of 25 desktops based on a gold image of 2 vCPUs and 6 GB of RAM
- A desktop pool of 50 desktops supporting up to 2 monitors
- A desktop pool of 25 desktops supporting up to 4 monitors
- A desktop pool of 200 desktops supporting the bulk of the user environment; based on a gold image of 2 vCPUs and 1.5 GB of RAM

Some key questions that can help determine use cases are as follows:

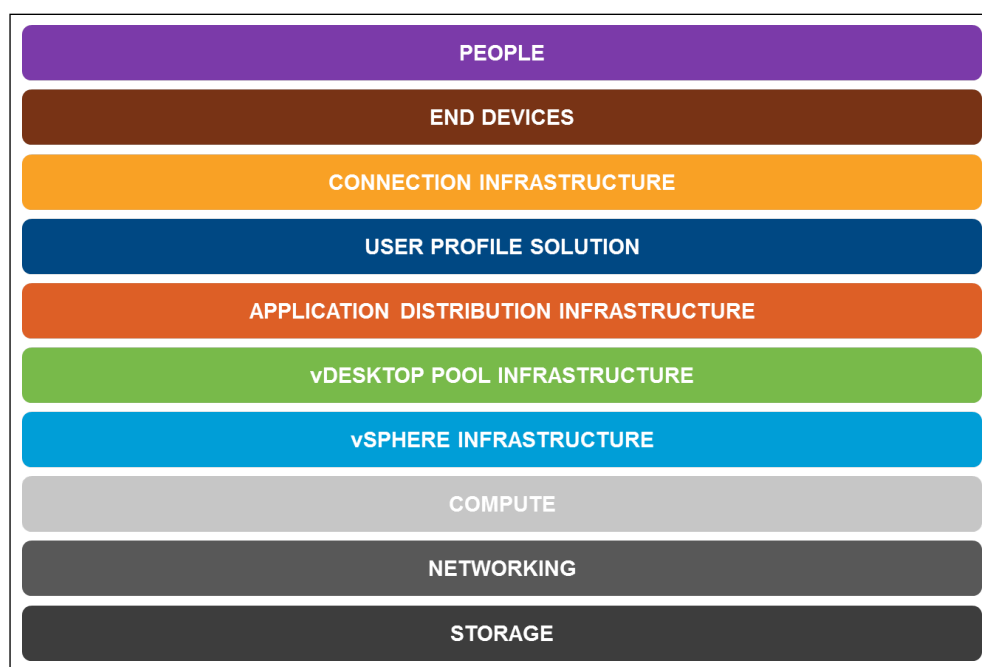
- What are the minimum performance requirements needed to provide a favorable vDesktop experience?
- Are there bidirectional audio requirements?
- How many monitors must be supported?
- Will the desktops be disposable (non-persistent)?
- How will the user profiles be managed (if applicable)?
- Are there any unique security or compliance needs?

By determining the various use cases supported by the VDI solution, the ultimate design can not only support the required use cases but also ensure that the underlying physical infrastructure is optimized.

At the end of the use case definition phase, the required inputs to design a solution have been collected, discussed with the organization, and agreed upon.

Design overview

A VDI solution builds on the complexity of a typical server virtualization solution by adding requirements for connection infrastructure, persona management (if applicable), desktop pool infrastructure, end devices, and so on, as shown in the following diagram, which shows the VMware View solution stack:



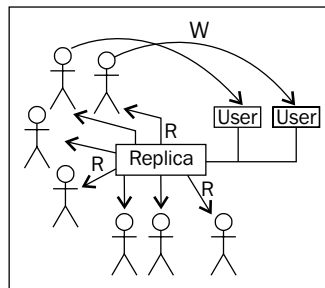
In addition, for experienced server virtualization architects, the requirements of a VDI solution are often significantly different (for example, the way a vSphere cluster may be designed), so it is important to respect the importance of sound VMware vSphere design while recognizing the new implications of a VMware View solution. For example, a VDI solution will likely have greater virtual machine to physical density than a classic server virtualization solution. In addition, the compute and storage demands can be more burst of nature, with higher peaks and shorter duration than a classic server virtualization solution.

While many of the following subsections (for example, storage, user persona management) will be covered in much greater detail later in this book, an overview of the methodology used for each design is provided in brief for each component in the following section.

Storage

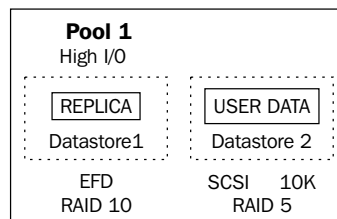
Storage is an area of high impact in regards to the overall VMware View solution. Without a properly designed storage infrastructure, the entire solution may ultimately fail.

This book focuses primarily on VMware View solutions that leverage the linked clone technology afforded by VMware View Composer. In designs using linked clones, a single replica disk is used for all vDesktops in a given pool. Therefore, storage considerations, for example, read IOPS are extremely important when designing the storage subsystem for a VDI solution:



In the preceding diagram, the heavy read load is illustrated as every user in a given desktop pool is reading from the same replica disk. However, each user will write to his/her own OS Disk (and persistent User Data Disk and/or non-persistent temporary data disk, if applicable).

In addition, understanding the application environment of the end users is important as applications such as Nuance's Dragon Medical could place a heavy load on the disk when performing transcription tasks. For users that require heavy storage I/O, it is advised to place them in their own desktop pool:

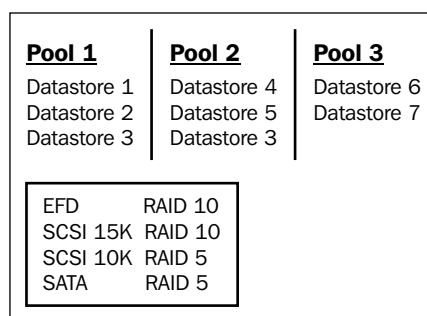


As shown in the preceding diagram, linked clones allow the use of storage tiering. In the preceding example, the replica disk (a high read I/O load) is placed on **Enterprise Flash Drives (EFDs)** protected with RAID 10. The User Data virtual disks are placed on SCSI 10K drives protected with RAID 5. Linked clones allow for the storage subsystem to be tailored economically, technically, and functionally to match the needs of the VDI.

Isolation at the data store level

When designing a VMware View solution, especially one leveraging linked clones, it is important to understand the inherent advantage of being able to separate desktop pools, user classes, and performance requirements on separate data stores.

In the following diagram, Pool 1 supports heavy users requiring an I/O intensive application, Pool 2 supports the general user population, and Pool 3 supports kiosks for walk-in customers:



In Pool 1, for example, the replica disk can reside on performance optimized EFDs (extremely high IOPS supported), with the user's write activity (for example, OS Disk) occurring on SCSI 15K disks in a RAID 10 set (high IOPS supported) and the user's non-persistent data residing on SATA RAID 5 (low IOPS supported) disks.

Furthermore, for Pool 2, the replica disks can reside on a separate EFD-based data store with the user's write activity occurring on a separate SCSI 10K RAID 5 data store. Pool 2 can also use the same data store for non-persistent temporary data, if so desired.

Finally, Pool 3 can also use separate data stores, from Pool 1 and Pool 2, for its replica disks and user OS Disks.

Why is this beneficial?

This is beneficial for the following reasons:

- **Cost savings:** In this solution, an entire storage array doesn't need to be populated with expensive EFDs; instead only enough EFDs need to be purchased to support the desired replica disks. On the contrary, for non-persistent and performance-insensitive data, cheap SATA disks can be used.
- **Performance tuning:** In this solution, the appropriate technology (for example, EFD) is used to deliver an optimized end user experience. By using tiered storage the disk optimized for heavy I/O can be used when necessary, but it doesn't have to be used for all disk activities.
- **Performance segregation:** In this example, extremely heavy use (for example, login storm, application batch processing, and so on) in Pool 1 should not impact end users in Pool 2 (assuming that the storage array network is not a bottleneck). Therefore, high I/O users (for example, bidirectional audio physicians) can be separated and segregated from moderate I/O users (for example, administrative staff).
- **Service-level agreements:** In this example, each pool uses its own data store for replica disks. However, if each pool uses the same data store for its replica disks, it would be possible for the actions and activities in one desktop pool to impact another, making adhering and enforcing **service-level agreements (SLAs)** extremely difficult.

Networking

There are several core networking components that must be considered when designing a VDI, **Dynamic Host Configuration Protocol (DHCP)** leases, **Domain Name System (DNS)** resolution, load balancing, and virtual switch technology. This is addressed in detail later in this book, but it is worth noting that engaging the organization's network team early in the conversation is highly recommended.

Compute

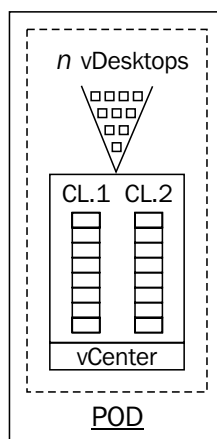
As addressed in *Chapter 1, Components of VMware View*, users per core is the primary measurement of consideration in the compute layer of a VMware View solution. With ever-increasing processing power and processing density becoming available, compute is becoming a layer that requires less attention, especially when using processors with six or more cores.

VMware vSphere and View desktop pool infrastructure

As discussed in more detail later in this book, a sound VMware vSphere infrastructure is extremely important to a robust VMware View solution. For seasoned vSphere architects, many of the concepts of a sound vSphere infrastructure will be familiar and practiced. However, VMware View solutions will likely exceed some of the maximums allowed, as defined by VMware. For example, in the current version of VMware vCenter, the maximum number of powered-on virtual machines is 10,000. In some large-scale VDI implementations, this number will easily be surpassed. Regardless, it is important to understand that 10,000 vDesktops may not be an acceptable notion from the management's perspective. This book will cover how to properly design a VMware vSphere infrastructure for large-scale VMware View solutions.

Pod architecture

As mentioned earlier, VMware View solutions may exceed the maximums allowed by VMware vSphere, the underlying virtual infrastructure platform. As such, collections of VMware vSphere environments, called pods, are used to provide modular building blocks for the VDI. A **pod** is a collection of physical servers running VMware ESX, one or more VMware vCenter Servers, and the supporting storage and networking infrastructure to provide n number of vDesktops. This number will vary with design, unique customer requirements, and the data gathered during the assessment phase:



In the preceding diagram, the pod consists of a VMware vCenter environment managing two clusters of eight hosts per cluster. The eight hosts per cluster configuration is the VMware View maximum configuration that is supported. Therefore, in most VMware View pods, a cluster will consist of between six and eight hosts per cluster (typically eight).

Application distribution infrastructure

Whether VMware ThinApp, Citrix XenApp, Microsoft App-V, or another solution is used to deliver applications to the desktops in an environment, understanding where the environment lives, how it is used, and its impact on the overall solution (for example, network implications) is important. This book will cover the key concepts of application virtualization as it relates to a VMware View solution. However, this book does not dive deep into the technical components of application virtualization.

User persona management

There are many options when it comes to managing the user persona in a VDI solution such as Microsoft Folder Redirection and the User State Migration Tool, Liquidware Labs ProfileUnity, and AppSense.

VMware's acquisition of **RTO** now provides an integrated persona management solution with VMware View 5. However, at the time of writing, the community of VMware View architects was not leveraging the built-in capability in large-scale solutions. Regardless, user persona management will be addressed later in this book as it is important to understand how they integrate (especially) in a non-persistent desktop pool, as well as potential network implications that user profiles may add.

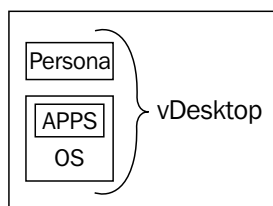
What is a user persona?

A **user persona** is a collection of settings, favorites, shortcuts, wallpapers, customizations, desktop icons, printers, and other settings unique to a given user.

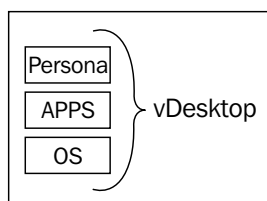
As illustrated previously, a vDesktop is composed of three abstract layers. These layers are OS, Apps, and Persona.

The OS layer is the execution environment for the vDesktops. This is primarily either Microsoft Windows XP or Microsoft Windows 7 (or planned to be Microsoft Windows 7).

The Apps (applications) layer can either be a separate layer by using an application virtualization solution (for example, VMware ThinApp) or can be included inside the OS layer:



In the preceding diagram, the Apps layer is not separate and is instead nested within the OS layer. This means that to update a single application for a desktop pool, the entire parent image must be updated:



In the preceding diagram, the Apps layer is independent of the OS layer, and thus can be updated independently. This allows administrators to update a single application without requiring massive changes to the composition of a VMware View desktop pool. In most instances, there are strong advantages to using an AppVirt solution for some or all of the applications for the VDI. The user base and performance characteristics of the applications may play a part in your overall decision on how you decide to deliver your applications.

Connection infrastructure

The connection infrastructure includes the brokers or VMware View Connection Servers, an optional load balancer, an optional WAN optimizer, a DNS solution, and any advanced routing capabilities (for example, Cisco Global Site Selector).

As ultimately all incoming connections will initially route to a VMware View Connection Server, it is imperative to keep at least one VMware View Connection Server available to end users at all times.

Once a connection has been successfully authenticated, the VMware View Client directly connects to the VMware View Agent within the vDesktop; this is referred to as a **direct connection**. For direct connections, a failure of the VMware View Connection Server once a user has already connected to their desktop does not impact existing connections. However, it does impact future connection requests.

It is possible to use the Microsoft protocol with VMware View. Microsoft RDP uses a tunnelled connection. This tunnelled connection opens a second HTTPS connection with the View Connection Server or View Security Server. In this scenario, the failure of the View Connection Server or View Security Server providing the secondary HTTPS connection will cause disconnection of the end user.

End devices

As detailed later in this book, there are various end devices that offer various benefits (and trade-offs). It's important to understand the multitude of devices available in the market to understand how they are incorporated into a VMware View solution. This book will cover the major categories of end devices, including thick clients, thin clients, zero clients, Apple iPads, and forward-looking end devices.

People

People are the most important layer of any VDI solution. It is important for the VDI architect to always understand that a project's success will very likely be measured by the people using the solution, their experience, their perception, and their transition to a vDesktop environment.

Not keeping people at the forefront of the solution can have catastrophic consequences. Everything from end devices to desktop pool configuration should be chosen with the happiness of the end user in mind.

Validation

Once a design has been formulated, vetted, and approved, then comes the time to validate the solution. Validation is normally performed on a representative subset of the full-scale solution. For example, in a solution that leverages two fully populated HP c7000 Blade chassis, a single half-populated chassis may be used to test the provisioning, basic functionality, and underlying storage performance. In some scenarios, disposable hardware is used simply to test the software components.

Leveraging an economically scalable hardware platform, such as the Nutanix converged virtualization appliance (<http://www.nutanix.com/>) or Pivot3 vSTAC (<http://pivot3.com/>) can help both prove out the design with a low cost of entry, as well as support a production solution in the future.

One of the best ways of validating a solution is to build a pilot environment according to the specifications detailed in the design document. Once the pilot environment is built, a pilot group of users can be enrolled to test user experience, functionality, and to apply a load on the underlying hardware.

VMware View Planner tool (formerly VMware RAWC)

The VMware View Planner tool, formerly known as the **Reference Architecture Workload Simulator (RAWC)** tool, is the best way to simulate end user connections and activity in a VMware View environment. For organizations looking to stress test VMware and competing solutions side by side, the Login VSI tool (<http://www.loginvsi.com/>) will support comparison testing. The VMware View Planner tool is for VMware View environments only.

The VMware View Planner tool is comprised of the following things:

- **Session launcher virtual machine:** This virtual machine launches the actual vDesktop sessions. Each session launcher virtual machine can support up to 20 concurrent sessions.
- **Controller virtual machine:** This virtual machine hosts the View Planner interface and the network share for the configuration and log files.
- **Target vDesktops:** The vDesktops used for stress testing have the View Planner code residing on each of the vDesktops (or as part of the parent image).
- **E-mail server virtual machine (optional):** This virtual machine is only necessary if Microsoft Outlook will be used as part of the View Planner testing and if there is no Microsoft Exchange Server available for testing purposes.

View Planner can be tailored to perform the following end user tasks:

- **Microsoft Word:** Insert text, save modifications, resize window, close application
- **Microsoft Excel:** Insert numbers, insert and delete columns and rows, copy and paste formulae, save modifications, resize window, close application
- **Microsoft PowerPoint:** Open a presentation, conduct a slideshow presentation, resize window, close application

- **Microsoft Outlook:** Set up a mailbox, send e-mail, execute send/receive, resize window, close application
- **Internet Explorer:** Browse a web page, resize window, close application
- **Windows Media Player:** View a video, close application
- **Adobe Acrobat Reader:** Open a PDF document, scroll through the document, resize window, close application
- **Java Runtime:** Run a Java application, close application
- **McAfee AntiVirus:** Execute a real-time virus scan, close application
- **7-Zip:** Compress files, close application

With this extensive list of capabilities, a variety of tests can be executed to test overall VDI performance, login storms, boot storms, storage performance, and so on.

Boot storm versus login storm

In researching, designing, discussing, and planning the VDI solutions, the boot storm and login storm terms will appear often; however, the two terms cannot be used interchangeably.

A **boot storm** is a significant enough number of vDesktops booting (or resuming from a suspension) to cause a decrease in the overall VDI performance. This can be caused due to one or more data stores that are unable to handle the I/O load, a file lock conflict with too many vDesktops/data stores, a DHCP lease request flood, and so on.

A **login storm** is a significant enough number of end users logging into the vDesktops to cause a decrease in the overall VDI performance. This can be caused due to creating initial profiles, streaming a user's persona, slow Active Directory performance, misaligned disks (for example, Windows XP without the proper offset), and so on.



Using RAWC in conjunction with a monitoring solution such as Liquidware Labs Stratusphere UX (User Experience) or RTO PinPoint can allow a VMware View architect to identify bottlenecks during the validation phase instead of during the implementation phase.

Comparing storage platforms

As RAWC View Planner provides a scientific approach to testing a canned (or random) series of tailored tests, the tests can become the control in an experiment where the storage platform is the variable.

For example, if the true advantages of EFD in a RAID 10 (over SCSI 15K RAID 10) are needed to be quantified, a saved RAWC test could be run against a desktop pool using EFD in the first iteration and SCSI 15K in the second iteration.

This technique can be used not only to test, quantify, and optimize a storage array and its configuration, but it can also be used to compare the storage arrays from multiple vendors.

Many thanks to Fred Schmischeimer for all of his help with the RAWC tool and for his information guide, *Workload Considerations for Virtual Desktop Reference Architectures* information guide.

Summary

VDI solution design is complex; by taking a measured approach that involves both hard data (metric collection) as well as organizational input (questionnaire), the chances of a suitable design increase drastically. This chapter discussed the layers involved in a VMware View solution, some of the pitfalls to watch out for, as well as some of the industry tools to use to help support the overall design and validation process.

The next chapter discusses one of the most important design considerations a VDI architect can make, *persistent or non-persistent*? The answer to this question is the cornerstone of the VMware View design, and, as such, needs to be fully understood to ensure the best possible design outcome.

3

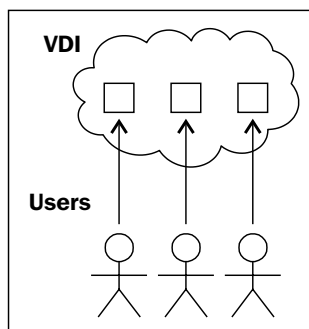
Persistent or Non-Persistent vDesktops

One of the most fundamental decisions that must be made early in the design process is whether to use persistent or non-persistent vDesktops. This decision may impact many areas of the overall VDI, including storage, desktop pools, and management.

In this chapter, we will cover:

- Persistent desktops
- Non-persistent desktops
- Multisite solutions
- How to choose for your organization

A **persistent desktop** is a vDesktop that is assigned to a specific end user and whose state is saved after logoff:



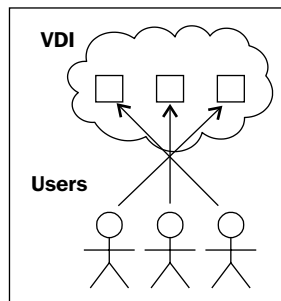
For example, in the preceding diagram, if User_1 signs into a VMware View environment for the first time and is automatically assigned vDesktop_7, he would connect to vDesktop_7 today, tomorrow, and until the assignment is removed. If vDesktop_7 has an issue and is unavailable, User_1 will not have a functioning work environment and will be unable to be productive.

When using persistent desktops, VMware View does not automatically reassign the end user to an available vDesktop if their assigned vDesktop is unavailable.

A persistent vDesktop has persistent data until:

- The vDesktop is unassigned from the specific user
- The desktop pool that the vDesktop is a member of is refreshed (linked clones)
- The desktop pool that the vDesktop is a member of is recomposed (linked clones)

A **non-persistent desktop** is a vDesktop that is not assigned to a specific end user and instead is made available to the end user population:



For example, in the preceding diagram, if User_1 signs into a VMware View environment, he is assigned one of the available vDesktops (for example, vDesktop_9). When he/she logs off and then logs back in to the VMware View environment, he/she is randomly assigned another available vDesktop from the pool.

A vDesktop may only have a maximum of one owner at any given time.

There are several settings that can be manipulated from the VMware View Admin console that dictate how fast a non-persistent vDesktop is unassigned from a user that executes a logoff.

Persistent desktops

Historically, persistent vDesktops have been used in VDI implementations as persistent vDesktops most resemble the physical world and allows both end users and IT administrators to be able to relate to the virtual world. This 1:1 relationship (of end user to vDesktop) is likely the most simplified deployment option available, but its design and cost considerations should be properly understood.

Persistent vDesktops are often the easier political sell to an organization, as each user still maintains possession of an individual (virtual) desktop asset. The following table explains areas related to the persistent vDesktops:

Area	Implications
Desktop pool sizing	It requires a vDesktop for every end user.
Availability	If a user's assigned vDesktop is unavailable, the user is unable to connect.
Recoverability – OS failure	VMware HA can be used to monitor a vDesktop's response to a basic ping command. If a vDesktop does not respond, it can be automatically restarted by VMware vSphere.
Recoverability – site failure	There is no easy way to recover from a site failure.
Cost	The VDI must support vDesktops for the entire target user population; this includes compute and storage requirements.

Example scenario

For this example, consider that Customer_A has 6,000 end users. They are targeting to move to a VMware View solution. Customer_A operates three shifts a day, with 2,000 end users working at any given time.

The customer has asked for help in scoping the hardware required for the solution (for example, in the form of a bill of materials or list of materials).

The platform will be Windows 7, with 1 vCPU, 2 GB of RAM, and a 50 GB C:\ drive.

The hardware platform is determined to consist of 2U servers with two 6-core processors per server (a total of 12 cores per server).

Using an average and conservative estimate of 10 VMs per core, it yields 120 vDesktops per server. Using 2 GB of RAM per vDesktop, 240 GB of RAM would be required to support 120 vDesktops. Adding in RAM for overhead and for amounts that can be easily ordered from any vendor rounds 240 GB of RAM up to 256 GB of RAM per server.

Therefore, the server specification used for this solution includes a 2U server with two 6-core processors and 256 GB of RAM. The following table explains about the server specifications and the costs related to the persistent vDesktop:

Area	Description
Physical server specification	A 2U server with two 6-core processors and 256 GB of RAM
Number of vDesktops/physical servers	120
Total end user population	6,000
Total number of end users that require a vDesktop at any given time	2,000
Total number of vDesktops that must be provisioned and available	6,000
Total number of physical servers required without $n + 1$ considerations	50
Estimated number of racks required	3
Cost per individual physical server	\$40,000
Subtotal for physical server costs	\$2,000,000
Total number of processors	100
Estimated VMware vSphere View Premier per vDesktop	\$250
Subtotal for VMware View licensing	\$1,500,000
Total cost	\$3,500,000

Even though Customer_A only has 2,000 end users online at any given time, the fact that Customer_A has 6,000 unique end users means that to support this environment with persistent vDesktops, the VDI must have 6,000 vDesktops. It is possible to use extremely strict timeout values and logoff parameters to decrease the total number of supported vDesktops (perhaps from 6,000 to 5,000), but it does add risk to the overall solution and may or may not be feasible in a given environment.

Using rough estimates of \$40,000 per physical server, the estimated total cost, including server costs and rough estimates for VMware vSphere and VMware View Premiere (excluding bundle or add-on pricing), is \$3,500,000.

This estimate does not include financial considerations for port costs. For example, if each server only required two network connections (for example, using 10GbE), then an additional 99 switch ports would be required for the production network connections and a single out-of-band management connection (for example, HP ILO).

This estimate does not include financial considerations for additional quantities of VMware vCenter Server Heartbeat, used to protect the VMware vCenter Servers in a VMware View environment. As a large VMware View solution will require additional VMware vCenter Servers likely protected by VMware vCenter Server Heartbeat (approximately \$15,000 per vCenter Server).

This estimate also does not include financial considerations for cooling and power costs associated with the servers.

Operationally, it's also harder to manage from a human resources perspective. As people leave and enter the company, vDesktop sprawl could potentially eat up resources, as there is no easy way to track and maintain the User Data Disks in conjunction with user accounts.

A final consideration is the amount of physical U-space the persistent solution requires. For environments that require minimal footprint (for example, tactical installation), every U is of significance.

In summary, persistent desktops are likely to be an inefficient solution for most environments, especially those of significant scale.

Non-persistent desktops

Non-persistent vDesktop solutions are starting to become more commonplace as VDI is not only adopted more within the industry but also implemented at a larger scale. As persistent vDesktops often require vast amounts of additional resources, that is, both technical, human, and financial, this book will focus primarily on solutions leveraging non-persistent vDesktops.

While persistent vDesktops are still what most people think of when they think of VDI, it's important to demonstrate the advantages and cost savings realized by leveraging a non-persistent solution in certain situations.

VDI is less about ownership of a particular virtual machine in the data centre and more about the availability of a desktop resource when needed, which is customized for the user (via that user's specific profile). While there are plenty of solutions that do not require customization of the vDesktop (for example, kiosk solution in a hotel), a large percentage of VDI implementations will be for unique users with potentially unique desktops at an organization. The following table explains about the areas related to the non-persistent vDesktops:

Area	Implications
Desktop pool sizing	It requires a vDesktop for the maximum number of concurrent users.
Availability	A user will be able to connect to a vDesktop as long as there is an available vDesktop in the pool.
Recoverability – OS failure	A user will be able to connect to a vDesktop as long as there is an available vDesktop in the pool.
Recoverability – site failure	While it still isn't out of the box to recover from a site failure, a non-persistent vDesktop forces the user's persona to live outside the vDesktop, thereby making the replication and recovery of a viable vDesktop environment much easier.
Cost	The VDI must support vDesktops for the target maximum concurrent users; this includes compute and storage requirements for peak load and not for theoretical maximum, based on the total number of users.

Example scenario

To continue from the example earlier in this chapter, Customer_A has 6,000 end users. They are targeting to move to a VMware View solution. Customer_A operates three shifts a day, with 2,000 end users working at any given time. The following table explains about the server specifications and the costs related to the non-persistent vDesktops:

Area	Description
Physical server specification	A 2U server with two 6-core processors and 256 GB of RAM
Number of vDesktops per physical server	120
Total end user population	6,000
Total number of end users that require a vDesktop at any given time	2,000

Area	Description
Recoverability – site failure	2,000
Total number of physical servers required without $n + 1$ considerations	17
Estimated number of racks required	1
Cost per individual physical server	\$40,000
Subtotal for physical server costs	\$680,000
Total number of processors	34
Estimated VMware vSphere View Premier per vDesktop	\$250
Subtotal for VMware View licensing	\$500,000
Total cost	\$1,180,000

The advantages of a non-persistent solution are clearly evident. For example, in a persistent solution, 50 physical servers were required (versus 17 in a non-persistent solution) to meet the demands of the user load. By saving a significant number of physical servers from being procured (and the associated software licensing), the overall VDI solution requires one third less funding. These savings come from having to support only the maximum concurrent user load instead of having a named vDesktop for every user that will connect to the VDI.

Other non-persistent notes and considerations

A non-persistent vDesktop solution should be viewed as volatile. It means that when a user logs off, anything written to the local disk of the vDesktop will be inaccessible, or even destroyed.

As such, for solutions requiring a user to maintain a profile, custom applications, unique device mappings, and so on, a persona management solution will need to be used as part of the overall VDI. Persona management is covered later in this book.

Another potential benefit of non-persistent vDesktops, if designed correctly, is that they can potentially reduce the number of help desk calls. This is because the solution focuses less on the availability of an individual's vDesktop, but instead focuses on the availability of a set number of vDesktop resources available in a pool. With an emphasis placed on desktop pool health (and likely user persona health), there's less worrying about a specific user's vDesktop as resources are assigned randomly at login for each end user.

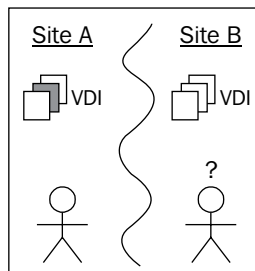
The potential drawback is that there could be an increase in supporting the persona management layer, depending on the solution chosen, its design, and its implementation.

Multisite solutions

A VMware View solution that spans multiple sites is not extremely atypical. College campuses, large corporations, and even small businesses, may have requirements to deliver vDesktops from more than one physical locations.

In these scenarios, there are a few qualifying questions to ask the organization. They are as follows:

1. Should the desktop experience be unique for each end user?
 - Essentially, should end user personas be saved to include changes to the desktop environment, customizations to applications, and so on?
2. If the answer to the preceding question is yes, should the user persona be consistent across all the sites?
 - For example, if Liliana logs into Site_A and makes modifications to her desktop, should those settings be reflected if Liliana were to log in to the VDI at Site_B?



The preceding diagram shows a multisite VDI with VMware View persistent vDesktops and it is based on a real-world scenario. In this example, Site_A and Site_B are owned by the same organization, Acme Corp. Acme must support worker mobility as two of their locations (Site_A and Site_B) are used by their entire workforce.

A user could be working at Site_A in the morning and then at Site_B in the afternoon for meetings.

The user, as shown in the preceding diagram, connects to the VDI in Site_A in the morning and as persistent vDesktops are being used, he gets his assigned vDesktop. Remember, when using persistent vDesktops, a user is assigned a specific vDesktop and will keep this assignment until unassigned by the VMware View Administrator.

When the user leaves Site_A and drives over to Site_B, the following two cases are possible:

1. He does not have a vDesktop assigned.
2. He has a second vDesktop assigned in the completely independent VDI running at Site_B.

Why is the user's vDesktop in Site_A not copied to Site_B?

This is because VMware View persistent vDesktops are independent virtual machines assigned to an individual user.

If using VMware View persistent vDesktops, replicating the changes to a peer persistent vDesktop in one site (for example, Site_A) to another site (for example, Site_B) is not an out of the box supported use case. Making persistent VMware View vDesktops behave in such a manner would require significant scripting, a deep understanding of the underlying storage platform, a deep understanding of the VMware View ADAM database, and likely an understanding of how to manipulate objects using PowerCLI.

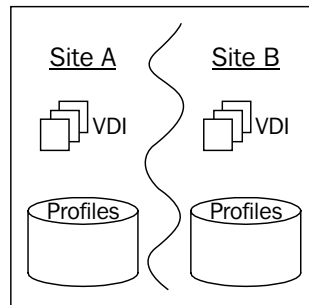
It would also add too many variables to make it a sustainable VDI model to properly support.

The authors' experience is that almost anything is possible with VMware View, given enough time, deep knowledge of VMware vSphere and VMware View, and ample time to test. However, VMware View was not designed to support multisite persistent vDesktops.

Third-party add-on solutions (for example, Unidesk) are potentially helpful in these scenarios.

Why is a non-persistent vDesktop best for a multisite?

The following diagram shows a multisite VDI with VMware View non-persistent vDesktops and persona replication:



Imagine the same aforementioned scenario. An organization has two sites, Site_A and Site_B. A user works at Site_A in the morning, connects to his/her vDesktop running on the VDI in Site_A, and then heads to Site_B for afternoon meetings. When at Site_B, the user connects to the VDI local to that site.

How can the VDI architect make the VDI experience consistent across both the sites?

By using non-persistent vDesktops, the user's persona is naturally separated from the underlying desktop operating environment. By using VMware View profile management, or a third-party solution such as AppSense or Liquidware Labs ProfileUnity, a non-persistent vDesktop can have the same look and feel of a persistent desktop (for example, customizations are retained) yet offer the advantages of non-persistent vDesktops (for example, greater flexibility).

In the preceding diagram, the user profiles are stored on a file share that is replicated between the two sites.

It does not matter which site's VDI the end user connects to as long as his/her profile has been replicated. If a non-persistent vDesktop solution with replicated profiles is implemented, it is important to ensure that there are no unnecessary files in the user persona. Proper filtering techniques can ensure that gigabytes of MP3s or downloaded movies are not considered as part of the user's persona, and prevent them from congesting the replication transmission.

Why distance matters

Replication is a function of physics. The size (throughput), speed (latency), and integrity (dropped packets) of the connection between sites as well as the amount of data that needs to be replicated help determine the total time required to replicate a set of profiles. If the goal is for users to not notice a difference whether they log in to Site_A or Site_B, the replicated copy of a given user's persona will have to be ensured.

This guarantee can only be provided if the underlying network, storage, and replication solution can meet the requirements.

Typically, the closer the two sites and the smaller the replication data set, the easier it is to meet these requirements.

Nevertheless, if a multisite VDI solution is to be implemented, it is important to perform a basic network survey to recognize any potential hurdles before actual implementation begins.

Profiles in the cloud

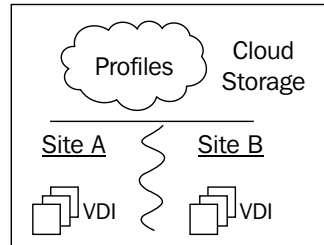
Multisite VDI solutions with local copies of the user profiles is likely the most common type of multisite VDI solution. This is because the technologies are likely already known by the VDI architect, and the supporting technical personnel are also familiar with similar solutions, even if not related to VDI in the least.

However, should there be a disruption or congestion in replication, it is possible for a user to log in at a site and not have their latest user persona. Even worse, it is possible to end up in a split brain scenario, where changes are lost or corrupted because two master copies of the user persona now exist.

While this is rare and can be combatted by proper replication design and monitoring of replication health, it is still possible.

Another type of multisite VDI solution still leverages non-persistent VMware View vDesktops and persona management, but instead of storing local replicated copies of the user personas, they are stored in the cloud.

The following diagram shows a multisite VDI with the VMware View non-persistent vDesktops and the cloud-based Persona storage:



The **cloud**, in this sense, is any external storage platform that maintains the user profiles. This could be an Amazon-based solution, a local cloud provider, or a peer's community cloud offering, just to name a few.

In this scenario, there is no cross-site replication because the user profiles are always read and written to the cloud-based storage platform.

The advantage is that replication issues are no longer a concern, but the drawback is that a connection to the cloud (internet, VPN, and so on) is required to load a user's profile. A lack of connectivity to the cloud storage means that user profiles cannot be read or saved.

In addition, most cloud providers charge a fee for inbound and/or outbound traffic. Exceptions to this rule (for example, Amazon Direct Connect) do exist however, and may come into play if deciding on a hosting partner.

Hybrid: persistent mixed with non-persistent

While oftentimes, the first VDI use case at an organization will be clearly persistent or non-persistent, as the VDI adoption increases, so does the amount of use cases that need to be supported.

For example, an organization can initially implement a VMware View solution to support their 250-seat classroom environment. This is likely a non-persistent solution. However, after seeing the benefits of VMware View for their classroom, the organization may decide to roll out VDI on a larger scale. Now instead of just supporting classrooms, the executive team has decided to adopt VDI to support their mobile lifestyle. In addition, the CEO has demanded that he needs to be able to use VDI from his Apple iPad.

In these types of situations, persistent vDesktops can make life a bit easier. There aren't any user profiles to worry about, per se, and the application distribution is conceptually the same as it is in the physical world. It may also be the easiest to troubleshoot for the executive team, as you know which executive has which desktop quite easily.

VMware View natively supports the ability to have both persistent and non-persistent desktop pools side by side. There are no real design considerations to be made other than those found in their respective solutions. Profile management can be applied across non-persistent and persistent vDesktops uniformly. One point of consideration is that in the above preceding example of a classroom and an executive team, there may be different support personnel responsible for each group. If that is indeed the case, then the appropriate permissions will need to be defined in the VMware View Admin console.

How to choose

Fortunately, with VMware View, both persistent and non-persistent vDesktops can be tested side by side to see which works best with an organization's requirements. However, when designing a solution for an organization, some guidelines can be followed to help choose between persistent or non-persistent vDesktops. The following are some questions with their suggested (not necessarily only) solution type. The suggestions assume that the reader will answer *yes* to the question:

- Will users be installing their own native applications? Persistent.
- Does the environment support a large percentage of shift workers? Non-persistent.
- Will application virtualization be used? Non-persistent.
- Will a roaming profile solution be used? Non-persistent.
- Will the solution support a disaster recovery event? Non-persistent.
- Will users be assigned their own specific desktop for application or operating system licensing restrictions? Persistent.

These high-level questions will help the architect steer the solution in the proper direction. It is important to keep in mind that that preceding suggestions are not steadfast answers. Finally, it is important to remember that persistent and non-persistent solutions can cohabit in the same VMware View environment.

Summary

The question of persistence is one of the cornerstone decisions of any VDI design. Persistence defines how volatile the vDesktops may be, how applications may be distributed, and the amount of underlying hardware required. For VDI architects, it is important to build a portfolio of proven designs. VDI is a complicated technology as there are a lot of moving parts. Reducing the number of variables for each project is important and this can be done by building some loose parameters. The following is an example mission statement taken from a real-world scenario:

"I'm the Director of IT for Acme and I have a 2,000-seat classroom environment I need to support, with frequent turnover."

A VDI architect's formula in this scenario may be:

- VMware View non-persistent vDesktops + VMware ThinApp + zero clients

By already having an idea of what the solution should look like, the architect can focus on some of the key variables as follows:

- How large is the desktop image?
- How will the applications be managed?

Building a VMware View solution from scratch for every project is not efficient and more error-prone than building a stable one from a few solid VMware View designs.

The next chapter discusses the end devices used to connect into the VMware View solution. End devices are another important part in a VDI solution as choosing the right end device can drastically improve the probability of success. Understanding the limitations of each end device type is important in choosing the right device for a given organization, and this will be discussed in the next chapter.

4

End Devices

VMware View is a solution that delivers a desktop experience to end devices, through the PCoIP protocol. The devices supported by VMware View vary greatly and include thick client, thin client, zero clients, and other devices, for example, the Apple iPad. This chapter will cover the various types of clients as well as the features that they do and do not support so that proper design considerations can be made in an overall solution.

Something to keep in mind when evaluating client devices for a VMware View solution is that most people associate the quality of their work computer with the size and appearance of their monitor.

For example, Jenson has a dusty HP workstation under his desktop connected to a 17-inch monitor. If the IT department replaces Jenson's dusty HP workstation with a zero client and 24-inch monitor, he is likely to already have a positive impression. Extend the positive perception by giving Jenson, who previously ran a Windows XP workstation, a Windows 7 vDesktop, and Jenson is likely to rave about his "new computer".

The device selection is important for the success of a VDI project as it is the gateway for users to connect to their vDesktops.

In this chapter, we will cover:

- Thick and thin clients
- Teradici PCoIP powered zero clients and other clients
- Choosing proper devices for your organization

Thick clients

A **thick client** is a laptop or desktop running a full version of a workstation operating system, for example, Windows XP. The thick client has a fully usable operating system and uses a natively-installed VMware View Client to connect to the VDI.

A few examples of thick clients are as follows:

- Dell OptiPlex workstation
- Lenovo laptop
- Apple MacBook Pro

The common advantages of thick clients are as follows:

- They support Local Mode (Windows-based thick clients only)
- Typically, they provide high performance (from a graphics offloading perspective)

The major difference between thick clients and thin clients is that Windows-based thick clients are able to support checked out Local Mode desktops. While Windows-based thin clients have the capability to support checked out Local Mode vDesktops, they are often locked down and don't have enough local storage to house the encrypted vDesktop image. Therefore, if there is a use case in a particular solution for end users in a disconnected state, those end users will typically need to be issued thick clients to meet the requirements of VMware View Local Mode.

In addition to supporting Local Mode, thick clients typically have extremely good performance as they are often ordered with more than 2 GB of RAM, a multicore processor and moderate-to-high performing local disk. With some entry-level thin clients, running multimedia applications may cause the thin client's CPU to reach 100 percent as it struggles to handle the multimedia redirection. Multimedia redirection causes the media to be rendered at the end device, placing importance on the end device's horsepower capabilities.

The drawbacks of thick clients are as follows:

- Another workstation to keep patched, maintained, and in compliance
- Potential of having data at the edge (as the device is still a fully functional desktop by itself)
- An additional operating system to license
- A volatile endpoint
- Higher target of theft (again, because the device is a fully functional device by itself)

- The users require additional training as they may be confused when they are navigating their vDesktop and when they are actually on the native OS

Oftentimes, during a migration to a virtual desktop solution from an existing physical desktop environment, the organization will be interested in phasing in thin or zero clients. This approach is common in organizations that have just recently purchased new thick clients (for example, workstations) and/or organizations that are not up for a refresh for several years to come. However, by phasing in thin or zero clients, the organization will still be forced to manage the underlying OS of the thick client.

In addition to forcing an organization to manage the underlying OS, the organization may (depending on their particular OS license agreement) be forced to license the OS on the thick client in addition to the OS of the virtual desktop. This could quickly increase the license count needed for an organization and increase the overall capital expenditure of the virtual desktop initiative.

Finally, a thick client is typically a machine that boots from a writeable partition. This means that the device may come up in a slightly different state on every boot. The consistency of a thin client or a zero client does not come into play, therefore, maintenance tasks such as reimaging thick clients are likely to be incurred and thus increase the operational expenses associated with each thick client.

Repurposing thick clients

Repurposing is the process of taking a thick client (for example, Dell OptiPlex workstation with Windows XP installed) and turning it into a purpose-built VMware View endpoint. Repurposing typically involves the following:

- Installing a streamlined operating system (such as Windows XP embedded or a Linux distribution)
- Installing little additional applications on the repurposed thick client other than the VMware View Client
- Preventing changes to the thick client's configuration

An alternative to reimaging thick clients with a new, purpose-built image, is to use a bootable solution. For example, by creating a Live CD with a Linux desktop operating system and the necessary components to run the VMware View Client, thick clients can simply be booted off of a CD (read-only). This ensures a uniform experience each and every time a user turns on his/her device. The newly released VMware View Client for Linux makes repurposing even more appealing for many organizations without the capital to invest in PCoIP zero clients, for example.

Repurposing thick clients is often viewed as a stop gap measure to a zero client implementation as well as a means of prolonging the life of aging desktop hardware.

Thin clients

Thin clients refer to purpose-built devices that run a streamlined OS (for example, **Microsoft Windows XP Embedded (XPe)**, SUSE Linux) meant to deliver a minimal desktop environment to the end user. From this minimal desktop environment, the end user launches the VMware View Client and is then connected to his/her virtual desktop. Thin clients often have a write-protected or write-filtered system partition.

A few examples of thin clients are as follows:

- ClearCube I8520
- Wyse R50
- Dell Latitude Mobile 13

The common advantages of thin clients are as follows:

- They provide a consistent desktop environment
- They allow third-party software (such as a VPN client) to be installed
- They have a smaller footprint than thick clients
- Typically, they consume less power than thick clients

As thin clients are typically locked-down OSs with a write-filter (not allowing any writes to the system partition, or only allowing writes in specific subdirectories), the end user will have a consistent experience during every boot process. This can help reduce operational expenses associated with patching, maintaining, and reimaging thick clients.

In addition, thin clients allow the installation of third-party software to create a streamlined OS image specific to an organization's needs.

Currently, zero clients do not support a **Virtual Private Network (VPN)** connection as there is no client built into the PCoIP firmware. However, VMware View 4.6 introduced the ability to use PCoIP through a security server, which is a preferred method of remote connectivity due to a better end user experience over a typical VPN connection.

For organizations that want to lockdown the end devices, provide a consistent experience, and require a VPN connection, thin clients are a solid choice for an end device.

The drawbacks of thin clients are as follows:

- Cost
- Performance
- Potential vendor lock-in for upgrades (for example, Wyse)

While thin clients often have less features and a lower specification in terms of the hardware platform, their cost is typically equal to that of thick clients for most organizations. This is partly due to the high volume orders typically associated with thick client purchases and the healthy discounts that go with such an order. This is also due to the fact that thin clients are still a lower volume business overall and the manufacturing costs are typically higher than that of traditional thick clients.

Also, as thin clients are often trying to use and deliver the minimum possible for the end user to have a favorable experience (and ultimately connect back to a virtual desktop with sufficient horsepower for a user's needs), the overall experience can suffer if the thin client's processor can't handle multimedia redirection (for example).

Teradici PCoIP-powered zero clients

Zero clients refer to a device with an embedded Teradici PCoIP chip that allows the device to immediately boot to the VMware View Client without the need for an underlying OS. Zero clients do not have a writeable system partition, nor a hard disk for that matter. Instead, zero clients boot off of a chipset embedded into the device.

This section specifically highlights **Teradici PCoIP-powered** devices as it is the author's opinion that other zero clients on the market that do not adopt the PCoIP model are destined for obsolescence. At the very least, they will likely not have the same levels of adoption within the VMware ecosystem.

A few examples of zero clients are as follows:

- Samsung NC240 monitor
- EVGA PD02
- ClearCube I9424
- Wyse P20

The common advantages of zero clients are as follows:

- Security
- Ease of configuration
- Cost

- Vendor diagnostics for management
- Often a smaller footprint than thin clients
- Often less power consumption than thin clients

Zero clients are the most secure end devices an organization can leverage for their VMware View solution as there is no hard disk inside a zero client. In a zero client-based solution, there is no chance for sensitive data to reside on an end device because the end device is not capable of storing such data.

Zero clients are often viewed as on a par or easier than thin clients to configure. This is because a zero client has very few settings (IP settings, VMware View Connection Server, and other minor settings) as the goal is to get the end users to their entitled virtual desktop as fast as possible.

Finally, zero clients are often viewed as not only the most secure but also the most affordable solution. This is because, for organizations looking to replace their desktops, for example, a Samsung NC240 with a keyboard and mouse is often significantly cheaper than a thin client (which still needs a monitor). For organizations looking to leverage existing monitors, a standard zero client (for example, EVGA PD02) may be one of the cheapest options on the market.

The drawbacks of zero clients are as follows:

- Currently, they do not support VPN inside the client
- They do not support Local Mode
- Currently, they do not offer Wi-Fi support natively
- No client side caching

For organizations that need their end devices to establish a VPN connection in order to connect to the VMware View Connection Server environment, a zero client will not suffice. This is because zero clients currently do not have a VPN client embedded into the firmware and have no means of establishing a secure connection to a remote peer.

It should be noted that a proper VMware View solution leveraging a security server should negate the need of a VPN connection.

Also, zero clients have no means to support a virtual desktop running natively in Local Mode. There are no mobile-centric (for example, a zero client in a laptop housing) zero client solutions today, so mobile users needing to leverage Local Mode may still be best served by a traditional thick client.

Other clients

Other clients can include items such as an Apple iPhone, Asus Transformer (Android-based tablet), Apple iPad, or a Dell Streak. The most popular peripherals right now are tablet-based end devices, primarily the Apple iPad and Android-based tablets.

The drawback of these other clients is that they may not support Local Mode. Also, devices, for example, the Apple iPad do not support Local Mode, meaning that when the end user is out-of-band (for example, on an airplane without Wi-Fi), the user will be unable to access his/her desktop.

For organizations deploying tablets as the end device, it is strongly encouraged to provide a keyboard solution with the tablets. Today, most of the users are native users of conventional keyboard and mouse solutions. Tablet devices offer a new type of input that many users are still growing accustomed to. While the user experience is often enjoyable, trying to type a long e-mail or perform some of the regular work tasks in certain organizations, may prove tedious.

The VMware View Client for the iPad, for example, supports Bluetooth keyboards, such as the Zagg Mate (<http://www.zagg.com/accessories/logitech-ipad-2-keyboard-case>).

Choosing the proper device

The following questionnaire can be used to help determine the right device or devices for a given organization. This questionnaire is based off of experience and should be used as a foundation when out in the field:

Question	Answer	Reason
Will new devices be purchased for this VDI project?	Yes	Zero clients as primary with other devices/tablets as secondary.
Will new devices be purchased for this VDI project?	No	Repurpose existing thick or thin clients.
Is security of paramount importance?	Yes	Zero clients as they do not offer a writeable hard drive in the unit.
Will the solution need to support regular video conferencing?	Yes	Investigate one of the upcoming integrated PCoIP zero client solutions from companies like Cisco.

Question	Answer	Reason
Will smartcard authentication be required?	Yes	Thick, thin, or zero clients will suffice; avoid tablets.
Is a minimal footprint and cable infrastructure desired?	Yes	Samsung integrated monitor zero client, with a third-party Wi-Fi solution, and wireless keyboard and mouse. It requires one cable (power) from the solution to the wall.
Are the majority of users going to be connecting from a fixed position?	Yes	Zero client.
Are the majority of users mobile or road-warriors?	Yes	Thick client (laptop).

The preceding table is simply a starting point. Ideally, a couple of devices can be tested during the pilot, to include a thick and zero client. This will help an organization understand how a zero client could work for them (or potentially not work in unique circumstances).

A one-cable zero client solution

Cable reduction is a topic that comes up often in education environments, or any environment, where the client environment must be built and torn down quickly. A typical physical desktop solution includes:

- Power cable for the workstation
- Network cable for the workstation
- USB cable for the keyboard
- USB cable for the mouse
- Video cable
- Power cable for the monitor

This solution equals a total of six cables to have sprawled across a desk or conference room. It also includes three cables (monitor power, workstation power, and workstation network) that need to connect to a wall jack.

The following solution requires just one cable to the wall and is based off of the integrated PCoIP monitor solution from Samsung (NC190 or NC240):

- Power cable for the Samsung monitor zero client
- NETGEAR WNCE2001 Universal Wi-Fi Internet Adapter
 - This allows the zero client to connect to the VDI over Wi-Fi without requiring a patch cable to run from the device to a wall jack
 - The WNCE2001 can also be powered by USB (to remove a power cable requirement)
- Logitech MK520 wireless keyboard and mouse
 - The transmitter occupies only one USB port and provides a wireless keyboard and mouse

This solution requires exactly one power cable and can be used in environments that require quick build up and tear down such as emergency response, training, and educational environments.

Summary

VMware View is an extremely flexible solution that can support a wide variety of end devices. This helps increase the overall success of a VDI initiative because it allows users to bring the device that works best for them in a given situation or state of connectivity. For example, a user can simply travel with an Apple iPad and use the VMware View Client to connect into their corporate vDesktop to perform light tasks, such as internal browsing, file management, and so on. When the user is back at his/her home office or his/her desk at work, they may have a full-blown desktop with the View Client to work from; allowing faster input, for example.

From the collective experiences of the authors of this book, PCoIP zero clients are often the best way to move forward for an organization, as it removes an unnecessary variable from the overall VDI solution. PCoIP zero clients are dependable, predictable, affordable, and secure.

The next chapter discusses how to properly size the VMware View solution, which is important to ensure a positive end user experience, redundancy, and cost effectiveness. Now that the ground work has been laid for VMware View, it is time to jump in and start designing the software and hardware infrastructure.

5

The PCoIP Protocol

The PCoIP protocol, developed by Teradici and licensed by VMware, is a purpose-built protocol for virtual desktop solutions on both LAN and WAN connections. PCoIP is a content-aware protocol, meaning that it has algorithms to differentiate between text and high-resolution pictures, for example, and then perform delivery optimization depending on real-time network characteristics.

VMware's own testing has shown that PCoIP can reduce display latency by more than 50 percent as compared to **Microsoft's Remote Display Protocol (RDP)** for common operations (VMware View PCoIP Network Sizing Guide).

In this chapter, we will cover the following topics:

- Why lossless quality is important
- Various PCoIP network fundamentals
- Multimedia redirection
- Teradici APEX offload card

PCoIP has many differentiators when compared to other protocols in competing VDI solutions; one such differentiator is the fact that the PCoIP is a host-rendered technology. Host rendering means that all pixels are rendered in the data center and then simply broadcasted to the end device. This means that there are no codecs to install at the end device.

One of the PCoIP features most often touted is the fact that the protocol can build to a lossless quality.

Consider that an end user is connecting to his/her vDesktop over a latent connection and attempting to render a web page. The web page consists of both high-resolution graphics as well as text. Notice that, initially, the text is crystal clear while the graphics are significantly compressed to conserve bandwidth.

Assuming that the display isn't changed by the user navigating to a different web page (for example), the visual will build to a perceptually lossless quality. This means that the human eye cannot tell the difference between what's displayed and the original version (with a higher number of pixels) rendered by the VDI. If there is still time before the user changes what they are trying to view, PCoIP will finally build to a fully lossless quality.

Why lossless quality is important

To understand why lossless representation is important, let's use an example. Whitney is a security agent tasked with screening packages that are entering a building. Her organization has implemented a VDI solution at her primary workstation. As packages enter the X-ray machine and the contents are displayed on her monitor, she has two seconds, as per the agency's policy, to determine whether it's a threat or not. If Whitney's agency is using a solution that is using a protocol that can't guarantee a lossless image at the end device, the visual representation on her screen may or may not be completely accurate. In this scenario, a lossless image delivered by PCoIP is of much greater value than a solution leveraging a solution and protocol that may have significant compression and may or may not be an exact visual representation of the virtual desktop.

Another example of where a lossless image is of critical importance is in a healthcare environment. For example, if VDI has been implemented at a hospital, where the clinicians are accessing their vDesktops from laptops, Apple iPads, and other end devices, it's important that the image being delivered is lossless. Without a solution capable of lossless rendering, a clinician could be looking at a **positron emission tomography (PET)** scan of a patient and be unable to determine whether the image suggests a particular diagnosis, for example, cancer, or not.

PCoIP network fundamentals

In order to understand how to size a network for PCoIP session delivery, it's important to know some of the key configurations and concepts of PCoIP. For example, the PCoIP protocol adds minimal overhead, with just 85 bytes of overhead in a standard 1,500 byte Ethernet packet.

For a PCoIP session to be established, a PCoIP-capable client must reside on the end device and the destination must be a PCoIP-capable host.

PCoIP-capable clients include:

- VMware View Client for 64-bit Windows
- VMware View Client for Linux
- VMware View Client for Mac
- VMware View Client for Apple iPad
- VMware View Client for Android

PCoIP-capable hosts include:

- Windows-based desktop operating system running the VMware View Agent software (physical or virtual)
- Windows-based desktop operating system with a PCoIP hardware host card (physical)

While it is possible to use PCoIP to connect to a Windows-based server operating system, it is currently not supported by VMware. However, the following URL will provide instructions on how to enable this solution:

<http://myvirtualcloud.net/?p=2811>.


The two types of PCoIP connections

There are two types of PCoIP connections – soft and hard.

Soft PCoIP is used when connecting to VMware View vDesktops. As a vDesktop cannot have a PCoIP host card, it uses a software implementation of PCoIP and is referred to as a **PCoIP software host**. Soft PCoIP can tolerate up to 250 milliseconds of round trip latency and is capable of displaying video at 30 **frames per second (FPS)**.

Hard PCoIP is used when connecting to a physical device with a Teradici PCoIP host card. As the connection is terminating at the PCoIP host card device, this is called **hard PCoIP host**. Hard PCoIP can tolerate up to 150 milliseconds of round trip latency. Hard PCoIP is capable of displaying video at 60 FPS.

Type of PCoIP	Round trip latency tolerance	Maximum frames per second for video
Soft	250	30
Hard	150	60

 To test network latency, use the ping `-l 1400 <destination_ip>`, where `<destination_ip>` is the IP address of the remote location. The `-l 1400` switch forces the ping test to use a packet size of 1400 bytes, which is the Teradici recommendation for testing network latency for PCoIP.

Both soft and hard PCoIP leverage local cursor technology, which ensures that cursor functionality for the end user is still favorable in high latency situations.

Multimedia redirection

Multimedia redirection (MMR) is the process of redirecting a media file from the PCoIP host (typically a VMware View vDesktop) to the end device. The more typical approach, known as **host video decoding**, is common practice with VMware View solutions.

MMR is only capable when the end device is an x86 client.

Additionally, the x86 end device must also have the appropriate codecs installed to support the type of media file being redirected. MMR is a technique that originally came to market years ago to support terminal services. Years ago, thin clients were gaining in popularity, primarily through the efforts of companies such as Wyse. As learned in *Chapter 4, End Devices*, thin clients have a locked down version of an operating system, for example, Windows XPe.

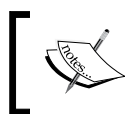
Media file types supported by MMR with PCoIP include:

- MPEG-1
- MPEG-2
- MPEG-4
- WMA
- MP3
- AC3
- WMV

PCoIP MMR does not support the redirection of Adobe Flash or Apple QuickTime. MMR does offer advantages by placing less of a demand on the server CPU hosting the vDesktops as the rendering of the media is done by one or more end device's CPUs. In addition, MMR can potentially require less network bandwidth as already-rendered visual data is not sent to the end device, but instead the media file to be rendered.

There are many disadvantages of MMR. For example, to use MMR an x86 end device must be used. As previously discussed in this book, there are many advantages (for example, price and security) to using PCoIP zero clients in a VMware View solution. By using MMR in a solution, thin or thick clients become the only available options.

For solutions looking to support video editors or video editing software, a hard PCoIP solution is quite likely the best (and only) viable solution. While the PCoIP protocol has made significant improvements since its initial launch years ago, a hardware-based PCoIP solution is the best approach.



PCoIP is a unique desktop delivery protocol in that it is not only available via software (soft). PCoIP has the ability to leverage the advantages of hardware in the form of a host card, for example.

The following table is an excerpt from a Teradici virtual desktop host presentation:

Description	MMR	Host video decoding
Server CPU load	Medium	Medium to high
Support any video codec	No	Yes
Support PCoIP zero clients	No	Yes
Requires application and patch management	Yes	No
Requires codec and patch management	Yes	No
WAN performance	Poor	Good
Operation below the native video bit rate	Video stutter	Smooth playback

As server CPU power and density increases, host video decoding will only become less concerning as the available horsepower in a given physical server increases with technology advancements.

The MMR perfect storm

One of the only legitimate reasons to use MMR in a VMware View solution is if there are requirements for frequent video use with a very specific codec. For example, if a public relations company watches videos of their clients frequently and the video is delivered in a specific codec, such as **audio video interleave (AVI)** file using DivX, it is possible that MMR to a thin or thick client with the DivX codec installed will outperform a solution relying solely on PCoIP.

Most organizations rely less on files such as AVI with DivX and more on Adobe Flash and Apple QuickTime, which will perform best with host video decoding.

Teradici APEX offload card

In 2011, Teradici announced the Teradici APEX 2800 offload card. The APEX card is a PCIe card that is used to offload the PCoIP protocol encoding from the physical server's CPU (abstracted as vCPU within the vDesktop) to the offload card. This offloading is for video only and does not help offload the audio channels of PCoIP. The APEX offload card is an integrated solution with VMware View.

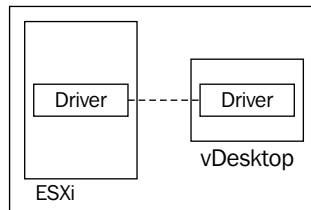
As mentioned earlier in this book, the fact that VMware View is capable of leveraging hardware solutions (for example, PCoIP zero clients, and PCoIP host card) provides unique capabilities in the market.

In many, if not most, VMware View solutions, the APEX offload card can be used to effectively increase the vDesktops that can be run on a physical server. Increasing this user per core density should measurably reduce overall costs of the VMware View solution, even with the price of the offload card built in.

There are three components to the Teradici APEX offload card solution:

1. The Teradici APEX offload card itself
2. APEX driver for ESXi
3. APEX driver for Windows vDesktop

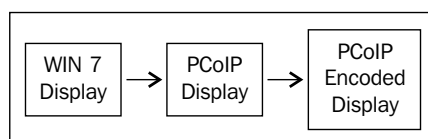
Without all three of the components installed and configured properly, hardware offload is not possible. The following diagram is an illustration showing both the required ESXi driver and Windows driver to support PCoIP offloading:



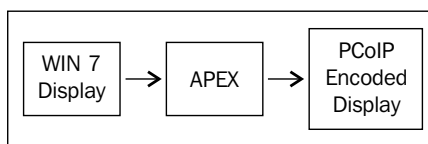
Hardware-assisted PCoIP protocol encoding is used to increase the number of users per core. Even in VMware View solutions that may be more memory constrained than CPU (for example, an environment heavy in Java applications with poor memory management), there can still be significant benefits realized from using the APEX card. As a general rule of thumb, the benefits are as follows:

- Task workers = 1.15x users per core
- Knowledge workers = 1.5x users per core
- Video workers = 1.75x users per core

For example, if a VMware View design that does not utilize the APEX card is capable of 10 knowledge works per CPU core, then using the APEX card will roughly increase the number to 15 knowledge workers per CPU. The following diagram is an illustration showing the encoding of a Windows 7 desktop with PCoIP and no encoding:



In the preceding illustration, the display of the Windows 7 desktop is encoded using the VMware View Agent, for example, and is done completely in software and by the virtual CPU of the vDesktop. The encoded display is then sent securely to the end device in use. With PCoIP, only the encoded display is sent to the device and no actual data of the desktop itself. The following diagram is an illustration showing the encoding of a Windows 7 desktop with PCoIP offload encoding:



In the preceding illustration, the display of the Windows 7 desktop is encoded using the APEX offload card. The encoded display is then sent securely to the end device in use. The offload card does not perform rendering, it performs encoding. Rendering is performed by the VMware View virtual graphics driver or virtual GPU.

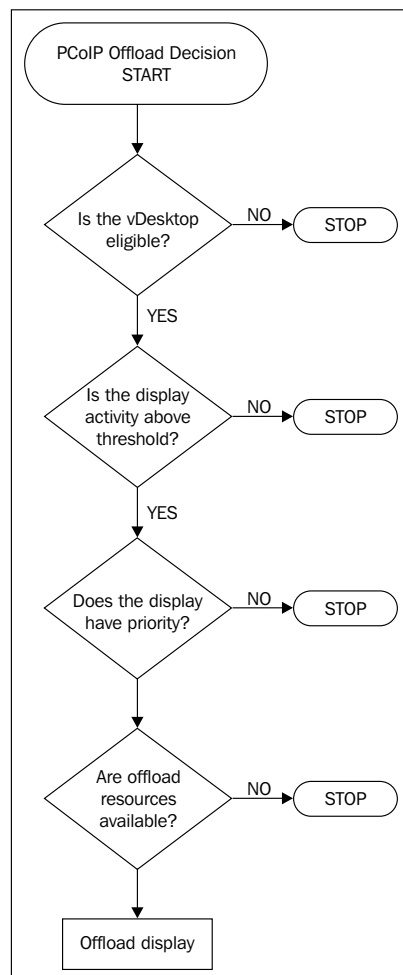
The current iteration of the APEX offload card can support encoding on up to 64 active displays simultaneously. This does not mean that a physical server cannot have more than 64 displays worth of vDesktops, but that only 64 displays can be offloaded at a given time.

The offload process

APEX 2800 ESXi driver monitors all vDesktops for image activity, whether the vDesktop is using hardware offloading or not. There are several factors that the APEX solution uses to determine if a vDesktop's display should be offloaded. They are as follows:

- **Eligibility:** Is the vDesktop eligible to have its display offloaded?
- **Imaging activity:** Is the vDesktop's imaging activity above the minimum threshold?
- **Priority:** Does the vDesktop currently have priority among its peers?

The following diagram shows a Teradici PCoIP APEX offload decision tree:



The switch between software and hardware encoding is a seamless process. In the current version of the solution, a tiny red dot appears in the upper-left corner of the display (can be disabled) to let the end user know that hardware encoding is being used. This should likely only be used during the test and pilot phase and not during an actual production implementation.

Defining the offload tiers

The Teradici APEX offload card uses priority as one of the factors in determining if a given vDesktop's display will be offloaded to the hardware card or not. Priority for a given desktop pool is defined within the VMware View Admin console under Policies. The offload priority is listed as PCoIP hardware acceleration priority. There are five priority settings available. They are as follows:

- Lowest
- Lower
- Medium (default)
- Higher
- Highest

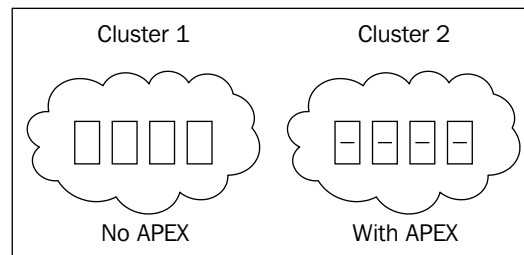
In many designs, only one or two offload priority settings will be used. For example, a design may have an end user population that is separated from the executive team. The executives may receive a "Higher" priority setting while everyone else is configured for "Medium".

It is best practice to use only "Highest" and "Lowest" if required by the number of priority groups defined for a given solution. By keeping "Highest" as an available option, if the need arises, a desktop pool can immediately be given priority over everyone else in an emergency.

Design considerations

The implementation of the APEX offload card is fairly straightforward; the driver installations are well-documented and enabling and configuring offload from the View Admin console is simple. The major consideration to make when using PCoIP hardware offloading is in regard to its capabilities across a cluster supporting a given desktop pool.

If a desktop pool's underlying vCenter cluster has both hosts with and without the APEX card, the vMotion (whether manual or triggered by DRS) could require a disconnect and reconnect to allow the PCoIP to initiate. The following diagram is an illustration showing one cluster without the APEX card and one cluster with the APEX card:



As vMotion task could prevent PCoIP offload from initiating, it is recommended to use the Teradici APEX offload card across all hosts in a given cluster. That's not to say that all hosts in a given VMware View solution need the Teradici APEX card, but if the card is to be leveraged, it should be leveraged cluster-wide.

Summary

As the outlined in this chapter, the Teradici PCoIP protocol is the technology behind the delivery of the desktop to the end device. Whether that end device is an Apple iPad or IBM laptop, PCoIP uses its intelligence to ensure that the best possible user experience is delivered. Understanding how the Teradici PCoIP protocol works is important in understanding the network requirements of a given VDI solution. In the next chapter, sizing the network will be discussed, which includes the discussion of PCoIP considerations. A working knowledge of the Teradici PCoIP protocol is a requirement for anyone designing a VMware View solution, otherwise, the network could become a limiting factor in an otherwise well-designed solution.

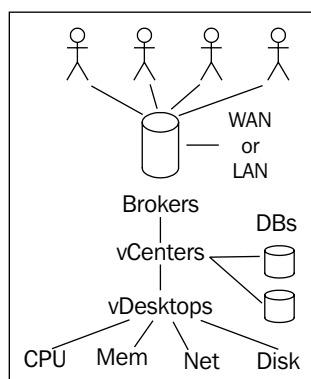
6

Sizing the VDI

Sizing is the process of determining how much horsepower any given component of the VDI solution requires, which is ideally based on metrics collected during the assessment phase. In most situations, the challenge will not be handling the average daily VDI workloads, but it will be handling the peaks. Peak loads in a VDI environment are often short in duration and may not be able to be mitigated through conventional techniques such as **VMware Distributed Resource Scheduler (DRS)** or manual vMotion balancing.

The components discussed in earlier chapters of this book, for example, VMware View Connection Server, require minimal sizing considerations when compared to the hardware components that must be sized. The reason being that the software components are primarily performing relatively lightweight work and merely brokering connections or performing provisioning tasks, which likely aren't happening constantly.

The following diagram shows the sizing layers of a VDI solution:



For example, having a properly sized and performing database infrastructure is important, as slow database response times can impact both View Composer tasks as well as tasks within vCenter. Also, it is important to ensure that the View Connection Server has adequate virtual or physical resources such as CPU and memory. However, the primary focus of this chapter is on sizing the physical components of the VDI.

To properly understand how to size a VDI, it's important to gather proper metrics during the assessment phase, which was covered in *Chapter 2, Solution Methodology*. Such metrics include the following:

- Number of concurrent users
- User classes and number of vCPUs, memory, and so on, per user class
- Network requirements
- USB redirection frequency

This chapter will focus on the following components from a sizing perspective, not necessarily from a redundancy perspective. This chapter is the n in $n + 1$. These components include:

- VMware View Connection Server
- VMware vCenter Server
- Server hardware
- Network infrastructure



Storage sizing is covered in *Chapter 8, Sizing the Storage*.



An improperly sized VDI could experience any of the following problems:

- Slow logons
- Poor PCoIP performance
- Inability to power on vDesktops due to reaching vCenter maximums
- Inability to log in to the VDI
- Authentication errors
- Random failures

Network considerations

While understanding the networking connectivity between the end users and the VDI is fairly obvious in a remote scenario, where the end user is removed geographically (for example, working from home) from the VDI, it's less obvious in a local scenario. While a local scenario may not blatantly cause a VDI architect to think about network sizing, it is still imperative to analyze and size the network component of a VDI solution even when all components reside on a **Local Area Network (LAN)**. This is the only way to truly confirm that the end user's experience should be as positive as possible.

Sizing the network

As a general rule of thumb, a typical task worker requires approximately 250 Kbps of network throughput for a positive end user experience. By generally accepted industry terms, a task worker is a user that has the following characteristics:

- He uses typical office applications or terminal windows
- He does not require multimedia
- He does not require 3D graphics
- He does not require bidirectional audio

However, where a task worker can potentially generate significant network bandwidth is with the use of USB peripherals. If a task worker requires USB peripherals to perform his job, then it is imperative to perform a network analysis of the specific USB peripherals in action prior to full-scale implementation.

The list of the consumables (Kbps) is as follows:

- PCoIP baseline = 250 Kbps
- PCoIP burst headroom = 500 Kbps
- Multimedia video = 1,024 Kbps
- 3D graphics = 10,240 Kbps
- 480p video = 1,024 Kbps
- 720p video = 4,096 Kbps
- 1080p video = 6,144 Kbps
- Bidirectional audio = 500 Kbps
- USB peripherals = 500 Kbps
- Stereo audio = 500 Kbps
- CD quality audio = 2,048 Kbps

The network checklist is given at <http://techsupport.teradici.com/ics/support/default.asp?deptID=15164>. But before that, you will be required to create an account at this site: <http://techsupport.teradici.com>.

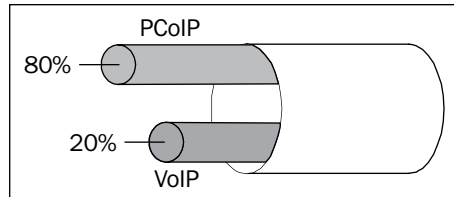
The other weights are as follows:

- Buffer = 80 percent
- Bandwidth offset = 105 percent

The minimum bandwidth to deliver acceptable performance is determined by the activity and requirements of the user's session. Some baseline numbers for the minimum bandwidth needed for a respective user type are as follows:

Description	Kbps
Office worker - low without multimedia	250
Office worker - high without multimedia	315
Office worker - low with multimedia	340
Office worker - high with multimedia	375

The following diagram is an illustration showing bandwidth provisioning of a given network connection:



In most environments, the only network traffic that should have a higher network priority than PCoIP is network traffic related to **Voice over IP (VoIP)** communications. Giving PCoIP a higher priority than VoIP could cause poor quality or loss of connections in certain environments with an improperly sized network. Therefore, it is recommended to give VoIP a higher priority than PCoIP (approximately up to 20 percent of the overall connection), give PCoIP traffic the second highest priority, and classify the remaining traffic appropriately.

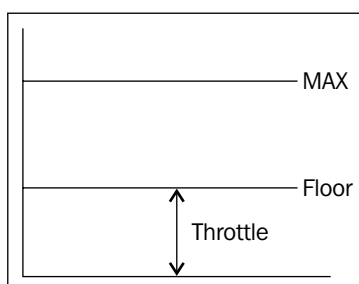
Network connection characteristics

Teradici has made significant improvements in the ability of the PCoIP protocol to handle high-latency and/or low-bandwidth scenarios. Teradici's PCoIP protocol is a purpose-built protocol for delivering a native desktop experience. In order to deliver the best possible end user experience, PCoIP will consume as much bandwidth as is available at a given time, up to the point where it can deliver a favorable end user experience. PCoIP is dynamic in nature, and as the available bandwidth changes, so does the amount of bandwidth that PCoIP attempts to consume. PCoIP initially uses **Transmission Control Protocol (TCP)** to establish the connection and then uses **User Datagram Protocol (UDP)** to transmit the desktop experience.

PCoIP also has two primary settings that should be understood, the PCoIP maximum bandwidth and the PCoIP bandwidth floor.

The PCoIP maximum bandwidth is the maximum amount of bandwidth a given PCoIP session is allowed to consume. Configuring this setting can ensure that end users never exceed a certain amount of bandwidth themselves. In addition, properly configuring the PCoIP maximum bandwidth provides a sense of insurance in a solution. Without limiting consumptions per session (even if the maximum is configured to be very generous) it is possible to have a runaway PCoIP session consuming a disproportionate amount of the available bandwidth. This disproportionate consumption could negatively impact the other users sharing the same network connection.

The following diagram is an illustration of the bandwidth floor and the bandwidth maximums:



The PCoIP bandwidth floor is the minimum threshold of bandwidth that must be available for PCoIP to throttle the stream. Following is an example:

An organization has 500 task workers and is looking to understand how large a network pipe they need to provide for their VMware View solution. The VDI users only use basic office applications and require no other capabilities.

Average Bandwidth Consumption = (Total Users * 250 Kbps) + (Special Need * Bandwidth Penalty) * Bandwidth Offsite * Buffer

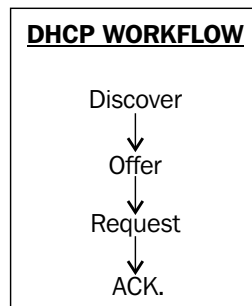
So, substituting the values given in the preceding example gives us the following output:

Average Bandwidth Consumption = (500 * 250 Kbps) + 0 * 80 percent = 100,000 KBps (approximately 97 Mbps)

DHCP considerations

While it is possible to cobble together a VDI solution that uses static **Internet Protocol (IP)** addresses, it is highly not recommended. Due to the potential volatility of a VDI and for ease of management, **Dynamic Host Configuration Protocol (DHCP)** is the preferred method for managing issuing the IP addresses of the vDesktops. When using DHCP, vDesktops do not own a specific IP address, but rather it leases it from a DHCP server.

A single DHCP scope consists of a pool of IP addresses on a particular subnet. A DHCP superscope allows a DHCP server to distribute IP addresses from more than one scope to devices on a single physical network. Proper subnetting can ensure that enough IP leases exist in a particular scope to serve the number of end devices requiring IP addresses. The following diagram shows a DHCP workflow:



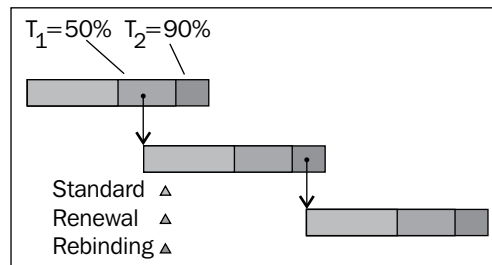
The workflow of a DHCP lease allocation is as follows:

1. The client broadcasts a DHCPDISCOVER message on its physical subnet.
2. Available DHCP servers on the subnet respond with an IP address by sending a DHCPOFFER packet back to the client.

3. A client replies with a DHCPREQUEST message to signal which DHCP server he/she accepted the DHCP OFFER packet from. The other DHCP servers withdraw their offer for a DHCP lease.
4. The DHCP server in the DHCPREQUEST message from the client replies with a DHCPACK packet to acknowledge the completion of the lease transaction.

DHCP reallocation occurs when a client that already has an address within a valid lease expiration window reboots or starts up after being shut down. When it starts back up, it will contact the DHCP server previously confirmed via a DHCPACK packet to verify the lease and obtain any necessary parameters.

After a set period of time (T_1) has elapsed since the original lease allocation, the client will attempt to renew the lease. If the client is unable to successfully renew the lease, it will enter the rebinding phase (starts at T_2). During the rebinding phase, it will attempt to obtain a lease from any available DHCP server.



In the preceding diagram, T_1 is defined as 50 percent of the lease duration and T_2 is defined as 90 percent of the lease duration.

For this example, assume a lease duration of 120 minutes (two hours).

A vDesktop boots and is successfully allocated a DHCP lease for a duration of 120 minutes from DHCP_SERVER_01. At T_1 (50 percent of 120 minutes, that is, 60 minutes), the vDesktop attempts to renew its lease from DHCP_SERVER_01. During the renewal period, the vDesktop successfully renews its DHCP lease. The lease clock is now reset back to a full 120 minute lease since the renew was successful.

This time the vDesktop is unsuccessful during the renewal period and enters the rebinding period. The vDesktop successfully obtains a new DHCP lease from DHCP_SERVER_03 with a lease of a fresh 120 minutes.

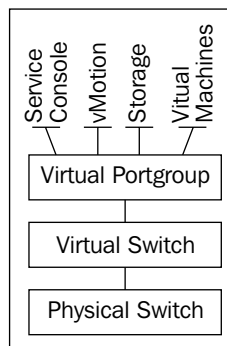
In most VDI scenarios, a DHCP lease time of one hour is sufficient. Typically, this is considerably less than the average DHCP lease time in default scopes used by most organizations.

If a desktop pool is set to delete a vDesktop after a user logs off, this could generate significant DHCP lease thrashing and a very short DHCP lease time should be considered (depending on the frequency of vDesktop deletions).

VMware View Composer tasks such as Recompose and Refresh should maintain the same MAC address throughout the process as the VMX settings related to the vNIC should not be altered. Therefore, the original lease would attempt to be reallocated during the boot process.

Virtual switch considerations

Virtual switch design for VDI environments is another component that may prove challenging for those unfamiliar with large-scale virtual infrastructure, or those accustomed to designing solutions with potentially high virtual machine volatility.



The preceding diagram shows, at a high level, the network components of a VDI environment. Not shown is the abstraction (that would reside in-between the physical switch and the virtual switch) done by the hypervisor.

When a standard vSwitch is created it has, by default, 120 ports. This parameter is defined at a kernel (hypervisor) layer, and any changes to the number of ports in a standard switch requires a reboot of the physical host.

When a distributed vSwitch, also known as a **dvSwitch**, is created, it has, by default, 128 ports. This parameter can be changed dynamically and does not require a reboot of the physical host for changing the number of ports from its original value of 128.

Standard versus distributed switches

Standard vSwitches are not impacted by the loss of a VMware vCenter Server, and are best used by functions such as Service console, vMotion, and storage connectivity as they can all be easily managed from the command line. However, in large VDI solutions leveraging multiple **Virtual Local Area Networks (VLANs)**, dozens or hundreds of physical hosts, dvSwitches help to greatly streamline the virtual network management across the virtual infrastructure.

VMware vSphere hosts keep a local cache of dvSwitch, dvPortGroup, and dvPort information to use when the VMware vCenter Server is unavailable. The local cache configuration copies are read-only and cannot be manipulated by the administrator.

Port binding

Port binding is the process of assigning a specific port, also known as a **dvPort**, to a specific **network interface controller (NIC)** on a specific virtual machine. Think of this assignment as analogous to taking a patch cable and plugging one end into the NIC on a physical desktop and the other end into an available switch. dvPorts decide how a virtual machine's network traffic is mapped to a specific distributed port group or **dvPortGroup**.

There are three types of port bindings used by dvSwitches; they are as follows:

- Static binding
- Dynamic binding
- Ephemeral binding

Static binding

Static binding assigns an available port on the dvPortGroup of the dvSwitch when a vNIC is added to a virtual machine. For example, if VM009 is a powered off Windows 2008 virtual machine and the administrator goes into Edit Settings and adds an NIC on dvPortGroup VLAN 71, a dvPort from the VLAN 71 dvPortGroup is assigned to the NIC, assuming one is available. It does not matter if the virtual machine VM009 is powered on or powered off, it is still assigned a dvPort and the dvPort will be unavailable to other virtual machines.

The assigned dvPort is released only when the virtual machine has been removed from the dvPortGroup. Virtual machines using static binding can only be connected to a dvPortGroup through the vCenter Server.

Advantage: The advantage of static binding is that a virtual machine can be powered on even if the vCenter Server is unavailable. In addition, network statistics are maintained after a vMotion event and after a power cycle of the virtual machine.

Disadvantage: The disadvantage of static binding is that the dvPortGroup cannot be overcommitted. In volatile VDI using non-persistent desktops that are deleted at the time of logoff, it is possible that the solution could run out of available dvPorts on the dvPortGroup. Static binding is strongly discouraged in environments leveraging VMware View Composer.

Dynamic binding

Dynamic binding assigns an available dvPort on the dvPortGroup when a virtual machine is powered on and its NIC is in the connected state. For example, if VM009 is a Windows 2008 virtual machine and the administrator goes into Edit Settings and adds an NIC on dvPortGroup VLAN 71, a dvPort from VLAN 71 dvPortGroup is not yet assigned. Once virtual machine VM009 is powered on, it is assigned a dvPort on the dvPortGroup and that specific dvPort will be unavailable to other virtual machines.

The assigned dvPort is released when the virtual machine has been powered down or the NIC is in the disconnected state. Virtual machines using dynamic binding can only be connected to a dvPortGroup through the vCenter Server.

Dynamic binding is useful in environments where there are more virtual machines than available dvPorts on a given dvPortGroup; however, the number of powered on virtual machines will not exceed the number of available dvPorts on a given dvPortGroup.

Advantage: The advantage of dynamic binding is that as a virtual machine doesn't occupy a dvPort until it is powered on, it is possible to overcommit the port on a given dvPortGroup. In addition, network statistics are maintained after a vMotion event.

Disadvantage: The disadvantage of dynamic binding is that as a virtual machine isn't assigned a dvPort until it is powered on, it must be powered on by the vCenter Server. Therefore, if the vCenter Server is unavailable, the virtual machine will not be able to be powered on. Network statistics are not maintained after the power cycle of a virtual machine as the dvPort is assigned at the time of boot.

Ephemeral binding

Ephemeral binding creates and assigns a dvPort on the dvPortGroup when a virtual machine is powered on and its NIC is in the connected state. For example, if VM009 is a Windows 2008 virtual machine and the administrator goes into Edit Settings and adds an NIC on dvPortGroup VLAN 71, a dvPort from VLAN71 dvPortGroup is not yet assigned. Once virtual machine VM009 is powered on, a dvPort is first created and then it is assigned a dvPort on the dvPortGroup and that specific dvPort will be unavailable to other virtual machines.

The assigned dvPort is released when the virtual machine has been powered down or the NIC is in the disconnected state. Virtual machines using ephemeral binding can be connected to a dvPortGroup through the vCenter Server or from ESX/ESXi. Therefore, if the vCenter Server is unavailable, the virtual machine network connections can still be managed.

When a virtual machine is vMotion'd, the original dvPort is deleted from the source dvPortGroup and a new dvPort is created on the destination dvPortGroup.

Ephemeral binding is useful in environments of high volatility, for example, a non-persistent VDI solution, where virtual machines are created and deleted often. The number of ports on a dvPortGroup is defined and limited by the number of ports available of the dvSwitch.

Advantage: The advantage of ephemeral binding is that as a virtual machine doesn't occupy a dvPort until it is powered on, it is possible to overcommit the port on a given dvPortGroup.

Disadvantage: Network statistics are not maintained after the power cycle of a virtual machine or after a vMotion event as the dvPort is created and assigned at the time of boot or vMotion.

Port binding and VMware View Composer

For VDI solutions leveraging View Composer, it is important to recognize that tasks such as Recompose, Rebalance, and Refresh will attempt to use the same port that has been assigned to the replica image.

Therefore, it is recommended to use dynamic or ephemeral (preferred) binding if VMware View Composer will be leveraged.

Compute considerations

Compute is typically not an area of failure in most VDI projects, but it is still important to understand the computing requirements of an organization before implementing a final design. Programs that can cause unforeseen failure from a compute perspective are as follows:

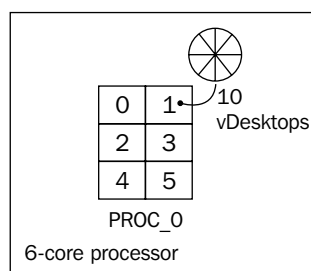
- Dragon Medical/Dragon Naturally Speaking
- Defense Connect Online
- AutoCAD
- Eclipse IDE

For VDI solutions that will be based on Windows XP, one vCPU can likely be used to address most basic computing needs. However, for VDI solutions leveraging Windows Vista, or more importantly Windows 7, two vCPUs may be necessary to ensure a favorable end user experience.

For the most accurate calculation of CPU requirements, a proper assessment of the environment should be performed. This will help identify potential pitfalls such as some of the applications listed previously, prior to rollout.

While both AMD and Intel-based x86 servers will suffice for VDI solutions, in large-scale and/or demanding environments, Intel-based solutions have consistently outperformed their AMD counterparts from a density (number of vDesktops per core) perspective.

In VDI solutions, there is also the potential for unnecessary CPU load due to tasks such as antivirus scanning, poorly-tuned applications, single-threaded processes, added visual effects, and impacts from video or audio processing.



The preceding diagram illustrates a single processor with 6 cores. As a safe baseline, 10 vDesktops per core is used for design purposes. For basic task workers, this number could be significantly higher, and there are multiple reference architectures that validate 15 to 18 vDesktops per core. The use of the Teradici APEX offload card could also increase users per core density.

Continuing to use 10 vDesktops per core as a baseline, and assuming that the server has 2 processors (of 6 cores each) that nets a total of 12 cores per physical server. With 12 cores per server and 10 users per core, that yields 120 users per physical server (6 cores per processor * 2 processors per server * 10 users per core). Using 1.5 GB of RAM for each vDesktop (the minimum recommendation for 64-bit Windows 7), the same physical server needs 180 GB of RAM (1.5 GB * 120 users). That's a relative sweet spot for memory, as most servers are configurable with 256 GB of RAM from the factory.

The following two tables have been extracted from the *Configuration Maximums*, VMware vSphere 5.0 guide at <http://www.vmware.com/pdf/vsphere5/r50/vsphere-50-configuration-maximums.pdf>. The tables explain compute maximums:

Host CPU maximums	Maximum
Logical CPUs per host	160

Virtual machine maximums	Maximum
Virtual machines per host	512
Virtual CPUs per host	2048
Virtual CPUs per core	25

Given the preceding information, we know that selected processors with significantly more cores per processor (for example, 24 cores per processor or 32 cores per processor) will not help vDesktop density on a given physical server.

The following table explains about memory maximums:

Host memory maximums	Maximum
RAM per host	2 TB
Maximum RAM allocated to service console	800 MB
Minimum RAM allocated to service console	272 MB

The reason why increased density will not be realized (or more accurately, maximized), is partly due to memory limitations and also due to existing VMware vSphere limitations. Let's assume, for the sake of argument, that a 32-core physical server was selected as the standard for a given VDI solution and it was shipped with a maximum supported 2 TB of RAM.

Using the conservative baseline of 10 vDesktops per core, that would yield 320 vDesktops per host, requiring 640 GB of RAM.

The following table explains about cluster maximums:

Cluster maximums	Maximum
Hosts per cluster	32
Virtual machines per cluster	3,000
Virtual machines per host	512

Comparing 320 vDesktops per host with the cluster maximums as defined by VMware, the maximum number of virtual machines per host would be reached.

Furthering the analysis from *Configuration Maximums, VMware vSphere* guide describes, "If more than one configuration option (such as, number of virtual machines, number of LUNs, number of vDS ports, and so on) are used at their maximum limit, some of the processes running on the host might run out of memory." Therefore, it is advised to avoid reaching the configuration maximums when possible.

As with all portions of a VDI design, it is important to leverage real-world metrics, when possible, to understand how vDesktops will be used, and how they will impact the underlying physical infrastructure.

Given the preceding calculations, it is advisable to conserve capital expenditure on high core count processors and instead focus the funding elsewhere. In most environments, six, eight, or twelve core processors will be more than sufficient in terms of performance as well as ensuring that vSphere maximums are not reached.

Working with VMware vSphere maximums

A strong case can be made that while VMware vSphere is by far the industry-leading hypervisor platform for server virtualization, its current maximums could be limiting in terms of mega-scale VDI environments. The following is a list of vCenter maximums taken from the *Configuration Maximums, VMware vSphere* guide that are most relevant to a VMware View solution:

vCenter Server scalability	Maximum
Hosts per vCenter Server	1,000
Powered on virtual machines per vCenter Server	10,000
Registered virtual machines per vCenter Server	15,000
Linked vCenter Servers (pod)	10
Hosts in linked vCenter Servers	3,000
Powered on virtual machines in linked vCenter Servers	30,000
Registered virtual machines in linked vCenter Servers	50,000

The preceding limitations will be analyzed with a solution example in the next section.

Solution example—25,000 seats of VMware View

A VDI architect has been hired by Company, Inc. to design a solution for 25,000 task workers in a single building. In this scenario, the networking and storage will be provided and will meet the necessary requirements of the VDI solution; therefore, the focus is on the physical server specification and the logical design of the VMware vSphere and VMware View environments.

Company, Inc. is looking for the following information:

- **Bill of materials (BOM)** for physical servers
- Logical design of the vSphere infrastructure
- Logical design of the View infrastructure

With a quick look at the requirements, the architect has determined the following:

- Powered on virtual machines per vCenter Server will be exceeded (limit 10,000)
- Registered virtual machines per vCenter Server will be exceeded (limit 15,000)
- Powered on virtual machines in linked vCenter Servers will not be exceeded (limit 30,000)
- Registered virtual machines in linked vCenter Servers will not be exceeded (limit 50,000)
- Maximum hosts per vCenter Server will not be exceeded (limit 1,000)
- Maximum virtual machines per host will not be exceeded (limit 320)

Solution design—physical server requirements

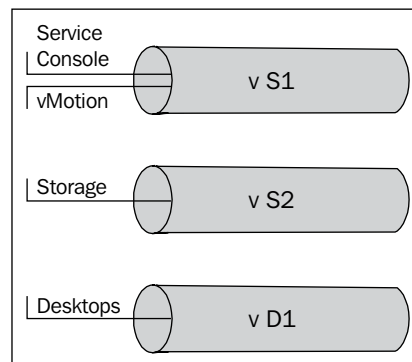
To support 25,000 task workers running Windows 7 vDesktops, the physical server sizing must be determined. Through initial testing, 10 vDesktops per core was a conservative estimate. As 4-core processors are being phased out, 6-core processors were chosen for their price and availability. Therefore, with 2 6-core processors per physical host, that yields 12 cores per host. Using 10 vDesktops per core and 12 cores per host yields 120 vDesktops per host. With 1.5 GB per vDesktop used for the environment, 180 GB of RAM is required for vDesktops. By allocating the maximum supported, 800 MB of RAM to the service console, that yields 181 GB of RAM required. Therefore, a server with 192 GB of RAM will support the environment nicely. In addition, the following vNetwork maximums exist:

vNetwork Standard & Distributed Switch	Maximum
Total virtual network ports per host	4,096
Maximum active ports per host	1,016
Distributed switches per vCenter	32

Given the preceding maximums, the following physical host design was leveraged:

Description	Value
Cores per processor	6
Processors per host	2
NICs per host	8
Memory per host (GB)	192
Approximate vDesktops per core	10
Approximate vDesktops per host	120
Standard vSwitches	2
Distributed vSwitches	1

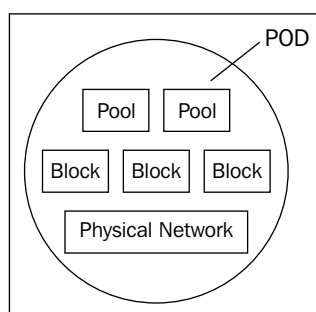
The networking configuration is as follows:



The preceding diagram represents two vNetwork standard switches and one vNetwork distributed switch. The first standard vSwitch, vS1, is used for service console and vMotion. The second standard vSwitch, vS2, is used for network-based storage. The only distributed vSwitch, vD1, is used for virtual machines.

Solution design—the pod concept

The concept of the pod is to give architects a method of creating building blocks to ease the design scalability for large environments. It also provides a conceptual framework for the solution architecture.



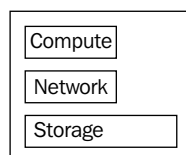
The main components of a VMware View pod are as follows:

- **Physical network:** This includes necessary switches, VLANs, network policies, and other network infrastructure required to support the VDI
- **vCenter blocks:** This includes hosts, vCenter cluster design, vCenter Linked Mode configuration, and so on
- **View Connection Server pools:** This includes View Connection Servers and (if applicable) View Security Servers

This concept of a pod can be carried through with the following architecture types:

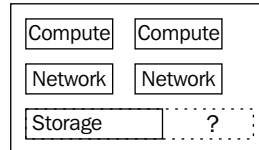
- Traditional
- Traditional in modular form
- Converged virtualization appliances

The **traditional** architecture type involves using servers (rackmount or blade), network switches, storage network switches (if applicable), and storage arrays. A traditional architecture approach is normally sufficient for an initial build-out but may not offer the scale-out capabilities of other approaches. The following diagram shows an illustration of a typical traditional architecture approach where disproportionate resources exist:



For example, using the preceding diagram, sufficient compute, network, and storage resources may exist for the initial rollout of 400 VMware View users. In this example, an overabundance of storage capacity exists.

The following diagram shows an illustration of a typical traditional architecture scale-out challenge:



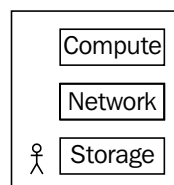
When the organization decides to add an additional 500 VMware View users, it runs into a problem. In the phase 1 rollout, an overabundance of storage capacity existed. However, to add capacity in a modular fashion, compute and network will still require an additional block of storage. Therefore, every addition will have some level of excess, which drives the price per vDesktop up due to architectural inefficiencies.

An organization likely would not want to accept these inefficiencies so it would redesign its requirements every step of the way. Designing the scale out for every additional phase of a VDI solution also drives up cost through added complexity and man hours.

In addition, every time a scale-out phase is re-architected, the chance of error becomes greater.

The **traditional in modular form** architecture type involves using servers (rackmount or blade), network switches, storage network switches (if applicable), and storage arrays. Whereas, a traditional architecture is normally not able to scale proportionately, a traditional in modular form is designed to scale in building blocks. This approach does not need re-engineering for each scale-out phase, and instead an organization relies on the traditional yet modular architecture for predictable scale-out design.

The following diagram shows an illustration of a typical traditional in modular form architecture approach, where proportionate resources exist:



There are typically two ways to implement a traditional in modular form architecture. The first is by spending the time to architect and test a customer design, where compute (for example, Dell blade) is combined with network switches (for example, Cisco) and a storage array (for example, NetApp). The danger with this approach is that if the person or team designing the solution has never designed a VDI solution before, they are likely to have a few lessons learned through the process that will yield a less than optimal solution. This is not to say that this approach is not suitable and should not be taken, but special considerations should be taken to ensure the architecture is sound and scalable. A seasoned VDI architect can take any off-the-shelf hardware and build a sustainable VDI architecture.

The second way to implement a traditional in modular form architecture is by implementing a branded solution such as the VCE Vblock (Cisco servers + Cisco switches + EMC storage) or FlexPod (Cisco servers + Cisco switches + NetApp storage), for example. These solutions are proven, scalable in a predictive manner, and they offer a known architecture for VDI. The drawback of these solutions is that they often have a high barrier to entry in terms of cost and scale out in large modular blocks (for example, 1,000 users at a time).

The third type of architecture uses **converged virtualization appliances**. Converged virtualization appliances are typically 2U to 6U appliances that comprise of one to many ESXi servers with local storage that is often shared among the ESXi servers in the appliance. The storage is typically shared through a virtual storage appliance model, where local storage is represented as either iSCSI or NFS storage to one or more ESXi servers in the appliance. The converged virtualization appliance model is relatively new to the VDI market.

Linked vCenter Servers

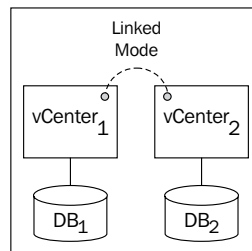
As the number of virtual machines per vCenter Server will be exceeded, more than one vCenter Server will be required for this solution.

The following table illustrates the vCenter maximums:

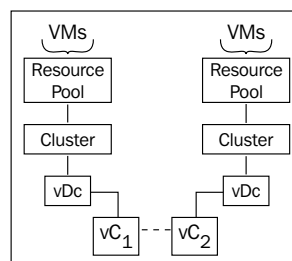
vCenter Server scalability	Maximum
Powered on virtual machines per vCenter Server	10,000
Registered virtual machines per vCenter Server	15,000
Linked vCenter Servers	10
Powered on virtual machines in linked vCenter Servers	30,000
Registered virtual machines in linked vCenter Servers	50,000

vCenter Linked Mode has a few basic prerequisites. They are as follows:

- Both vCenter Servers must reside in a functional DNS environment, where **fully qualified domain names (FQDNs)** of each vCenter Server can be resolved properly
- Any vCenter Server participating in Linked Mode must reside in an Active Directory domain
- If the vCenter Servers are in separate Active Directory domains, the respective domains must have a two-way trust
- Both vCenter Servers must reside in a functional **Network Time Protocol (NTP)** environment, where time synchronization of the vCenter Servers is no more than 5 minutes adrift of one another
- Windows RPC port mapper must be allowed to open the **Remote Procedure Call (RPC)** ports for replication; this is covered in detail at <http://support.microsoft.com/kb/154596>
- Both VMware vCenter Servers have the VMware vCenter Standard Edition license (versus foundation, for example)
- Separate databases for each VMware vCenter Server



VMware vCenter Linked Mode connects two or more vCenter Servers together via ADAM database replication to store information regarding user roles as well as VMware licensing. VMware vCenter Linked Mode does not do any form of database replication. If VMware vCenter Linked Mode would fail for any reason, the two (or more) vCenter Servers would still be viable as standalone instances.



As shown in the preceding diagram, where there are two separate vCenter Server instances (vCenter1 and vCenter2), the virtual data centers, clusters, resource pools, and virtual machines are unique to their respective instance of vCenter.

Joining multiple vCenters together with vCenter Linked Mode forms what is known as a pod. A pod can consist of up to 10 vCenter Servers in Linked Mode.

vCenter Servers

Using calculations from preceding sections, this solution is expected to have approximately 120 vDesktops per host; this means that 209 physical hosts are needed to support the vDesktop portion of this solution (not taking into account a virtualized vCenter, database, and so on).

Due to the nature of the end user population, the time they log in, the conservative nature of the original assessment (for example, 10 vDesktops per core), it has been decided that there will be no HA requirements for the vSphere Servers supporting vDesktops.

It has also been determined that the management infrastructure including the View Connection Servers, vCenter Servers, database server, and a few other components require three physical hosts. In order to provide a level of protection, it has been determined to use an $n + 1$ solution and utilize 4 physical hosts.

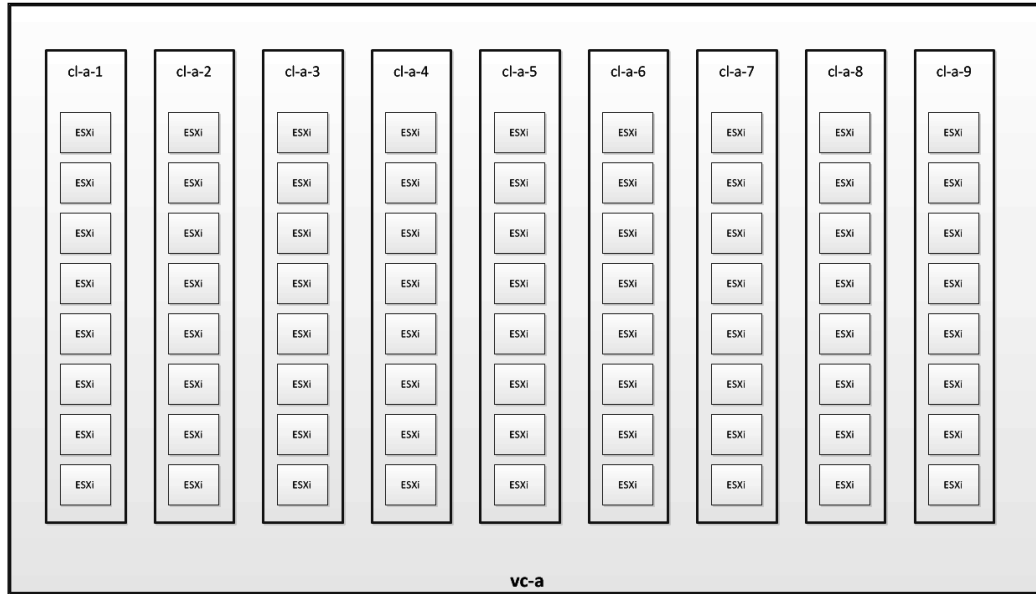
It was determined previously that any given vCenter can have a maximum of 10,000 powered on virtual machines at any given time. This solution will need to support more than 25,000 powered on virtual machines; therefore, this solution will require 3 vCenter Servers.

To balance the load across the vCenter Servers, the clusters have been as equitably divided as possible.

Note that the naming conventions used for the clusters in this example are:

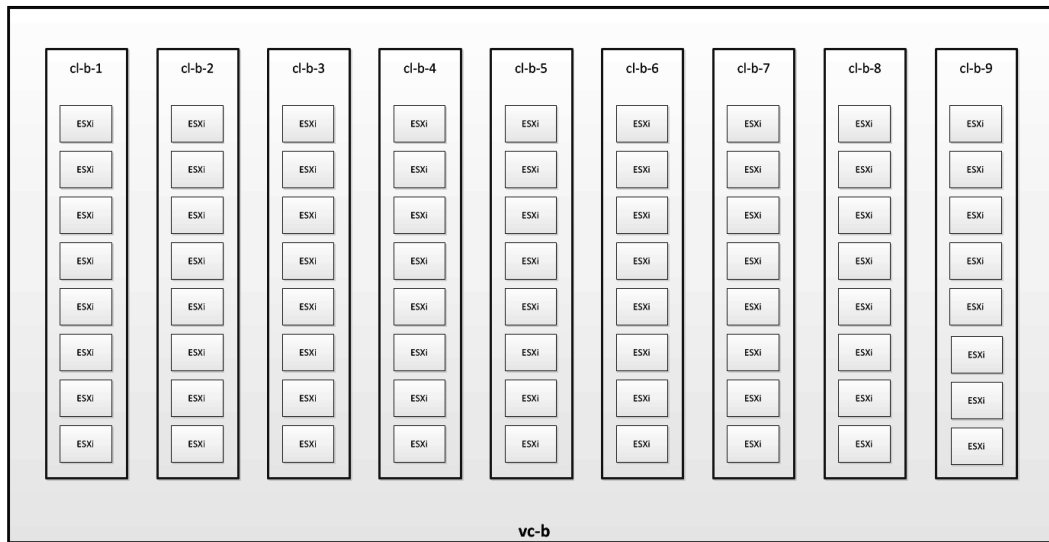
- vCenter Server: vc-{letter}, for example, vc-b
- Clusters: cl-{letter of vCenter}-{number}, for example, cl-c-6

The vCenter Servers are named **vc-a**, **vc-b**, **vc-c**, respectively. Their details along with the diagrams are as follows:



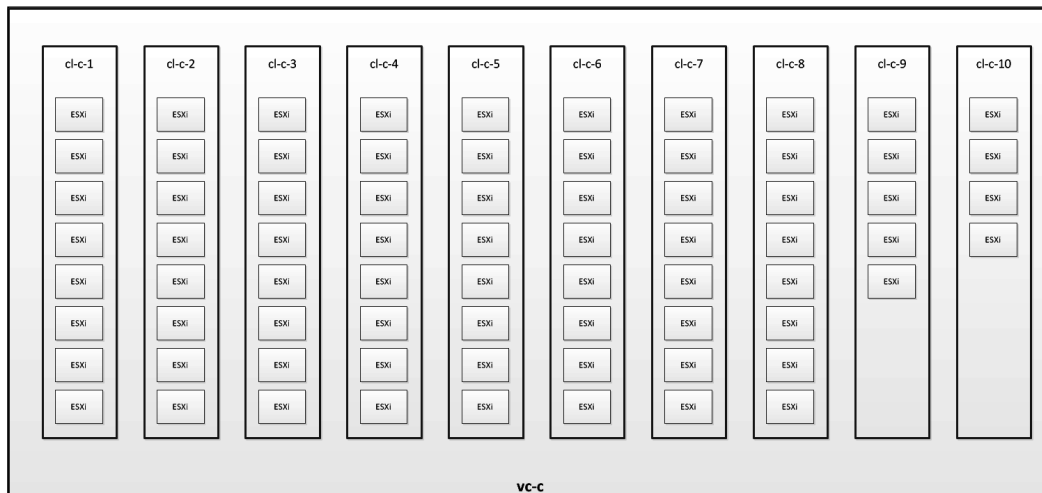
The preceding diagram explains about vCenter Server **vc-a**. The following list gives the details about **vc-a**:

- 9 clusters of 8 hosts each (**cl-a-1**, **cl-a-2**, ..., **cl-a-9**)
 - Total of 72 hosts
 - Total of 8,640 vDesktops (120 vDesktops per host multiplied by 72 hosts)



The preceding diagram explains about vCenter Server **vc-b**. The following list gives the details about **vc-b**:

- 9 clusters of 8 hosts each (cl-b-1, cl-b-2, ..., cl-b-9)
 - Total of 72 hosts
 - Total of 8,640 vDesktops (120 vDesktops per host multiplied by 72 hosts)

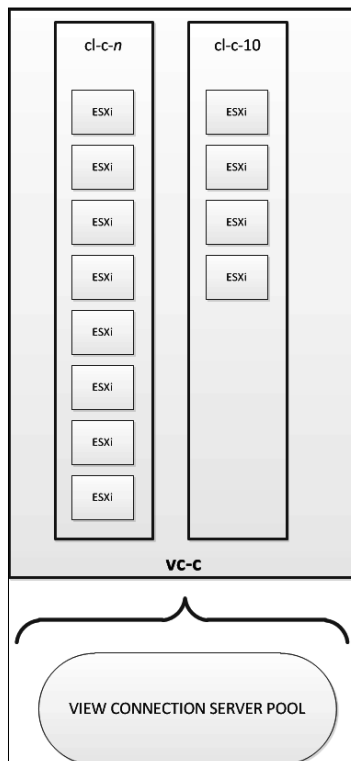


The preceding diagram explains about vCenter Server **vc-c**. The following list gives the details about vc-c:

- 7 clusters each having 8 hosts
- 1 cluster of 5 hosts
- 1 cluster of 4 hosts
- 1 cluster of 4 hosts dedicated to management
 - Total of 69 hosts
 - Total of 7,800 vDesktops and approximately 30 vServers (View Connection Server, database server, vCenter server, and so on)

vCenter vc-c has a cluster (cl-c-10) dedicated for hosting the infrastructure virtual machines. These virtual machines include:

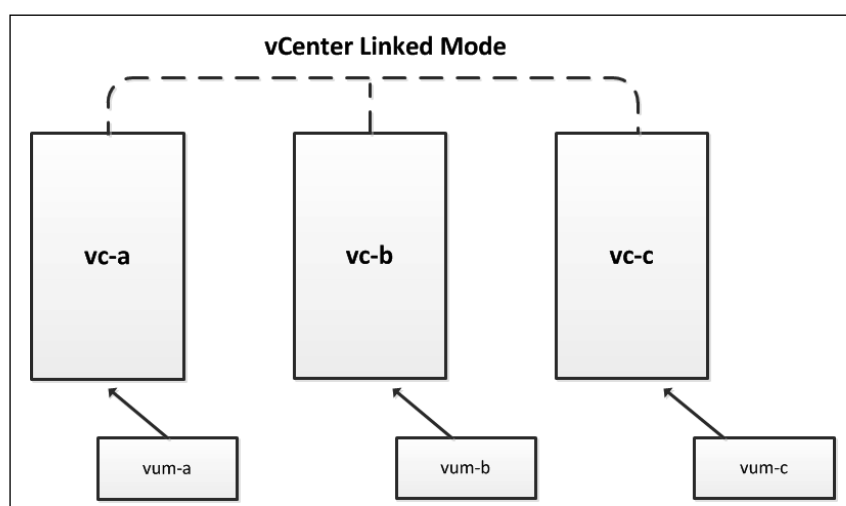
- 3 VMware vCenter Servers (vc-a, vc-b, vc-c)
- 15 View Connection Servers
- Supporting infrastructure (if needed) such as database servers, Liquidware Labs TM, and so on



VMware Update Manager Servers

VMware Update Manager is a solution that automated the application of patches to both vSphere Servers and virtual machines. It's most often used to patch vSphere Servers in large environments as it handles the task of placing a host in maintenance mode, migrating virtual machines, patch application, reboots, and normalization with a minimal amount of user interaction.

VMware Update Manager Servers can only be paired with one VMware vCenter Server instance at a time. Therefore, in this solution three VMware Update Manager Servers will be required (one per vCenter Server instance).



VMware vCenter Server Heartbeat

In this section, we have added a note about VMware vCenter Server Heartbeat. In most VMware View solutions, one or more highly available VMware vCenter Servers are required. vCenter Server is of paramount importance because if vCenter is unavailable, the following problems would be faced:

- New vDesktops cannot be provisioned
- vDesktops cannot be recomposed, refreshed, or rebalanced
- vDesktops cannot be deleted from the View Admin console

Therefore, vCenter Server Heartbeat is often an affordable insurance policy for the vCenter Servers in a VDI solution.

As noted previously, VMware Update Manager can only be linked to one instance of the VMware vCenter Server. However, it's important to note that a pair of vCenter Servers joined by VMware vCenter Server Heartbeat is considered to be only one instance.

Therefore, the solution does not require additional VMware Update Manager Servers just because VMware vCenter Server Heartbeat is being leveraged.

Solution design—pools

Here, we will cover View Connection Servers.

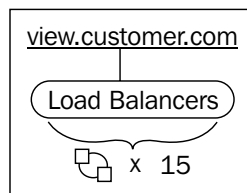
View Connection Servers

As illustrated next, the VMware View infrastructure introduces its own maximums in addition to those already imposed by the VMware vSphere infrastructure. The View Connection maximums are given in the following table:

Connection Servers per deployment	Maximum
1 Connection Server supporting direct RDP or PCoIP	2,000
7 Connection Servers (5 hot + 2 spare) supporting direct RDP or PCoIP	10,000
Maximum hosts in a cluster when not using View Composer	32
Maximum hosts in a cluster when using View Composer	8

If a solution like Unidesk TM was used in lieu of View Composer, the end design could support more hosts per cluster.

For the solution example, whereby 25,000 vDesktops must be supported, it's important to understand how many end users will be logging in at any given time. A VMware View Connection Server can support 2,000 direct PCoIP connections at any given time. In this example, all 25,000 end users could potentially log in at the same time. Therefore, a minimum of 13 View Connection Servers are required ($2,000 \times 13 = 26,000$ simultaneous direct PCoIP connections supported).



In order to provide a level of redundancy in case of a View Connection Server outage, it is advised to add in $n + 2$ (or more) solutions. For example, increasing the required number of View Connection Servers, that is, 13 to a total of 15 View Connection Servers, provides the ability to support a maximum of 30,000 simultaneous PCoIP connections. Therefore, even if two View Connection Servers fail, all 25,000 users would be able to log in to the VDI without incident.

The 15 View Connection Servers should be placed behind a redundant load balancing solution and should be configured to check that the View Connection Server is online via a simple ping (if **Internet Control Message Protocol (ICMP)** is allowed) and HTTP GET on the View Connection Server's URL. The entire pool of View Connection Servers should be accessible by a single name, such as `view.customer.com`, whereby end users would use `https://view.customer.com` to access the View environment.

By leveraging the HTTP GET to verify functionality of a View Connection Server, a server whose applicable services have stopped will not successfully reply to the GET command and, therefore, will be removed from the load balancing pool.

Solution design—the formulae

The following are some formulae to calculate the minimum number of vCenter Servers, Connection Servers, and Pods:

- Minimum number of vCenter Servers = Number of Desktops / 10,000
- Minimum number of View Connection Servers = Number of Simultaneous Connections / 2,000
- Minimum number of vCenter Pods = Number of vCenter Servers / 10

Summary

As detailed in this chapter, there are many design considerations to make such as DHCP lease time, the minimum number of vCenters, and the number of cores to buy in a server platform. For large environments of thousands of vDesktops, it may be easiest to start with the vSphere maximums and work down. For small environments or PoCs that don't require a massive virtual infrastructure, the concepts covered in this chapter are still relevant as a successful PoC can grow rapidly in adoption. Finally, the concept of a pod architecture, or a collection of vCenter Servers, is typically new to those familiar only with designing virtual server solutions on the VMware vSphere platform. They can take some time to understand the new concepts and working up against the vSphere and vCenter maximums.

7

Redundancy

When building a proper VDI, it's imperative to understand all of the possible failure points within the solution so redundancy can be built in to mitigate any failures. While sizing a VDI incorrectly will cause a slow response time or poor end user experience, failing to build proper redundancy could render the solution unreachable. In a VDI solution there are physical failure points to consider such as network switches, power supplies, and hard drives. There are also software failure points such as the VMware vCenter Server, VMware View Connection Server, and the database server(s) to take into consideration.

This chapter analyzes the potential points of failure within a VDI and offers up suggestions to provide redundancy for each component.

Physical infrastructure

In-depth coverage of designing a highly available virtual infrastructure is outside the scope of this book. However, understanding and utilizing the following VMware vSphere features are important to implement a robust VDI.

VMware High Availability

VMware High Availability (HA) can be used to monitor and protect against physical host failures and can also be used to monitor and protect vDesktops themselves. VMware HA works by monitoring physical hosts in a given cluster. If a host is unable to communicate to the specific default gateway on a service console interface for 15 continuous seconds, an HA failover event is triggered. vSphere 5 also introduced Datastore Heartbeating, which is used when a network heartbeat failure has occurred. Datastore heartbeats provide an additional level of host isolation verification.

For more information on VMware HA, please refer to the *HA Deepdive* article by Duncan Epping at <http://www.yellow-bricks.com/vmware-high-availability-deepdiv/>. You can also check it in his book *vSphere 5 Clustering Technical Deepdive*.

A host is determined to be isolated when a host has stopped receiving heartbeats from other hosts in the cluster and the specified isolation address cannot be pinged.

If the Isolation Response for the HA cluster is set to **Leave Power On**, the vDesktops and other virtual machines on the host will remain powered on. Just because a host has lost network connectivity on its service console interface does not necessarily mean that the vDesktops have lost network connectivity.

If the Isolation Response for the HA cluster is set to **Power Off**, the vDesktops and other virtual machines on the host will be powered off. This setting avoids the possibility of a split-brain scenario.

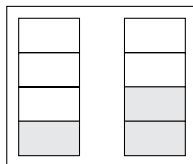
With the advancements in vSphere 5, the host isolation events are highly verified and accurate and very likely indicate an actual host problem. Therefore, in VMware View solutions, it's preferred to set the Isolation Response to **Power Off**.

If a specific host containing vDesktops has been isolated from the cluster, it will perform the following:

- All virtual machines and vDesktops will be powered off.
- Users with active connections will be disconnected from their vDesktops.
 - If the vDesktop is part of a persistent desktop pool, the user will be able to log back in once their specific vDesktop has been powered up and is online. The estimated outage time is 2 minutes.
 - If the vDesktop is part of a non-persistent desktop pool, the user will be able to log back in immediately to a vDesktop, assuming that there is an available vDesktop in the pool on a host that is currently online. The estimated outage time is less than or equal to 30 seconds.

Do you even need VMware HA?

VMware HA provides a level of protection for host failures, whereby vDesktops residing on a host that has failed will automatically be powered up without intervention from the virtual infrastructure administrator.



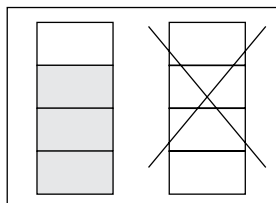
If admission control is set to **strict**, the vDesktops will only be powered on if there are available resources on another host in the cluster.

VMware HA works by determining the slot size, or minimum amount of CPU and memory to support a failover of the most intensive virtual machine (or vDesktop).

For example, if vDesktop_A has 4 GHz of CPU and 2 GB of RAM while vDesktop_B has 1 GHz of CPU and 6 GB of RAM, the slot size will be 4 GHz of CPU and 6 GB of RAM (with additional calculations taken into consideration for memory overhead).

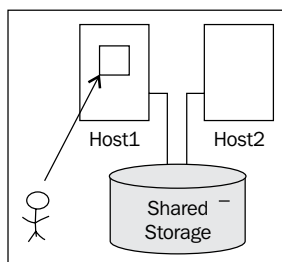
In VMware View environments, there will likely be large quantities of vDesktops with identical specifications (for example, a Windows XP vDesktop with 2 GHz of CPU and 2 GB of RAM), therefore a concept known as **slot fragmentation** is primarily avoided. Slot fragmentation requires the collective availability of sufficient resources in a cluster to support virtual machines being powered on during an HA event, but the lack of sufficient resources on an individual physical host to support a virtual machine's requirements.

For more information on slot fragmentation, please see Duncan Epping's very thorough article at the same URL: <http://www.yellow-bricks.com/vmware-high-availability-deepdiv/>, as mentioned earlier.

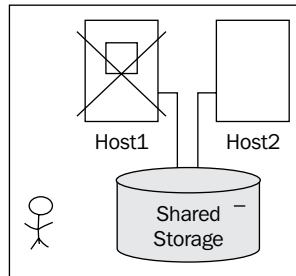


As of vSphere 4.1, HA also works in conjunction with DRS to free resource slots should slot fragmentation occur within a cluster. This would involve a failed server having to wait for virtual machines to be vMotion'd across hosts in the cluster until enough slots exist to power on necessary virtual machine(s).

With a VMware View solution based on persistent vDesktops, HA should be used.

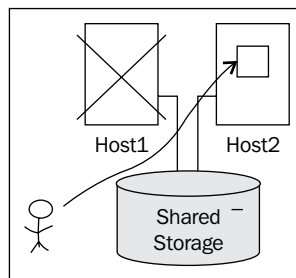


In the preceding diagram, the end user is connected to a vDesktop on Host1. Both Host1 and Host2 are part of the same cluster and can see the same shared storage. As illustrated in the diagram, the actual virtual disk files for the vDesktop reside on the shared storage and not on local storage within Host1.



As illustrated in the preceding diagram, when Host1 fails, the end user is disconnected from the vDesktop. In a persistent vDesktop solution, the end user is assigned to a specific vDesktop. In this case, the vDesktop is unavailable as it resides on Host1, which has just failed.

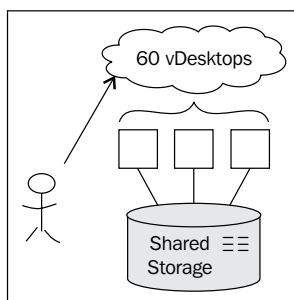
The end user will be unable to work until the vDesktop is back online or the end user is manually assigned to another (available) vDesktop resource.



As illustrated in the preceding diagram, VMware HA has powered the end user's vDesktop up on Host2, an available host in the cluster. By default, there is no notification to the end user that his/her vDesktop is now available, so the end user will need to try repeatedly for the time it takes vDesktop to come online on Host2. This typically takes between 1 and 3 minutes.



An advanced solution concept for persistent vDesktop environments is to monitor for individual vDesktop outages. If a user's persistent vDesktop is determined to be offline, an e-mail can be sent to the end user (who would likely receive it on his/her mobile device) letting him/her know that his/her vDesktop is currently unavailable but that the resolution is in progress. The same concept can then be used to detect when the vDesktop is back online and available (for example, by adding a 2 minute wait when a vDesktop enters the VMware Tool's OK state) and then notify the user that his/her vDesktop is now available.



For solutions using non-persistent vDesktops, the use of VMware HA is the topic of greater debate. While non-persistent solutions rely on a pool of vDesktops spread across multiple hosts in a cluster, end users are not assigned to an individual vDesktop. When a host fails in a non-persistent solution, any end users connected to vDesktops on that specific host lose their connectivity. The end users can then reconnect to the VMware View environment, and, as long as another vDesktop is available, that user will successfully connect to a resource. This is because vDesktop assignment is done randomly at the time of log in when using non-persistent vDesktops.

Non-persistent example

For this example, Company_A has 60 end users and has created a non-persistent vDesktop pool with the following settings:

- Numbers of end users = 60
- Desktop pool size (maximum number of desktops) = 60
- Number of spare (powered on) desktops = 0
- Power setting = Always On
- Provision all desktops up front

With these settings, when the pool is initially built, it will automatically provision 60 vDesktops and power them on.

The count is authoritatively held by VMware View in the ADAM database. If the pool's power settings are set to **Always On**, VMware View will create a pool of 60 vDesktops and immediately power all of them on. No matter what load exists from the end user community, 60 vDesktops will always be powered on. If 61 end users try to log in concurrently, 1 end user will be unable to access a resource.

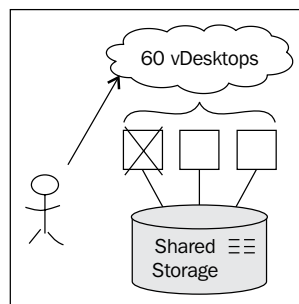
Imagine a scenario where Host1 hosts 30 vDesktops and Host2 hosts 30 vDesktops. The desktop pool is configured to host 60 vDesktops.

In this environment, if Host1 suddenly fails, the 30 vDesktops being hosted on Host1 enter an "Agent Unreachable" state. While the VMware View Connection Server has recognized that there are now 30 vDesktops that are unreachable, it does not provision 30 new vDesktops on the available hosts in the cluster (for example, Host2).

Therefore, without using HA to restart the vDesktops on another host, the pool's total number of vDesktops will be reduced. By using VMware HA, the pool's total number of vDesktops will not be reduced, although there could be a decrease in overall performance (if the ability to exceed available resources is allowed).

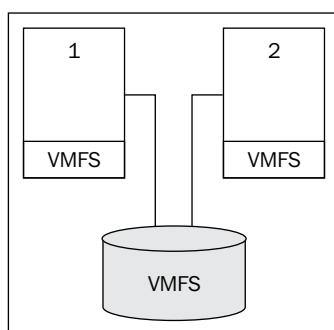
There are two design paths for non-persistent vDesktop solutions. The first is to simply use VMware HA to ensure that any vDesktops that reside on a failed host are restarted on another available host in the cluster. This is likely the easiest configuration and results in 5–10 minutes of downtime (as vDesktops power up and enter a useable state).

The second design path is to design the desktop pool(s) with enough vDesktops to sustain a host failure. It is important to ensure that the number of used vDesktops does not exceed the legally licensed amount from VMware. However, by building a desktop pool with additional capacity (for example, extra 30 vDesktops), and outage of one host has minimal impact on the end user environment. For those users that were connected to a vDesktop on the failed host, they simply log back into the VMware View environment and connect to one of the already provisioned, already available, extra vDesktops.



Using local storage

We have added a note about using local storage in this section. As will be covered later in this book, local storage is a viable option for certain VDI solutions. If the end user's vDesktop resides on local storage to Host1, during a host failure VMware HA would not be able to bring the vDesktop up on another host (for example, Host2) as other hosts would not have access to the virtual disk files residing on the local storage on Host1.



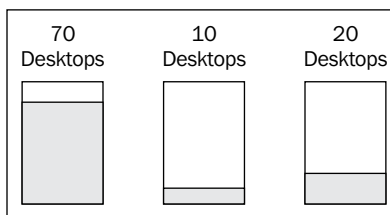
As illustrated in the preceding diagram, both Host1 and Host2 have a local **Virtual Machine File System (VMFS)** data store as well as access to a shared VMFS data store on the **Storage Area Network (SAN)**. If Host1 has an outage, any vDesktops or templates that were stored on the local VMFS data store on Host1 would be unavailable.

If a persistent solution was in use and vDesktops were placed on the local VMFS data store, end users would not have access to their vDesktops during a host outage. VMware HA would not matter as the vDesktops were not on shared storage, but instead on the local VMFS data store of the host.

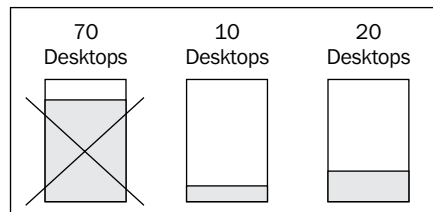
Therefore, it is imperative to use a non-persistent solution when placing core virtual disks of a vDesktop on local storage as end users are not specifically assigned to a unique vDesktop. If the physical server hosting vDesktop_17 – a persistent vDesktop assigned to employee User_LL – were to fail, User_LL would not be able to connect to a desktop resource.

VMware Distributed Resource Scheduling

While **VMware Distributed Resource Scheduling (DRS)** does not provide resilience, it does minimize the potential impact during an unpredicted physical host failure. By balancing the processing load across all of the available hosts in a cluster, a host failure will have approximately the minimum impact possible. This is true for both virtual machines running a server OS and vDesktops. In a VMware View solution with virtualized VMware vCenter Server(s) and/or VMware View Connection Server(s), it is prudent to use VMware DRS to balance the load in the cluster and minimize the impact. For additional considerations, please refer to the *Anti-affinity* section explained next.



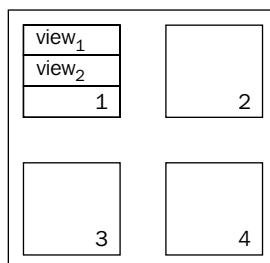
In the example given in the preceding diagram, there are three hosts in a cluster without VMware DRS enabled. On the first host in the cluster, there are 70 vDesktops. On the second one, there are 10 vDesktops, and on the third host, there are 20 vDesktops. As VMware DRS is not enabled, the load (and therefore the number of vDesktops) is not balanced across all of the available hosts in the cluster.



Continuing the example, if the first host were to have an unpredicted outage, 70 vDesktops would be impacted. If DRS was enabled and all vDesktops had roughly the same CPU consumption, approximately 34 vDesktops would be placed on each host. This will drastically reduce the number of end users that would experience an outage should a physical host fail.

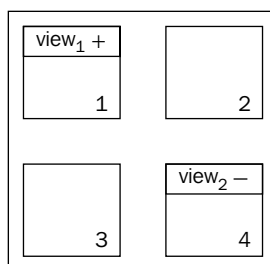
Anti-affinity

Affinity and **Anti-affinity** are settings within VMware DRS that determine how virtual machines in a given cluster react to one another.



In the preceding diagram, DRS is enabled for the four-host cluster and set to **Automatic**. There are no affinity or anti-affinity rules set. The VMware View solution requires two View Connection Servers, both of which have been virtualized and placed in the aforementioned cluster.

Through normal DRS activities, both View Connection Servers find themselves on Host1. If Host1 were to have an outage, no new connections would be permitted into the VDI.



In the previous illustration, the two View Connection Servers, View₁ and View₂ have been placed in an anti-affinity rule, and they have opposing polarity. This rule states that the two View Connection Servers are never to reside on the same host as long as there are available hosts in the cluster.

With anti-affinity in place, a single host outage would not have the potential to bring down the entire VMware View Connection Server environment.

VMware vCenter Server

VMware View uses VMware vCenter for all provisioning tasks. Without a functioning VMware vCenter Server, it is impossible to create, refresh, recompose, rebalance, or delete vDesktops. Therefore, utmost importance must be placed on protecting the VMware vCenter Server(s) used in the solution.


There are two primary components to the VMware vCenter Server service. They are as follows:

- VMware vCenter Server service
- Backend database

The VMware vCenter Server service should be protected in a way that eliminates downtime or provides a **recovery time objective (RTO)** of less than one minute. For extremely active VDI, a prolonged downtime can result in an inability to provide desktop resources to requesting end users. The most robust way to protect the VMware vCenter Server service is VMware vCenter Server Heartbeat (more on this is mentioned next).

The backend database used by VMware vCenter can reside on the same server (physical or virtual) or can reside on a separate database.

Component	Option 1	Option 2
vCenter Server	Virtual	Physical
Database location	On vCenter Server	On one or more separate servers
Database protection	Backup solution	Clustering solution

 The words in the preceding table that appear bold are the recommendations.

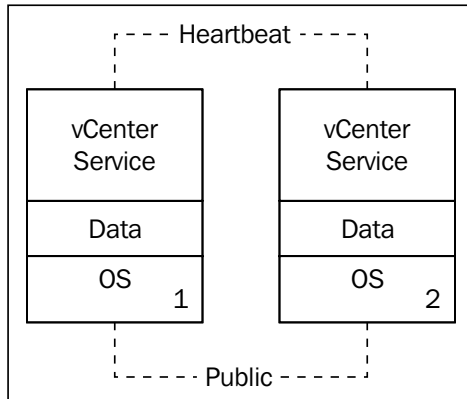
Cooperation from an organization's database team is important in VMware View solutions as a vCenter outage, which is typically database related, could wreak havoc on the VDI. Therefore, engaging with an organization's database team is a recommended part of the design process.

VMware vCenter Server Heartbeat

VMware vCenter Server Heartbeat (vCSH) is a product from VMware that is powered by Neverfail. vCSH monitors and protects all of the necessary components of a functioning VMware vCenter Server infrastructure, including:

- **Server:** vCSH protects from a physical or virtual server failure or operating system fault (for example, Blue Screen of Death)
- **Network:** vCSH protects the network identity including IP address and DNS name of the vCenter Server
- **Application:** vCSH protects the application environment specific to VMware vCenter Server service, and required files and registry entries
- **Performance:** vCSH monitors the performance of the underlying physical or virtual server
- **Data:** vCSH monitors all data and relevant applications, and maintains a copy of the data, including database (if local)

VMware vCenter Server Heartbeat requires two instances of VMware vCenter Server that are joined into a vCSH pair. The vCSH pair functions as a single VMware vCenter Server instance and shares the hostname, IP address, and other relevant information and configuration.



vCSH replicates data asynchronously from the primary vCenter Server in the pair to the secondary vCenter Server in the pair. VMware vCenter is also aware of the VMware View Composer service to ensure that it's protected as well.

Why VMware vCenter Server Heartbeat should be used

For production environments, sizable deployments, or solutions with high criticality, VMware vCenter Server Heartbeat should always be used as it protects the most vulnerable component of the VDI.

The storage array typically has high availability built-in with RAID, multiple storage processors, power supplies, fans, extra disks, and so on.

The physical hosts are protected by multiple NICs, multiple power supplies, and VMware functionality, such as VMware HA.

The networking layer is protected by redundancy in hardware and network paths.

The View Connection Server is protected by duplicate instances, load balancing, and potentially by VMware Fault Tolerance.

However, as a single instance of VMware vCenter Server can be responsible for over a dozen physical hosts and potentially many hundreds of vDesktops, the protection of its state is of paramount importance.

VMware vCenter Server Heartbeat has the ability to protect not only the VMware vCenter Server service, but also the vCenter database (if local) and the VMware View Composer service.

VMware View

VMware View is responsible for processing incoming requests for vDesktops, interacting with VMware vCenter Server to provision, recompose, and delete vDesktops; as well as a variety of other tasks necessary for a properly functioning VDI.

Replica

When installing the VMware View Connection Server, there are four installation types. They are as follows:

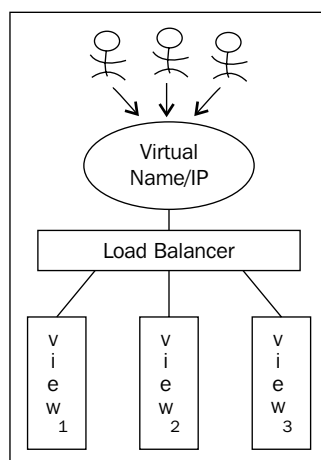
- Standard
- Replica
- Security
- Transfer

For the first VMware View Connection Server instance in a View Connection Server pool, the standard installation should be selected. However, to eliminate the View Connection Server as a single point of failure, a second (and additional) View Connection Server can be installed. Once a View Connection Server exists in the infrastructure, additional View Connection Servers can be joined to the original, forming a View Connection Server pool. To join a new View Connection Server to an existing View Connection Server or View Connection Server pool, select the **replica** installation mode.

When a replica View Connection Server instance is created, it copies the VMware View LDAP configuration from the existing View Connection Server instance.

Load balancing

The VMware View Connection Servers are responsible for brokering the connection between an authorized end user and a vDesktop in the VDI. Therefore, if there are no available VMware View Connection Servers, no new connections can be made to the VDI. However, the existing connections will not be affected if there are no available VMware View Connection Servers.



Therefore, it is important to protect the available View Connection Server(s) in the VDI. It is best practice to have a minimum of two VMware View Connection Server(s) (or potentially one protected by VMware Fault Tolerance). The easiest way to accomplish resilience for the VMware View Connection Servers is to use a load balancing solution. There are various load balancing solutions available, including **Microsoft Network Load Balancer (NLB)** and hardware appliances from companies like F5 (preferred).

A load balancing solution will create a virtual IP address that will be used by end users to connect to the VDI. Behind the virtual IP address will be the actual IP addresses of all of the VMware View Connection Servers in the load balancing pool. If a VMware View Connection Server is not responsive to a ping (for example), it will be removed from the load balancing pool to ensure that incoming end user requests are not routed to an unavailable View Connection Server.

Many load balancing solutions also offer the ability to monitor availability via HTTP GET and similar commands to ensure that not only is the server online but that it is also responsive to web-based requests.

VMware Fault Tolerance

VMware Fault Tolerance (FT) can be used as an additional layer of protection for the VMware View Connection Server(s). VMware FT protects a VM by creating and maintaining a secondary VM that is identical to the primary one. The secondary VM is continuously available to replace the primary VM in case of failure of the host where the primary VM resides.

In VMware FT, there is no downtime (unlike VMware HA). VMware FT does have several limitations including supported hardware, number of vCPUs (currently 1), and so on.

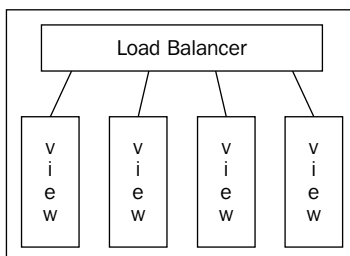
VMware FT takes inputs and events that occur on the primary VM and transfer them to the secondary VM, which is running on a separate host in the cluster.

VMware FT does impact the virtual infrastructure design, as it requires a separate and dedicated NIC for FT Logging. The FT Logging and vMotion NICs must reside on separate subnets. Also, no more than four VMware FT-enabled virtual machine primaries or secondaries can reside on any single ESX host.

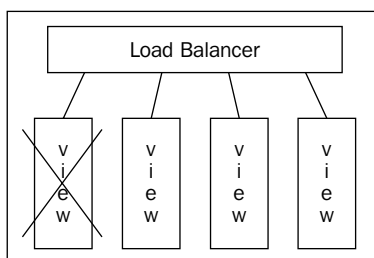
Design impact when using VMware FT

To properly illustrate the design impact of using VMware FT, the following example will be used.

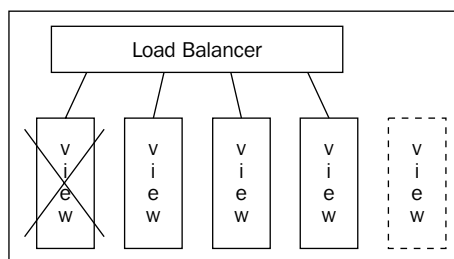
Through a thorough analysis, it has been determined that CustomerA requires four VMware View Connection Servers to satisfy the demand of incoming requests.



As part of the solution, four VMware View Connection Servers are installed and configured, and placed behind a load balancer.

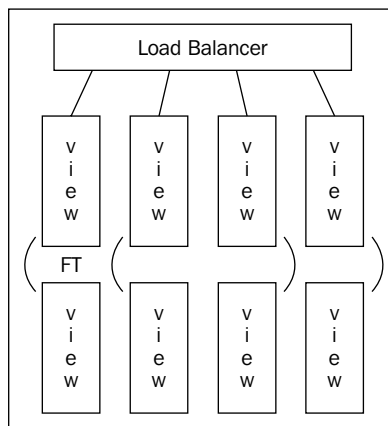


If a VMware View Connection Server has a critical fault (for example, Blue Screen of Death), only three connection servers are available to the end users. There is a possibility that the three remaining connection servers cannot handle the load and some end users are unable to connect to the VDI.

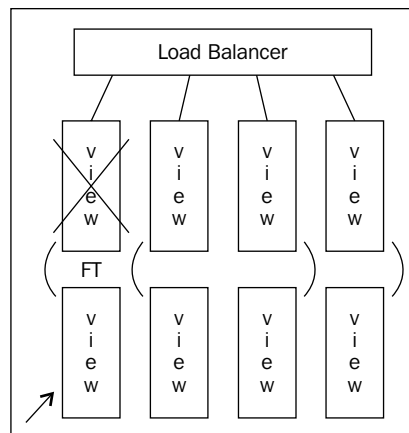


The only way to restore connectivity for a failed View Connection Server that's had a critical fault is to either restore from a previous backup or build a new View Connection Server, and add it to the load balanced pool, as illustrated in the preceding diagram.

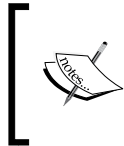
If a VMware View Connection Server resides on a host that has a failure, VMware HA will power the virtual machine up on an available host in the cluster. There may be several minutes of downtime for some end users (if the remaining three View Connection Servers are unable to handle the load). However, after 3 to 5 minutes, full connectivity should be restored.



By leveraging VMware FT to protect the View Connection Servers, there will be zero impact should a physical host failure occur in this solution (assuming anti-affinity has separated primary and secondary VMs).



In the event of a physical host failure that impacts a View Connection Server, VMware FT immediately makes the secondary (shown in bottom row in the preceding diagram) active. Through technology from VMware vLockstep, the secondary is an identical copy of the primary that resided on the failed host. Once the failover has successfully occurred, the secondary is now marked as the primary. In addition, a new secondary is spawned from the newly appointed primary.



However, VMware FT does not protect against guest operating system failures (for example, Blue Screen of Death). Therefore, coupling VMware FT with VM Monitoring via VMware HA is the most robust solution possible.

While VMware FT is a useful technology in protecting VMware View Connection Servers, most virtualization architects would prefer to simply add an additional View Connection Server to the original design instead of adding the complexity of VMware FT.

Parent vDesktop and templates

Virtual machine templates are used by VMware View when deploying vDesktops with the **full virtual machine** option selected. Standard virtual machines (not templates) are used by VMware View when deploying vDesktops with the **View Composer linked clones** option selected. For a virtual machine to be seen by View Composer, it must have at least one snapshot. View Composer deploys all vDesktops in the pool from the selected snapshot.

It is important to understand that if the parent vDesktop (for Linked Clone pools) or the gold vDesktop template (for Full Desktop pools) are not available, then new vDesktops cannot be provisioned.

Templates

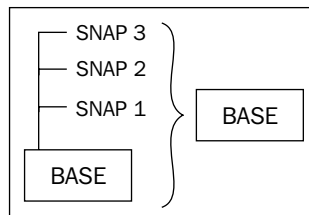
Virtual machine templates are a bit confusing to protect. When creating a virtual machine template or adding a virtual machine template to the inventory, the administrator must select a specific host within a cluster. According to the **graphical user interface (GUI)**, "Choose a specific host within the cluster. On high-availability clusters and fully-manual dynamic workload management clusters, each template must be assigned to a specific host."

Therefore, if the gold template for vDesktops resides on Host1 and if Host1 experiences a failure, VMware HA will not recover this template. Instead, the original template will be shown as unavailable within vCenter. From this point, the original inventory entry in vCenter for the template can be removed and then the template can be re-added. This is possible because while the host is unavailable, the virtual machine template actually resides on shared storage (assuming that best practice was followed).

When assisting an organization with operational readiness, the preceding recovery process should be listed in a Standard Operating Procedure manual.

Parent vDesktops with snapshots

To protect the parent vDesktop and its snapshot, a simple clone virtual machine task will not suffice. This is because the clone task consolidates the snapshot tree and thus removes all snapshots associated with the base virtual machine.



Therefore, it is imperative that VMware HA should be used to protect the parent vDesktop. As the parent vDesktop is simply a virtual machine with snapshots (as opposed to a virtual machine template in the preceding scenario), VMware HA will change the ownership of the parent vDesktop to an available host in the cluster during a host outage.

User personas

For environments that leverage a user persona solution, such as Liquidware Labs ProfileUnity™, placing the user personas on a highly available network share is critical to ensure end user data is always available. By using the **Distributed File System (DFS)** service or **Distributed File System Replication (DFS-R)** service, a file share storing user personas will still be available in the event of a file server failure. In addition, with DFS-R, user personas can be replicated to other servers in the same site or other sites. DFS-R enables a VDI to provide **Continuity of Operations (COOP)** by ensuring that the file share containing the user personas has its data replicated offsite.

Microsoft DFS also leverages Active Directory sites to ensure that an end user is retrieving their persona from the nearest server participating in the DFS/DFS-R group. In addition, site costing can be used to state which is the least expensive target selection for end users attempting to retrieve their user persona from a network share.

The following table shows a summary of the types of failures for all components:

Component	Type of failure	Protected by	Downtime	Notes
vCenter Server	Underlying physical host	VMware HA	Approximately 10 minutes	During the outage, vDesktop tasks such as provision, Recompose, and so on are unavailable. vCenter may take longer to start (as opposed to View Connection Server) because of the database actions that are performed during an initial service start.
vCenter Server	Underlying physical host	vCenter Server Heartbeat	Less than 1 minute	It requires a second vCenter Server instance.
vCenter Server	Operating system (Blue Screen of Death)	vCenter Server Heartbeat	Less than 1 minute	It requires a second vCenter Server instance.
vCenter Database	Any	Clustering solution	Less than 1 minute	It requires two database servers.
vCenter Database	Database corruption	Backup/restore/snapshots	Varies	The time to restore depends on the solution used, speed of media, and throughput available.

Redundancy

Component	Type of failure	Protected by	Downtime	Notes
View Connection Server	Underlying physical host	VMware HA	Approximately 5 minutes	It requires multiple View Connection Servers behind a load balancer to mitigate impact to the end users.
View Connection Server	Underlying physical host	Load balancer	0 minutes	It requires multiple View Connection Servers behind a load balancer to mitigate impact to the end users. Without VMware HA, the total number of inbound connections may be impacted.
View Connection Server	Underlying physical host	VMware FT	0 minutes	It does not protect against guest operating system failures (see VM Monitoring with VMware HA).
View Connection Server	View Connection Server service	Load balancer	0 minutes	It requires multiple View Connection Servers behind a load balancer to mitigate impact to the end users.
View Connection Server	Operating system (Blue Screen of Death)	VM Monitoring with VMware HA	Approximately 5 minutes	It requires multiple View Connection Servers behind a load balancer to mitigate impact to the end users.

Component	Type of failure	Protected by	Downtime	Notes
View Composer	Underlying physical host	VMware HA	Approximately 10 minutes	During the outage, vDesktop tasks such as provision, Recompose, and so on are unavailable. vCenter may take longer to start (as opposed to View Connection Server) because of the database actions that are performed during an initial service start.
View Composer	Underlying physical host	vCenter Server Heartbeat	Less than 1 minute	It requires a second vCenter Server instance.
View Composer	Operating system (Blue Screen of Death)	vCenter Server Heartbeat	Less than 1 minute	It requires a second vCenter Server instance.

Summary

Thus far, some of the most important design considerations (for example, persistent or non-persistent) have been addressed, as well as proper sizing of the overall VDI. Designing a VMware View solution that is highly resilient is also paramount to a production-quality solution. In the next chapter, the last major hurdle, storage design, will be discussed. An improperly sized VDI can result in a poor end user experience. A VDI without redundancy can result in unexpected outages and downtime. A VDI with improperly designed storage can not only result in poor end user experience but also significantly add to the overall cost of the VDI solution. As storage is often one of the expensive components, (if not the most expensive component) of a VDI solution, judicious sizing that will still meet the requirements is key.

8

Sizing the Storage

The storage layer is perhaps one of the most critical components in a VMware View design. For many VDI professionals, this is likely to be the major issue when called in for a performance troubleshooting exercise. Commonly, the storage layer is the root cause of the performance issue. Why is storage so critical? To answer this question, we first need to understand how Intel®-based desktops work and interact with Windows operating systems before diving into the world of storage for VDI.

Physical desktops have always had dedicated hard disks to rely upon and only a single Windows kernel had access to the disk causing a single I/O stream. Despite being dedicated to the desktop, that device also faced disk contention. This contention could have been generated due to an excessive amount of disk I/O operations or I/O block sizes. The important thing is that no matter what type of operation causes the contention, the end result is high response time, also known as **latency**.

Recent disk technology advancements such as **Solid State Drives (SSD)** drastically reduced the latency implications, and thus improved the end user experience. The most advanced SSD can ingest and deliver an enormous amount of I/O and throughput.

With the ability to use faster disks, users became spoilt with the given enhanced performance. However, microprocessors and RAM technology also evolved in such a fast fashion that technologies like Intel Core i7 and DDR3 quickly made even the fastest SSD become the performance bottleneck once again.

With a few exceptions, VDI implementations do not utilize a single dedicated disk. Instead, VDI uses a pool of disks to provide storage capacity, I/O, and throughput to vDesktops.

Most VDI implementations require shared storage to provide shared datastores to multiple servers. Shared storage is also a key enabler for VMware vSphere features such as vMotion, DRS, and Fault Tolerance. Implementations that utilize a combination of floating pools and roaming profiles may opt for storage appliance solutions, where no persistent data is stored on those appliances.

Local disks, **Dedicated Attached Storage (DAS)**, and even diskless solutions can be employed in VDI deployments. It is important to understand the use cases and implications behind each of the approaches.

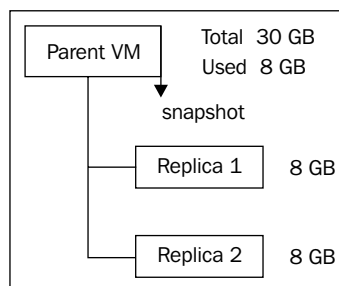
Storage architecture decisions made during the VDI design phase will have a deep impact on how the infrastructure will perform and operate. The type of storage and transport protocol chosen will determine how VMware View and vSphere will operate. Yet, the type of storage and protocol will also dictate how datastores should be designed or how many desktops per datastore should be used.

VMware View Composer

The VMware View infrastructure includes VMware View Composer as an optional component. View Composer runs as a Windows service on the vCenter Server(s) and enables View Manager to rapidly clone and deploy multiple virtual desktops from a single centralized standard base image. View Composer was originally designed to reduce the total storage required in VDI deployments; however, today View Composer also provides essential management features such as the Refresh and Recompose operations.

View Composer uses linked clone technology. Unlike a traditional virtual machine model wherein each VM exists as an independent entity with dedicated virtual disks, View Composer creates dependent VMs all linked to a master VM. This master VM is called the **Parent VM** in VMware terminology.

The Parent VM is used as a base image, and a snapshot and copy are taken from the Parent VM to create the replica image, which will serve as the master VM disk for all linked clones in a desktop pool.

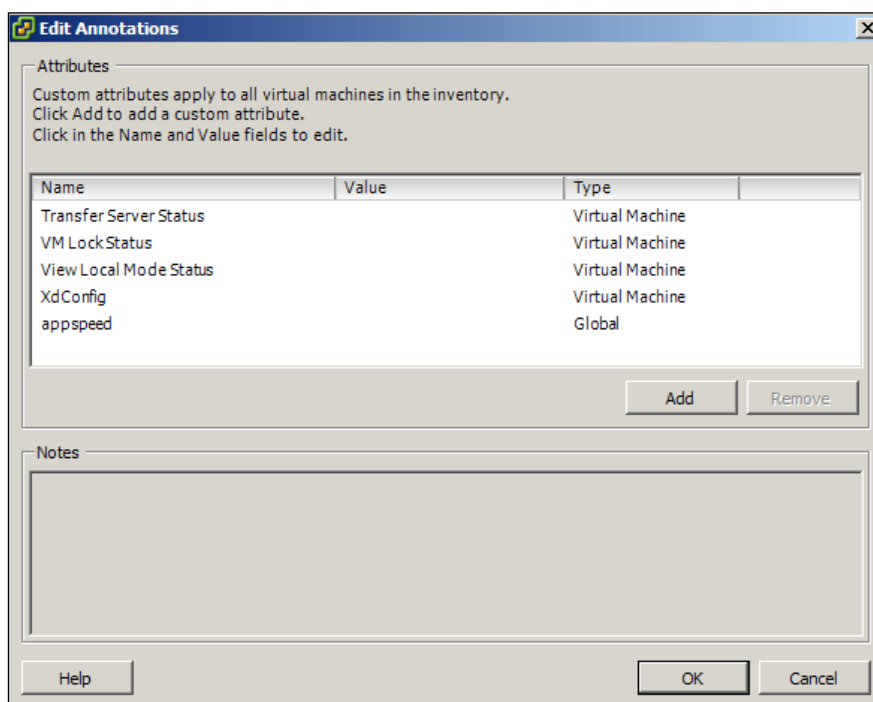


The replica disk is created as a read-only thin provisioned entity from the Parent VM to ensure that any subsequent changes to the Parent VM do not impact the linked clone desktops. As mentioned previously, the replica is thin provisioned which means that only the data contained within the Parent VM is copied to the replica. As an example, if the Parent VM was created with a 40 GB disk but only 20 GB appears on the guest's Windows NTFS volume, then the replica will be 20 GB.

The replica image is a protected entity in vCenter Server via a **VM LockStatus** parameter added to the VM annotations, as seen in the following screenshot. If a replica needs to be deleted for any reason, the process outlined in KB1008704 must be followed. It is given at the following URL:

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1008704.

The following screenshot shows the **VM LockStatus** parameter:

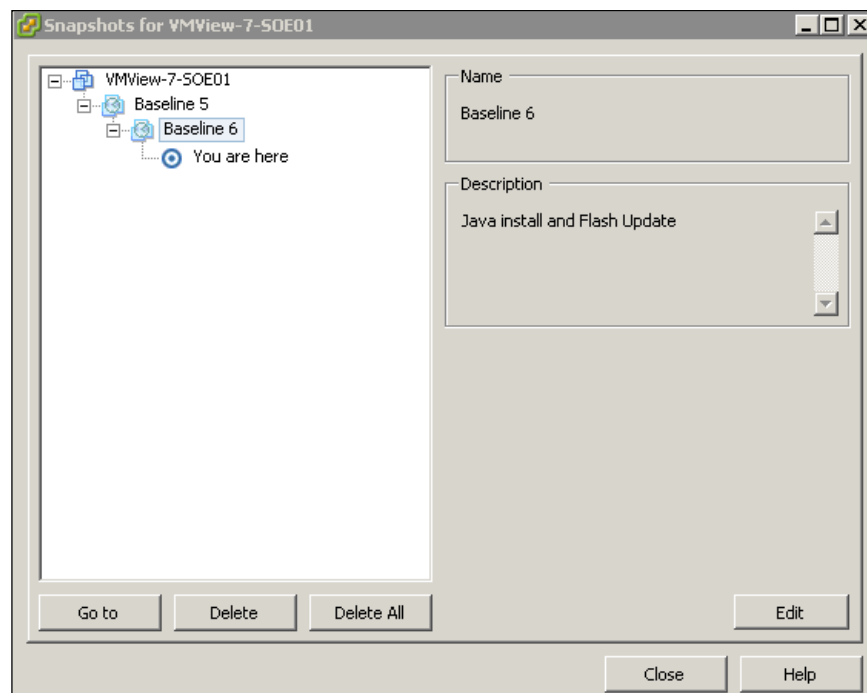


A Parent VM may contain several snapshots that represent changes introduced to the base image. These differences may be due to fixes, patches, and upgrades required by the Windows Guest OS. The deployment of new applications to the base image, application upgrades, or even Windows configuration changes can be added to the snapshots.

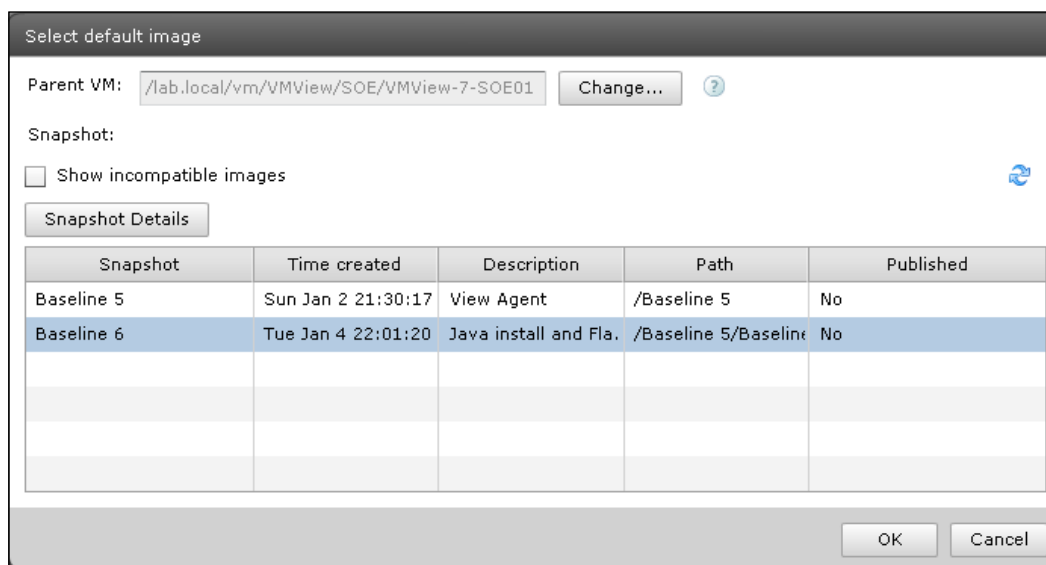
After each modification, the Parent VM must be shutdown by the administrator and a new snapshot is taken.

The snapshot to be used in a desktop pool is selected during the desktop pool configuration. A single snapshot is assigned to the entire desktop pool. However, it is possible to individually recompose virtual desktops using different snapshots from the same or different Parent VM.

The following screenshot demonstrates VMware vCenter Server Snapshot Manager with a few snapshots already taken. It is recommended practice to annotate the changes made to the Parent VM in the description field:



The VMware View snapshot selection during the desktop pool configuration process can be seen in the following screenshot, which shows snapshot selection in the View Admin console:



In releases prior to VMware View 4.5, a unique replica disk was created in each datastore hosting virtual desktops for a desktop pool. Additionally, for each different snapshot in use by a desktop pool, a new replica disk used to be created for each datastore in use.

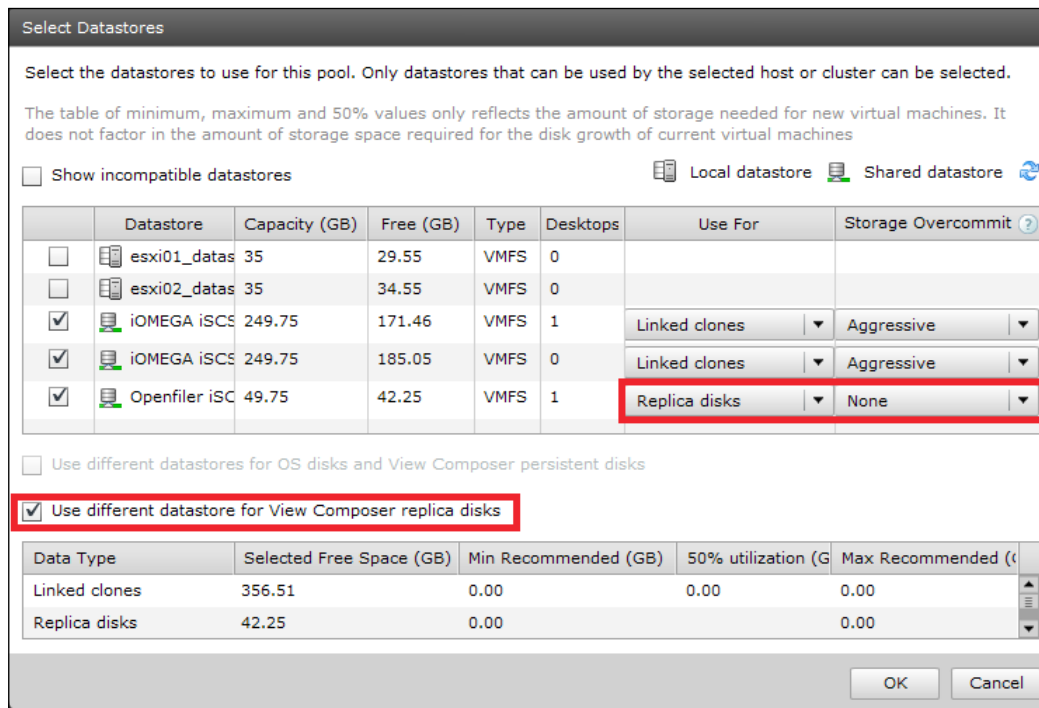
Only a single snapshot can be assigned to the desktop pool at any one time. However, after selecting a different snapshot for the desktop pool and triggering Recompose action, a second replica disk representing the new snapshot is created in each datastore in use by the virtual desktops using the replica disk. In this case, each datastore may contain two replicas for the desktop pool.

In a Recompose operation, the original replica image is only deleted after all desktops in the datastore are recomposed with the new base replica disk and the old replica is not required anymore. Therefore, it is important to ensure that there is ample space available on the datastore(s) dedicated to storing replicas.

This scenario is still applicable with VMware View 5.0 when the administrator does not select the optional Dedicated Replica Datastore feature during the datastore selection.

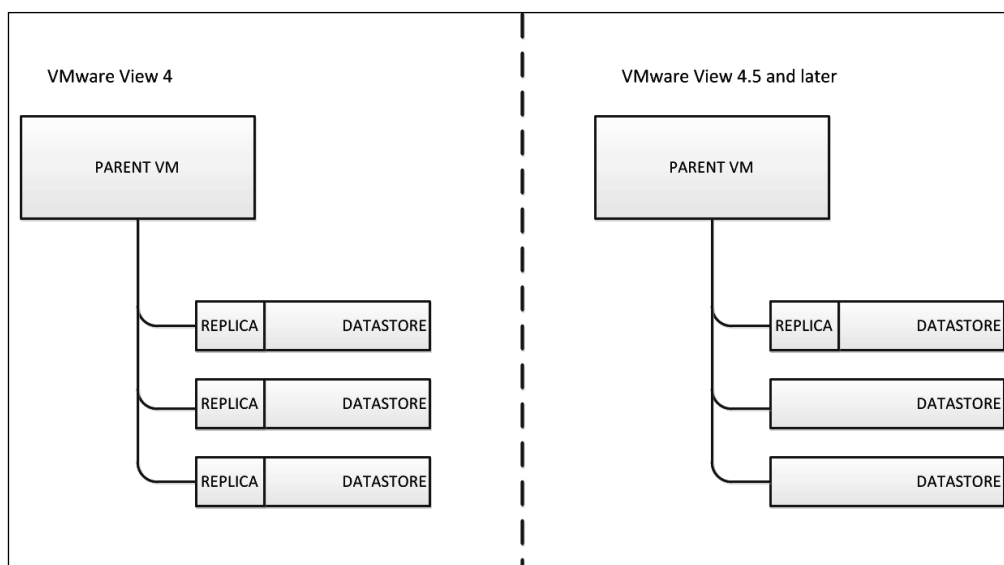
VMware View 4.5 and later implemented the ability to specify a unique datastore to host replica disks for an entire desktop pool. This piece is part of the VMware View Tiered Storage feature.

The following screenshot shows datastore selection in the View Admin console:



If the desktop pool is large and eventually uses the entire vSphere cluster resources, it is possible to end up with a single replica disk for the entire cluster. For VMware View 5.0, the maximum number of virtual desktops supported in a single desktop pool is 1,000. While it is possible to go further than 1,000 desktops, it is not recommended nor supported.

In some cases, where multiple snapshots are in use, multiple replicas will be created in the single datastore selected during the pool configuration. The following diagram demonstrates the differences between not using and using the Dedicated Replica Datastore option in VMware View 4.5 and above:



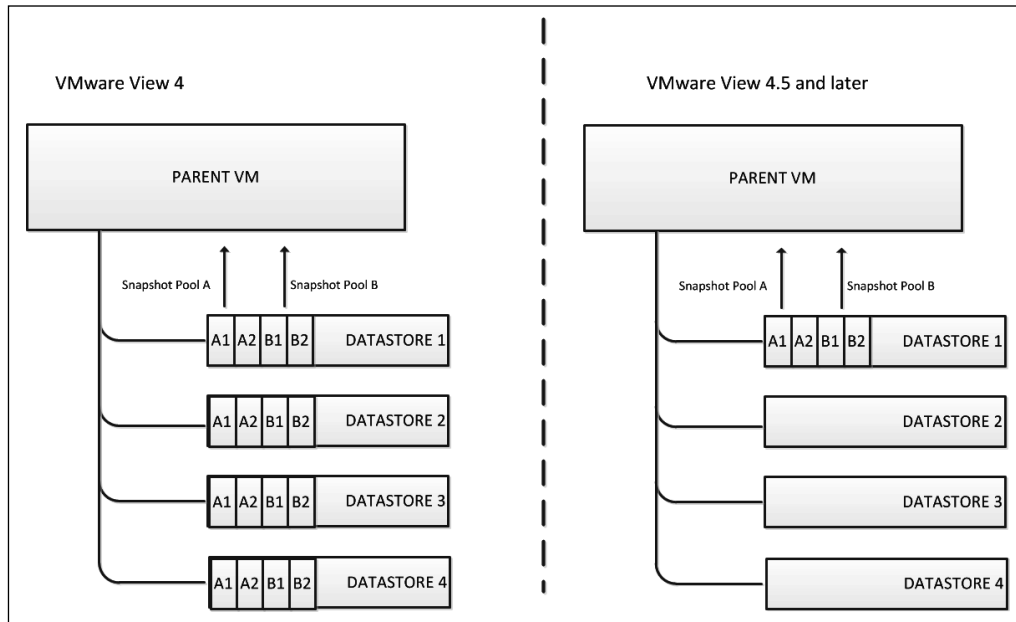
In the following sample scenario, VMware View is running 256 virtual desktops across 2 desktop pools with 2 snapshots in use for each pool. If the replica disk size is 20 GB, the total storage allocation for the replica disks would be 320 GB, being 80 GB per datastore.

Number of desktop pools * replicas in use * number of datastores * replica size = $2 * 2 * 4 * 20 \text{ GB} = 160 \text{ GB}$.

Using the same sample scenario with Dedicated Replica Datastore, the total storage allocation for the replica disks would be 80 GB in a single datastore.

Number of desktop pools * replicas in use * replica size = $2 * 2 * 20 \text{ GB} = 80 \text{ GB}$

The following diagram demonstrates both scenarios. It is an illustration showing the difference between the replica disk placement when using multiple snapshots and a Dedicated Replica Disk Datastore in View 4.5 and above:



Running the same calculations for the preceding scenario with 2,000 virtual desktops, the storage savings provided by the View Composer technology could be as large as 6.3 TB. The more virtual machines and desktop pools in the environment, the bigger the number of replicas per datastore when the Dedicated Replica Datastore option is not selected.

It is possible to break down the amount of replica disks and storage consumption if all datastores are not selected for each desktop pool, therefore, limiting the placement of virtual desktops across datastores.

The number of datastores and datastore sizes must provide the capacity and performance requirements for the provisioning of the required number of desktops. This concept assumes that the administrator will carefully manage and select the datastores in use by the desktop pool, not allowing all datastores to be used by all desktop pools.

The following screenshot shows a solution not using a Dedicated Replica Datastore option:

Select Datastores

Select the datastores to use for this pool. Only datastores that can be used by the selected host or cluster can be selected.

The table of minimum, maximum and 50% values only reflects the amount of storage needed for new virtual machines. It does not factor in the amount of storage space required for the disk growth of current virtual machines

☐ Show incompatible datastores Local datastore Shared datastore

	Datastore	Capacity (GB)	Free (GB)	Type	Desktops	Storage Overcommit ?
<input type="checkbox"/>	esxi01_datastore1	35	29.55	VMFS	0	
<input type="checkbox"/>	esxi02_datastore1	35	34.55	VMFS	0	
<input checked="" type="checkbox"/>	iOMEGA iSCSI Disk0	249.75	171.46	VMFS	1	Moderate
<input type="checkbox"/>	iOMEGA iSCSI Disk1	249.75	185.05	VMFS	0	
<input checked="" type="checkbox"/>	Openfiler iSCSIi Disk0	49.75	42.25	VMFS	1	Moderate

☐ Use different datastores for OS disks and View Composer persistent disks

☐ Use different datastore for View Composer replica disks

Data Type	Selected Free Space (GB)	Min Recommended (GB)	50% utilization (G)	Max Recommended (G)
Linked clones	213.71	0.00	0.00	0.00

OK Cancel

A linked clone disk is also called a **delta disk** because it accumulates delta changes. After the replica disk is created, View Composer starts to create the linked clone virtual desktops. Each linked clone has a unique delta disk and is linked to the replica disk.

Delta disks contain only the differences from the original read-only replica disk that are unique to the cloned virtual desktop, resulting in significant storage savings. Linked Clone disks will grow over time according to block write changes requested by the Guest OS, and may grow up to the maximum size of the Parent VM.

As an example, if the Parent VM was originally configured by the administrator with a 30 GB flat disk, this will be the maximum size of the delta disk.

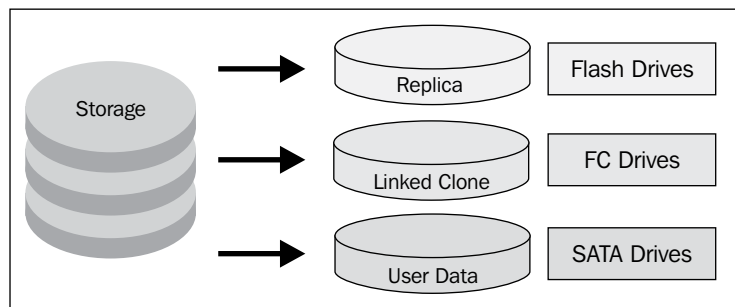
View Composer allows for great storage savings; however, there will be dozens or hundreds of linked clone virtual desktops using the same datastore to read that single existing replica disk. If the Dedicated Replica Datastore option is not in use, the replica is only used by the desktops hosted in the same datastore.

All virtual desktops accessing the replica disk will cause I/O stress on LUNs, RAID group, and disks, and may create I/O contention. The I/O contention is caused by multiple virtual desktops and users accessing the same datastore, all at the same time.

Each datastore is normally backed by a LUN, if **Fibre Channel Protocol (FCP)** is in use or by an Export if NFS is in use. Both LUN and Export are backed by a RAID Group configuration that encompasses a pool of disks configured to support the workload. Those disks and LUNS together must be able to meet the required performance specifications for the replica disk. These specifications are **Input/Output Operations Per Second (IOPS)** and throughput.

If a decision to use a dedicated replica datastore is made during the design phase, it is recommended to allocate Tier 1 storage, for example, SDD to host the replica disk.

The following diagram is an illustration showing a typical virtual disk to drive type associations:



Storage vendors have different solutions and architectures to solve response time and latency issues. Some of the solutions available are automated storage tiering, storage pools, diskless environments, inline I/O de-duplication, and local host caching.

VMware vSphere files

Mentioned in the following table are the files and disks created for the virtual desktop provisioned through VMware View and are the standard files for any virtual machine created on a vSphere hypervisor:

File type	Description
.vmx	The .vmx file is the primary configuration file for a virtual machine. It contains information such as the operating system, disk sizes, networking, and so on.
.vmsd	Information and metadata about snapshots.
.vmxf	Supplemental configurations file for virtual machines that are in a team. Note that the .vmxf file remains if a virtual machine is removed from the team.
.vswp	The .vswp file is a swap file created for each virtual machine to allow for VM memory over-commitment on an ESXi host. This file is created when a VM is powered on and will be equal in size to the unreserved memory configured for the VM. When VMs are created, the default memory reservation is 0 MB, so the size of the .vswp file is equal to the amount of memory allocated to the VM. If a VM is configured with a 1024 MB memory reservation, the size of the .vswp file will be equal to the amount of memory allocated to the VM minus the 1024 MB reservation.
.vmss	The .vmss file is created when a VM is suspended and is used to save the suspended state. In essence, this is a copy of the VM's memory and will always have the size of the total amount of allocated RAM. One thing to note is that a .vmss file is created when the machine enters the suspended state, however it's not removed when the VM is removed from the suspended state. The .vmss file is only removed when the VM is powered off. If a VM is configured with 2048 MB memory, the size of the .vmss file will be 2048 MB.
.nvram	This is the file that stores the state of the virtual machine's BIOS.
.vmsn	Snapshot state file, which stores the running state of a virtual machine at the time you take the snapshot.
-flat.vmdk	The -flat.vmdk is a raw disk file that is created for each virtual disk allocated to a given VM and will be of the same size as the virtual disks added to the VM at the time of creation. It's a pre-allocated disk file only available with full clone VMs.
.log	VM log files are relatively small.

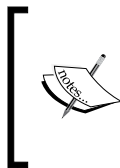
VMware View specific files

Mentioned in the following table are the files and disks created for virtual desktops provisioned through VMware View. Some of the disks are only created by View Composer or when assigning a persistent or disposable disk to the desktop pool:

File type	Composer	Description
replica-GUID.vmdk	Yes	Replica VM is used to spin-up linked clone VMs.
-internal.vmdk	Yes	Data configuration for Quick Prep/Sysprep.
VM-s000 [n] .vmdk	Yes	Created when a virtual machine has snapshot(s). This file stores changes made to a virtual disk while the virtual machine is running. There may be more than one such file. The 3-digit number after the letter (000 after s in this case) indicates a unique suffix added automatically to avoid duplicate filenames.
VDM-disposable-GUID.vmdk	Yes	Redirected Windows OS pagefile and temporary files.
.log	No	VM log files.

Tiered storage

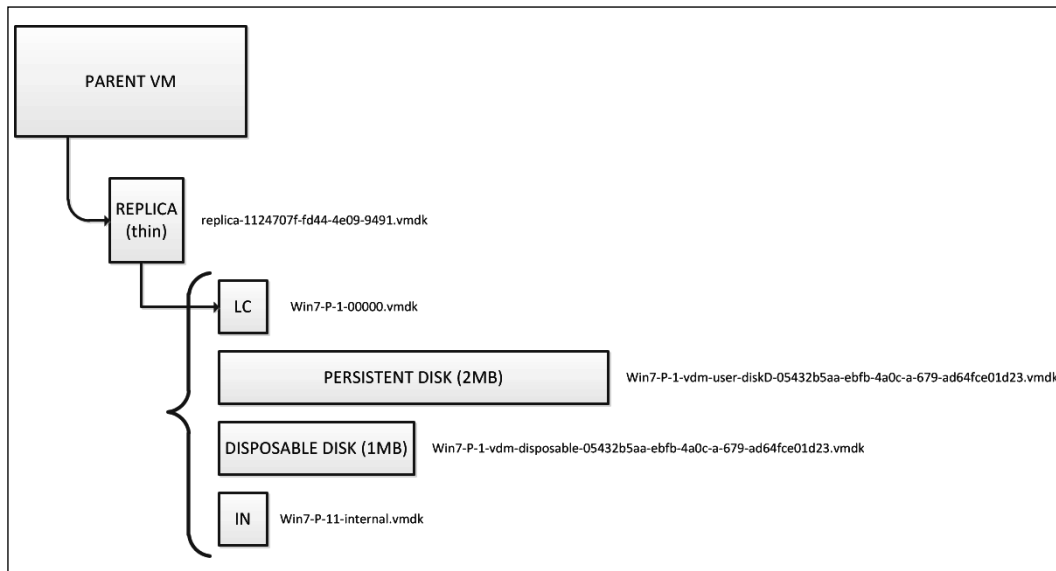
VMware View 4.5 and above allow administrators to select different datastores to host different types of virtual disks (replica, linked clone, and persistent). Data performance classification is an important part of storage tiering implementation and allows administrators to select datastores that provide the most appropriate storage tier in regards to performance, cost, and capacity for each type of disk in use.



Important: Do not confuse the storage tiering feature provided by VMware View with auto storage tiering offered by storage vendors. The solution provided by VMware View is static and does not automatically move data around to achieve best performance. The solutions are complementary to each other.

With the introduction of storage tiering and the ability to segment workloads across datastores and types of disks, it is important to understand what type of disks and data are created for each virtual desktop. The type of disks created may differ for each implementation. Desktop pools that utilize linked clone technology may have additional virtual disks that are not created when using traditional full clone provisioning.

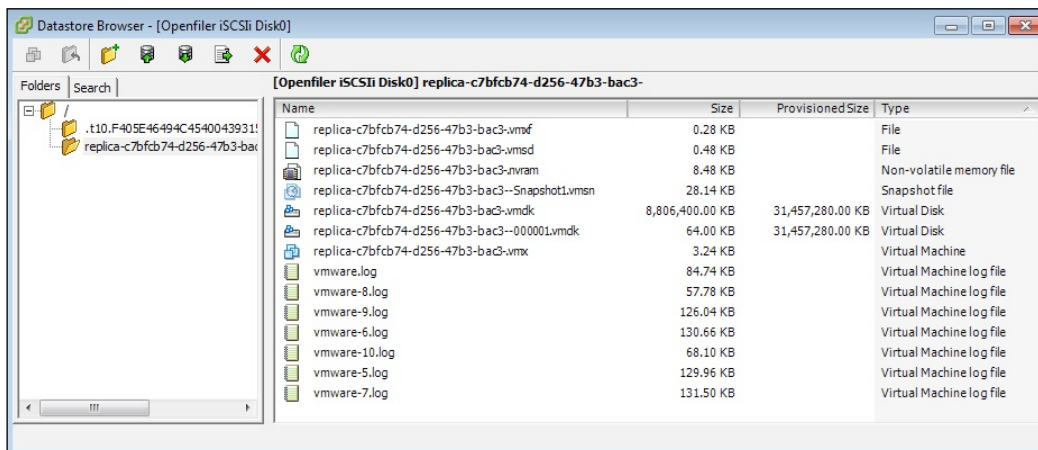
The following diagram is an illustration showing the multiple types of virtual disks in use by VMware View:



Replica disk

The `replica-GUID.vmdk` folder contains all files required to run the virtual machine, however, it will be exclusively used as read-only and as the base for linked clone virtual desktops. The following screenshot demonstrates the folder and files created to host a replica disk.

Despite the fact that the provisioned size of the disk is set to 30 GB (31,457,280 KB) in the following example, only 8 GB is in fact used. The reason for this is that View Composer makes use of vSphere VMFS thin provisioning technology to create replica disks. This is an automatic setting that cannot be changed via View Manager UI and will work independently of the Parent VM being thin or thick provisioned. For NFS deployments, thin is also the only provisioning mechanism.



Name	Size	Provisioned Size	Type
replica-c7bfc74-d256-47b3-bac3-vmx	0.28 KB		File
replica-c7bfc74-d256-47b3-bac3-vmx	0.48 KB		File
replica-c7bfc74-d256-47b3-bac3-nvram	8.48 KB		Non-volatile memory file
replica-c7bfc74-d256-47b3-bac3--Snapshot1.vmsn	28.14 KB		Snapshot file
replica-c7bfc74-d256-47b3-bac3-vmx	8,806,400.00 KB	31,457,280.00 KB	Virtual Disk
replica-c7bfc74-d256-47b3-bac3--000001.vmdk	64.00 KB	31,457,280.00 KB	Virtual Disk
replica-c7bfc74-d256-47b3-bac3-vmx	3.24 KB		Virtual Machine
vmware.log	84.74 KB		Virtual Machine log file
vmware-8.log	57.78 KB		Virtual Machine log file
vmware-9.log	126.04 KB		Virtual Machine log file
vmware-6.log	130.66 KB		Virtual Machine log file
vmware-10.log	68.10 KB		Virtual Machine log file
vmware-5.log	129.96 KB		Virtual Machine log file
vmware-7.log	131.50 KB		Virtual Machine log file

Internal disk

The `internal.vmdk` disk is a small disk and contains the configuration data for Quick Prep/Sysprep. In previous VMware View releases, operations, for example, Refresh would incur a full desktop deletion, followed by provision and customization of a new virtual desktop. This process used to take a long time to complete and would normally draw a high number of Compute and Storage resources.

VMware View 4.5 and later started implementing a different technique to refresh virtual desktops that make use of the vSphere snapshot technology.

A Refresh operation is simply a snapshot revert-back operation. The internal disk is created to store the Active Directory computer account password changes that Windows performs every so often as per default AD policy setting. The computer account password is encrypted before being stored on the internal disk.

Whenever the domain computer account password is changed, VMware View Agent stores another encrypted copy of the password in the disk. This ensures that domain connectivity is maintained when a desktop is refreshed.

The internal disk is connected to the desktop; however, it does not get a drive letter assigned to it.

The following screenshot shows the internal disk screenshot from within a guest vDesktop showing the internal disk:

Name ^	Date modified	Type	Size
\$RECYCLE.BIN	27/03/2011 12:59 PM	File folder	
System Volume Information	27/03/2011 9:02 AM	File folder	
domstate.dat	31/03/2011 10:53 AM	DAT File	1 KB
sim.dat	27/03/2011 8:50 AM	DAT File	1 KB
simvol.dat	27/03/2011 8:50 AM	DAT File	1 KB

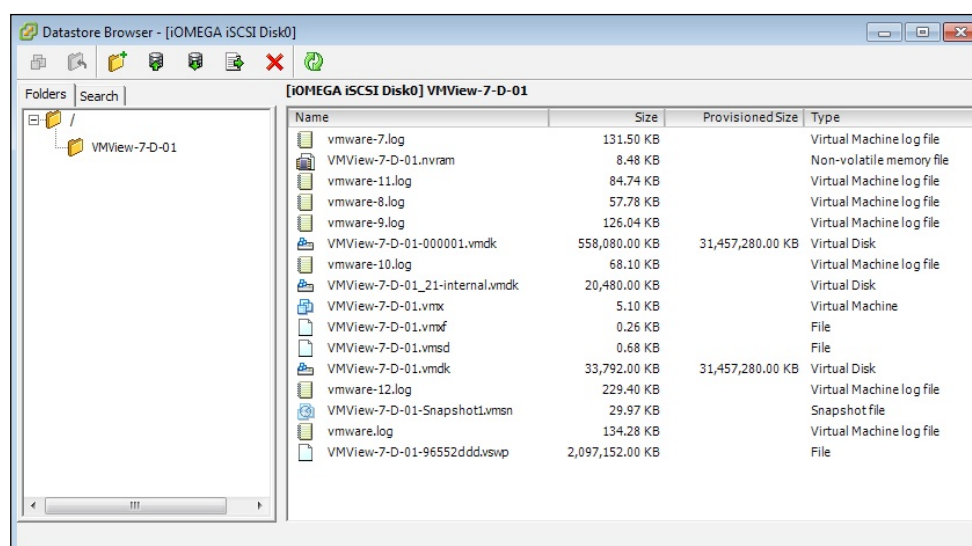
The internal disk is the only disk created by VMware View that is not thin provisioned. Its size is so small that being thick provisioned doesn't change the capacity requirements in the solution.

Delta/differential disk

The following screenshot demonstrates folders and files created in the VM folder to host a linked clone desktop. In the following example, the delta disk is VMView-7-D-01-000001.vmdk.

After the customization process is complete, the VM is shutdown and View Composer takes a snapshot of the linked clone.

The following screenshot shows the folders and files created in the respective VM's folder:



Name	Size	Provisioned Size	Type
vmware-7.log	131.50 KB		Virtual Machine log file
VMView-7-D-01.nvram	8.48 KB		Non-volatile memory file
vmware-11.log	84.74 KB		Virtual Machine log file
vmware-8.log	57.78 KB		Virtual Machine log file
vmware-9.log	126.04 KB		Virtual Machine log file
VMView-7-D-01-000001.vmdk	558,080.00 KB	31,457,280.00 KB	Virtual Disk
vmware-10.log	68.10 KB		Virtual Machine log file
VMView-7-D-01_21-internal.vmdk	20,480.00 KB		Virtual Disk
VMView-7-D-01.vmx	5.10 KB		Virtual Machine
VMView-7-D-01.vmx	0.26 KB		File
VMView-7-D-01.vmsd	0.68 KB		File
VMView-7-D-01.vmdk	33,792.00 KB	31,457,280.00 KB	Virtual Disk
vmware-12.log	229.40 KB		Virtual Machine log file
VMView-7-D-01-Snapshot1.vmsn	29.97 KB		Snapshot file
vmware.log	134.28 KB		Virtual Machine log file
VMView-7-D-01-96552ddd.vswp	2,097,152.00 KB		File

After the snapshot is taken, data is no longer written to the base .vmdk file. Instead, changes are written to the delta disk. A delta disk will be created every time a snapshot is taken. It is important that any requirements to use snapshots are considered when defining the datastore size requirements.

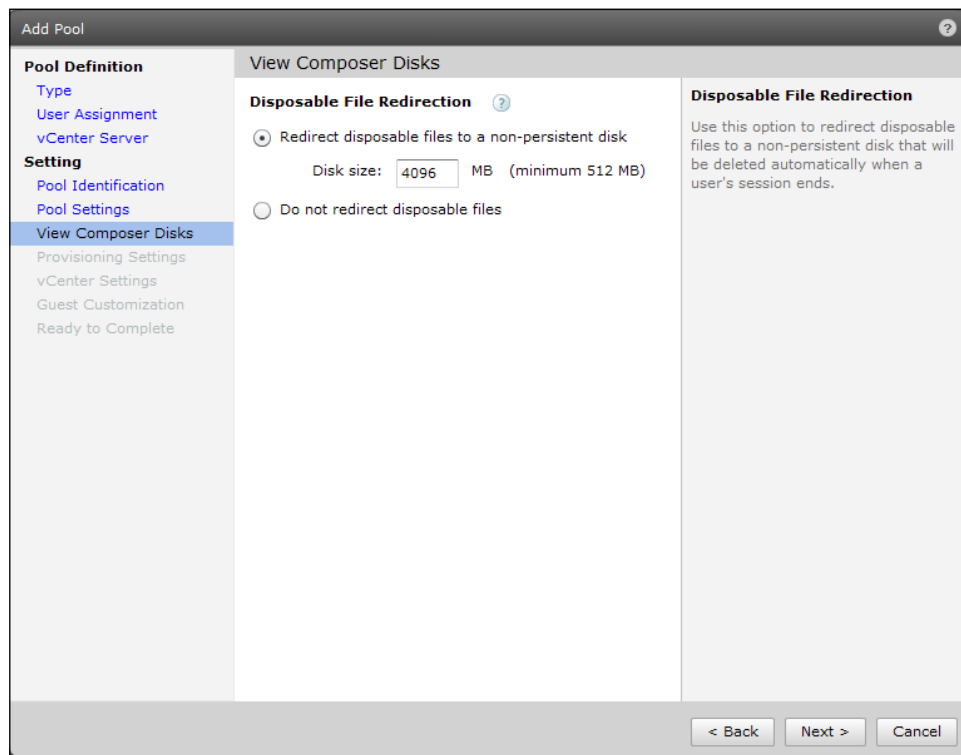
Disposable disk

VMware View allows for the creation of an optional fixed-size non-persistent disk for each virtual desktop. When disposable disks are assigned to a desktop pool, VMware View redirects Windows temporary system files and folders to a disposable disk.

Disposable disks are automatically deleted when the virtual desktop is powered off, refreshed or recomposed, meaning that temporary files are also deleted during these operations.

Disposable disks are also thin provisioned and will grow over time to the maximum size set during the desktop pool configuration.

The following screenshot shows the **Disposable File Redirection** configuration within the View Admin console:



The disposable disk is hardcoded to register itself as the first available drive on the Windows desktop. This behavior may cause some implications while trying to map network drives in Windows virtual desktops. Even when the CD-ROM is not in use, the first available drive letter would be E: because C: is taken by the OS and D: by the disposable disk.

The files that are commonly offloaded to disposable disks are Windows paging files, VMware log files, and temporary internet files.

Windows paging files






When Windows is constantly running out of physical memory, it will start to page memory to disk. This paging process will incur in block writes that will increase the size of the disposable disk. As a recommended approach, administrators should make sure that virtual desktops have enough virtual memory available to avoid disk paging. Disk paging has a negative impact on performance.

Temporary internet files

User temporary files are kept on the system or persistent data disk and this may include files written to %USERPROFILE%\AppData\Local\Temp and Temporary Internet Files. These are the temporary files that can grow very fast and consume disk space.

As the temporary files are offloaded to the disposable disk, the steady growth of the delta file is reduced. Instead of utilizing delta disks that will grow according to block changes, View Composer utilizes the disposable disk. However, it is important to size the disposable disks according to the virtual desktop and workload requirements. Once assigned to a virtual desktop pool, this setting cannot be changed through the View Manager.

The following screenshot shows a disposable disk with Windows temporary files:

Name ^	Date modified	Type	Size
 \$RECYCLE.BIN	7/09/2010 1:03 AM	File folder	
 System Volume Information	6/09/2010 12:51 AM	File folder	
 TEMP	7/09/2010 9:13 AM	File folder	
 pagefile.sys	6/09/2010 1:52 PM	System file	1,048,576 KB
 simvol.dat	4/09/2010 4:50 PM	DAT File	1 KB



When configuring a linked clone pool, make sure that the disposable disk is larger than the Windows paging file size plus overhead for temporary files.

Persistent disk

Persistent data disk is the new name for what used to be called **User Data Disk (UDD)** in previous releases of VMware View and maintains the similar characteristics to the UDD. Persistent disks were created to maintain a one-to-one relationship between users and virtual desktops.

When selected during the desktop pool configuration, the persistent disk is created and VMware View Agent makes modifications to the Windows Guest OS to allow the user profile to be redirected to this disk.

In VMware View 4.5 and later, persistent disks can be managed. The disk can be detached and reattached to virtual desktops. Note that this will only work if the disks are created in VMware View 4.5 or later. If a VMware View environment has been upgraded from earlier releases, these operations will not be available.

Just like disposable disks, persistent disks cannot be disabled or enabled once they are configured for the desktop pool. It is also not possible to change the size through the View Manager graphical user interface after the initial configuration. However, it is possible to change the persistent disk's size for virtual desktops being newly provisioned in the desktop pool if the administrator changes the pool settings.

When configuring persistent disks, you should make sure that the size of the disk is adequate for your users. If Active Directory Folder Redirection or VMware View Persona Management is not in use, the user profile size could be several gigabytes in size.

As a general rule, if you are using persona management or Windows roaming profiles, make sure the disk is large enough to cater to the user's roaming profiles, or apply quotas to roaming profiles.

The following screenshot demonstrates the persistent disk selection screen during the desktop pool configuration. It shows the configuration of persistent disks within the View Admin console:

Add Pool

Pool Definition

- Type
- User assignment
- vCenter Server

Setting

- Pool ID
- Pool Settings
- View Composer Disks**
- Provisioning Settings
- vCenter Settings
- Guest Customization
- Ready to Complete

View Composer Disks

Persistent Disk

- ☒ Redirect Windows profile to a persistent disk
 - Disk size: 2048 MB
 - Drive letter: D
- ☐ Store Windows profile in OS disk

Disposable File Redirection ⓘ

- ☒ Redirect disposable files to a non-persistent disk
 - Disk size: 2048 MB (minimum 50 MB)
- ☐ Store disposable files in the OS disk

Redirect Windows Profile

Windows profiles will be redirected to persistent disks, which are not affected by View Composer operations such as refresh, recompose, and rebalance. These Persistent disks can also be detached and subsequently attached to other vCenter Server virtual machines.

Disposable File Redirection

Use this option to redirect disposable files to a non-persistent disk that will be deleted automatically when a user's session ends.

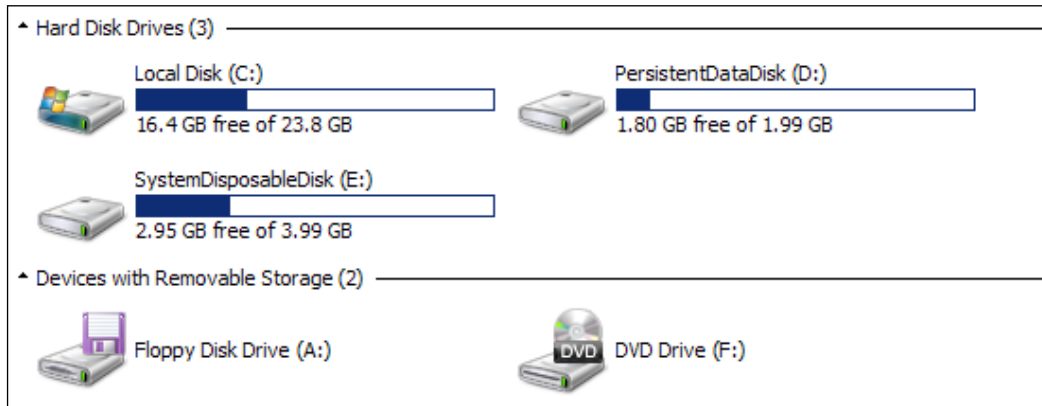
Supported Features

- ✓ Local Mode
- ✓ PCoIP
- ✓ Storage savings
- ✓ Detach and Attach persistent disk
- ✓ Persona management

< Back Next > Cancel

In the Windows OS, the persistent disk and the disposable disk can be seen in Windows Explorer. Important folders will be locked and users cannot delete them. However, through the use of Windows Group Policy, it is possible to hide the drives while still making them available for use.

The following screenshot shows the various disks, as seen within the guest vDesktop:



The persistent disk's drive letter is selected during the desktop pool configuration process and the content is similar to the following screenshot. In the **Users** folder, the profile folders and settings are found:

Name ^	Date modified	Type	Size
personality	1/04/2011 1:01 PM	File folder	
personality.bak	1/04/2011 12:17 PM	File folder	
Users	1/04/2011 10:41 AM	File folder	

Storage overcommit

With the introduction of linked clone technology and the ability to specify when each virtual desktop is refreshed or recomposed, there is an opportunity to specify how much storage should be overcommitted to help reduce storage consumption.

Let's assume that not every virtual desktop will utilize the full provisioned storage at the same time, thereby leaving a gap for storage utilization and overallocation (overcommit).

During the desktop pool provisioning process, the administrator has the option to select "Refresh OS Disk after log off" with one of the following options:

- **Never:** If the **Never** option is selected, then virtual desktops will never execute the delta disk Refresh operation. The delta disk will grow with every block change up to the limit of the disk itself. If the disk size defined for the virtual desktop is 40 GB, this is the limit. When 40 GB is reached, then vSphere VMFS starts reutilizing the blocks just like it does with full clones. You will not run out of disk space in this case.
- **Always:** If the **Always** option is in use, then virtual desktops will be refreshed every time a user logs off from the desktop. Assuming that only a few gigabytes have been added to the delta disk during use, they will then be recuperated when the virtual desktop is refreshed.
- **Every x number of days:** If the **Every x number of days** option is selected, then virtual desktops will be refreshed on the number of days defined, independent of the utilization of the delta disk. Delta files grow over time based on a number of factors that include Windows and application utilization. Therefore, while selecting this option, it is important to understand how big the delta can get during that period, so you are able to size datastores accordingly.
- **At y percent of disk utilization:** If the **At y percentage disk utilization** option is selected and *y* is set to 50 percent, then the virtual desktop will be refreshed when half of the total provisioned storage is utilized by the delta. This calculation does not include additional disks such as persistent or disposable. If a virtual desktop has been created with a total disk size of 40 GB, the Refresh operation would happen when the user logs off and the delta disk utilization is more than or equal to 20 GB.

An important aspect to remember is that linked clones start at a fraction of its full provisioned size. The storage capacity savings provided by VMware View Composer through the Refresh operation allow administrators to decide how the available storage capacity should be utilized until the storage is fully occupied. As an example, the administrator who selected the **Always** option knows that delta files, on average, will grow up to 300 MB while the desktop is in use during business hours.

Based on the storage utilization, it is possible to enforce the placement of more virtual desktops per datastore than would be possible with full clone virtual desktops. This is called the **storage overcommit level**.



VMware View does not allow administrators to configure the maximum number of linked clones per datastore and the limitation on the number of desktops comes from the datastore size. It is critical to size datastores appropriately to support the required number of desktops, yet be compliant with VMware View and View Composer maximums and limits.

Typically, the overcommit level is defined based on how virtual desktops are used. If a desktop pool with floating assignment has desktops that have **Always** Refresh option after logoff, storage consumption will be low and you may set the overcommit to **Aggressive**. However, if virtual desktops are not frequently refreshed, you may prefer to set it to **Conservative**.

Storage overcommit level options

It is possible to define different overcommit levels among different types of datastores to address different levels of capacity, performance, or availability provided. For example, a NAS datastore may have a different overcommit level than a SAN datastore; in the same way an SSD datastore can have a different overcommit level than an FC datastore.

Option	Storage overcommit level
None	Storage is not overcommitted.
Conservative	Four times the size of the datastore. This is the default level.
Moderate	Seven times the size of the datastore.
Aggressive	Fifteen times the size of the datastore.



It is recommended practice to always match vSphere datastores with LUNS or Exports on a one-by-one basis. Administrators should avoid using large storage LUNS backed by multiple datastores.

The number of linked clones per datastore defined by the storage overcommit level is based on the size of the Parent VM. Based on a 30 GB VM and a 200 GB datastore, VMware View would be able to fit approximately 6 full clone virtual desktops. However, if using overcommit level 7 (Moderate), VMware View would be able to fit approximately 42 desktops.

VM size (GB)	Datastore size (GB)	Overcommit level	Number of full clone VMs	Number of linked clone VMs
30	200	4	6	24
30	200	7	6	42
30	200	15	6	90



It's possible to run out of storage capacity. When the storage available in a datastore is not sufficient, VMware View will not provision new desktops, however, the existing linked clone desktops will keep growing and eventually fill up the datastore. This situation is more common with overcommit level set to **Aggressive**.

To make sure that linked clones do not run out of disk space, administrators should periodically refresh or rebalance desktop pools to reduce the linked clone footprint to its original size.

The following screenshot demonstrates storage overcommit selection during the desktop pool provisioning or configuration process:

Select Datastores

Select the datastores to use for this pool. Only datastores that can be used by the selected host or cluster can be selected.

The table of minimum, maximum and 50% values only reflects the amount of storage needed for new virtual machines. It does not factor in the amount of storage space required for the disk growth of current virtual machines

☐ Show incompatible datastores ☐ Local datastore ☒ Shared datastore

	Datastore	Capacity (GB)	Free (GB)	Type	Desktops	Use For	Storage Overcommit
<input type="checkbox"/>	esxi01_data	35	29.55	VMFS	0		
<input type="checkbox"/>	esxi02_data	35	34.55	VMFS	0		
<input checked="" type="checkbox"/>	iOMEGA iSCS	249.75	182.25	VMFS	0	Linked clones	Aggressive
<input checked="" type="checkbox"/>	iOMEGA iSCS	249.75	184.48	VMFS	1	Linked clones	Moderate
<input checked="" type="checkbox"/>	Openfiler iSC	49.75	41.39	VMFS	1	Replica disks	None

☐ Use different datastores for OS disks and View Composer persistent disks

☒ Use different datastore for View Composer replica disks

Data Type	Selected Free Space (GB)	Min Recommended (GB)	50% utilization (GB)	Max Recommended (GB)
Linked clones	366.73	0.00	0.00	0.00
Replica disks	41.39	0.00		0.00

OK Cancel

Storage protocols

VMware View is supported by the VMware vSphere, and therefore supports multiple storage protocols for storing data. VMware vSphere is capable of using Fiber Channel, iSCSI, **Fiber Channel over Ethernet (FCoE)**, and NFS.

The main considerations for protocol choice for VMware View are maximum throughput, VMDK behavior, and the cost of reusing existing versus acquiring new storage infrastructure. These considerations affect network design and performance.

The intention of this section is not to cover each protocol or how they perform in a VDI environment. The numbers in the following table are based on VMware View 5 and vSphere 5 and are intended to help with the decision on the storage protocol to be used:

	Fiber Channel	iSCSI	FCoE	NFS
Type	Block	Block	Block	File
VAAI	Yes	Yes	Yes	Yes
Transmission rate	4 Gbps or 8 Gbps	Multiple 10 Gbps	Multiple 10 Gbps	Multiple 10 Gbps
Maximum number of hosts	8	8	8	8
LUNs/Exports per host	256	256	256	256
Clones per datastore	64 to 140	64 to 140	64 to 140	Not validated

Maximums and limits

Designing a large-scale VMware View solution is a complex task. The same challenges faced in large deployments may be faced in small deployments if VMware validated maximums and limits are not observed.



It is recommended to use a conservative approach when sizing the VDI environment.

There are tools to help administrators to understand requirements and constraints from graphics, CPU, memory, and storage perspectives. Other tools help to calculate the infrastructure size based on the number of virtual desktops, average IOPS, memory size, percentage of shared memory, percentage of used memory, percentage of read/write IOPS, and so on.

No matter what results these tools provide, the VDI architect should always ensure that the numbers are within VMware vSphere and VMware View maximums and limits.

64 – to 140 linked clones per datastore (VMFS)

For FC arrays with support for **vStorage APIs for Array Integration (VAAI)**, the maximum number of linked clones per datastore is 140. The VAAI primitives that augment the number of virtual desktops per datastore is called **hardware assisted locking** or **Atomic Test and Set (ATS)**.

The vSphere VMkernel has to update VMFS metadata for operations involving the virtual desktops stored in the VMFS. Updates to metadata occur as a result of powering on/off virtual desktops, suspending/resuming virtual desktops, and various other operations.

In a VDI environment, that would mean the VMkernel may have to update metadata for many hundreds of virtual desktops, and would therefore have to lock the entire VMFS in order to update its metadata just to power on a single virtual desktop. That operation takes no more than a few milliseconds, but does become problematic when powering on many virtual desktops simultaneously.

Hardware assisted locking, available with vSphere 4.1 and compatible vendor array code, allows the VMkernel to lock metadata at the block level within the VMFS stored on the array and allow multiple operations to occur simultaneously within the VMFS, which in turn allows many desktop VMs to be powered on at the same time.

250 linked clones per datastore (NFS)

For datastores backed by **Network File System (NFS)**, there is no limitation on the number of virtual desktops per datastore because NFS doesn't present the same SCSI reservation problems. However, to date there is no official validation from VMware on the maximum number of virtual desktops that can be hosted in a single datastore backed by NFS. The recommendation thus far is to keep the number of virtual desktops per NFS datastore to below 250.

32 full – clones desktops per datastore (VMFS)

It would be good if it was possible to answer the "Why?" question for every maximum and limit dictated by the documentation. Some of the widely known limits are either based on quality assurance tests or field experience. Some other limits are based on best practices that have been set for prior technology, but due to the lack of additional tests they remain valid.

We tried to understand where the 32 virtual desktops limit when using full clone came from, and the answer we received from VMware was that there was no good answer, except that the limitation is based on server workloads, SCSI reservations, and storage administrators not doing a good job at sizing the infrastructure.

If this limit was set for virtual server workloads, it could have been set years ago when storage architectures didn't have features such as advanced caching and VAAI support, among others.

8 hosts per vSphere cluster with View Composer

VMware View will stop the provision of new virtual desktop if the number of hosts in a cluster with View Composer surpasses 8. The behavior is hard-coded into View Composer. However, the source of the limitation lies in the VMFS layer.

VMFS structure only allows for a maximum of 8 hosts to access to read or write a single VMDK file. In a linked clone implementation, all hosts in a cluster may have virtual desktops reading storage blocks from the same replica disk.



At the time of writing, VMware was working to validate up to 16 hosts per vSphere cluster with View Composer.

1,000 clones per replica

The number of clones per replica also determines the number of linked clones that may coexist in a single desktop pool. This number is resulting from VMware's QA validation labs, however this is a soft limit and despite not being recommended, it can be increased.

In previous releases, the VMware View validated limit was 512 virtual desktops per replica or desktop pool. This limit was a result from a maximum of 64 linked clones per datastore multiplied by 8 hosts per vSphere cluster. ($64 * 8 = 512$).

Storage I/O profile

The I/O storage profile produced by each virtual desktop is entirely dependent on which type of Windows OS is in use, the applications deployed, and even how each user individually interacts with the environment.

IOPS (pronounced as **eye-ops**) is a common performance measurement used to benchmark computer storage devices such as **hard disk drives (HDDs)**, **solid state drives (SSDs)**, and **storage area networks (SANs)**. As with any benchmark, IOPS numbers published by storage device manufacturers do not guarantee real-world application performance.

According to Wikipedia, predictions of what the average virtual desktop I/O profile will likely be is one of the most difficult tasks when designing a VDI solution. The reason for that is the lack of information about the workload that will be running in each one of the virtual desktops at the design time.

It is possible to use pre-trended numbers as a baseline; however, despite the indication of what the workload would likely be, it could be a point out of the curve in some cases.

In a nirvana scenario, a VDI pilot project has been operational for a little while and data can be collected and trended appropriately.

A few metrics must be taken into consideration to size storage correctly. They are as follows:

- **Storage size:** How much storage capacity is required?
- **LUN size:** How many LUNs and/or datastores are required?
- **Tier type:** What type of disk is required and what is the disk placement?
- **IOPS (cmd/s):** What is the number of I/O commands per second?
- **Read/write ratio:** What is the read and write ratio?

The first three items in the list may be calculated without major understanding of the I/O workload; however, it would require knowledge about the virtual desktop storage capacity utilization.

The real problem lies with the I/O per second and the read/write I/O pattern. Without those values, storage architects/administrators will probably not be able to provision storage with the performance that the virtual desktop infrastructure requires.

IOPS, also known as the **disk I/O profile**, will differ for each type of Windows OS. The profile is also dependent upon the type and number of applications deployed, including services running in the OS. The I/O profile is also dependent on how users interact with their virtual desktops. VMware and partners have validated I/O profiles that can be used as a baseline. It is highly recommended that you find out the correct I/O profile for your particular VDI environment.

VMware View documentation establishes some I/O baselines:

- **Light (5 IOPS):** Light users typically use e-mail (Outlook), Excel, Word, and a web browser (Internet Explorer or Firefox) during the normal workday. These workers are usually data entry operators or clerical staff.
- **Heavy (15 IOPS):** Heavy users are full knowledge workers using all the tools of the light worker (Outlook, Excel, Word, Internet Explorer, and Firefox) and also working with large PowerPoint presentations and performing other large file manipulations. These workers include business managers, executives, and members of the marketing staff.

Another study performed by PQR Consultants (Herco van Brug) demonstrates I/O profiles as follows:

	Windows XP	Windows 7
Light	3 to 4	4 to 5
Medium	6 to 8	8 to 10
Heavy	12 to 16	14 to 20

The *VDI & Storage – Deep Impact v1-25 hot!* article can be found at <http://www.virtuall.nl/whitepapers/solutions>.

It is common to talk about average IOPS per virtual desktop; however, when sizing the VDI solutions, it is crucial that the peaks are also catered for. Otherwise the storage infrastructure will be under heavy stress and will not be able to deliver the required IOPS and throughput.

Common scenarios where high performance and high throughput are required are during boot and login storms. As an example, a Windows 7 desktop can generate up to 700 IOPS during boot time. Another good example is if the "Refresh on logoff" option is used in conjunction with floating pools, it is common to see utilization peaks at the end of the work shift when business users start to log off.

Now, we have an existing paradigm where, from a cost perspective, storage infrastructure should be sized for the average performance requirements over time, but from a performance perspective it should be sized for those peaks. For that reason, storage vendors have implemented their own proprietary caching solutions to optimize storage arrays to deal with the high peaks yet volatile VDI I/O requirements.

Most architecture documents or white papers published will demonstrate some divergences on these numbers. If you want to run your own I/O benchmarking during the pilot phase, use storage array admin tools, VMware vCenter client, or tools, for example, vscsiStats that will provide you with a much more granular overview.

Read/write I/O ratio

The read/write I/O ratio will determine how many disks are required to support the VDI workload in a RAID configuration. Now we will learn how critical it is to understand the read and write I/O ratio to allow administrators to properly size storage arrays from a frontend (storage processors) and backend (disks) standpoint.

In the last topic, we talked about the total number of IOPS that a virtual desktop produces during boot, log in, and steady state utilization. IOPS may be read or written. Every time a disk block is read, we have a read I/O and every time a block is written, we have a write I/O.

The Windows operating systems are by nature very I/O intensive, and most of those I/Os are write I/Os. This is actually a very interesting subject. During workload simulations, it is possible to identify that Windows is constantly issuing more write than read I/Os. However, the most interesting fact is that even when Windows is idle, it still produces more write than read I/Os.

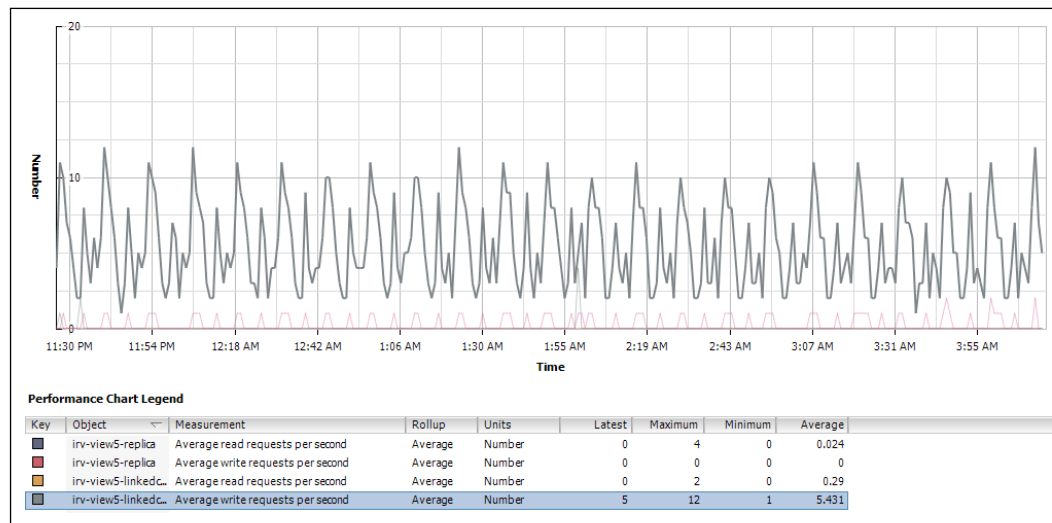
The same study performed by PQR Consultants (Herco van Brug) says that:

"The amount of IOPS a client produces is very much dependent on the users and their applications. But on average, the IOPS required amount to eight to ten per client in a read/write ratio of between 40/60 percent and 20/80 percent. For XP the average is closer to eight, for Windows 7 it is closer to ten, assuming the base image is optimized to do as little as possible by itself and all I/Os come from the applications, not the OS."

During an experiment conducted by Andre Leibovici and published in his blog at <http://myvirtualcloud.net/?p=2138>, it was possible to clearly identify the intensive write I/O pattern in contrast to the read I/Os during a Login VSI workload generation.

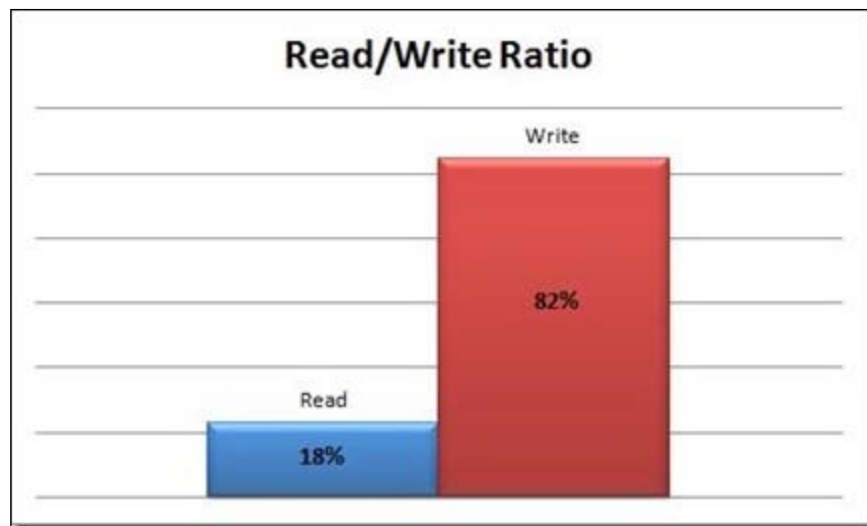
Sizing the Storage

The following screenshot shows I/O testing performed at <http://myvirtualcloud.net/>:



VMware View Reference Architecture documentation points us to read and write ratios with the following magnitudes: 70/30, 60/40, and 50/50. However, it's not uncommon to see VDI workloads with 10 percent reads and 90 percent writes.

The following diagram shows an illustration of the read/write ratio from real production data:



The reason we are focusing so much on the I/O pattern is because it determines how many drives are required to support the VDI workload. There are numerous proprietary technologies that reduce the impact of the I/Os on the physical drives. However the methodology to calculate the number of Input/Outputs Per Second required will not change.

In many cases, architects will be asked to design a solution without knowing what the I/O profile will look like. For most of those cases there is an ongoing pilot. It's very common for organizations to try to understand costs before actually going to a pilot and that's what makes it a difficult task to guess IOPS. If all organizations went first for a small pilot and then decided to buy the whole infrastructure to support the VDI solution, we would be living in an ideal world.

If you are designing VDI architecture without knowledge of the I/O profile, you should be extremely conservative to avoid undersizing the storage solution. If this is the case, you should use the Heavy I/O profile for all virtual desktops with read and write ratios of 20 reads and 80 writes. Hopefully, you will not be in this situation.

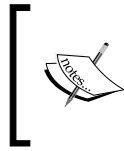
The RAID type selected will determine the performance and number of hard drives required to support the workload based on the amount of IOPS and read/write ratio. When sizing the storage infrastructure, the RAID group selected will add a write performance penalty due to the requirements to stripe the data and record the parity across disk drives. The read I/Os do not suffer a penalty for different types of RAID groups.

The most common RAID types utilized with VDI workloads are RAID 5, 6, and 10:

- RAID 10 adds a write penalty of 2
- RAID 5 adds a write penalty of 4
- RAID 6 adds a write penalty of 6

The preceding numbers can be tabulated as follows:

RAID level	I/O impact	
	Read	Write
RAID 0	1	1
RAID 1 (and 10)	1	2
RAID 5	1	4
RAID 6	1	6



It could be argued that for RAID 10, the read impact is 0.5 as the Windows operating system is able to read the same block off two disks at the same time, or read half of one disk and half of the other. Therefore, we get twice the read performance.

The formula commonly used to calculate these penalties is as follows:

$$\text{VM I/O} = \text{VM Read I/O} + (\text{VM Write I/O} * \text{RAID Penalty})$$

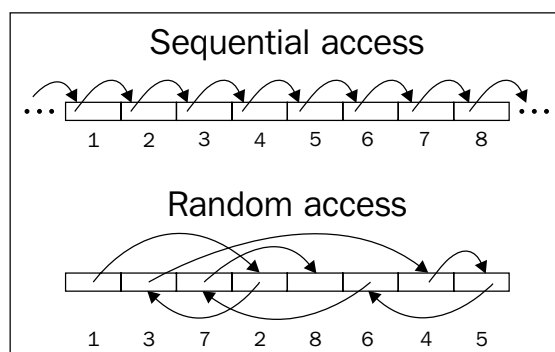
Other important information for architecting a VDI solution is that Windows operating systems predominantly have small random I/Os. Jim Moyle, in his paper *Windows 7 IOPS for VDI: Deep Dive* at http://jimmoyle.com/wordpress/wp-content/uploads/downloads/2011/05/Windows_7_IOPS_for_VDI_a_Deep_Dive_1_0.pdf, defines the Windows nature to generate small I/Os:

"This is due to how Windows Memory works, memory pages are 4 K in size, as such windows will load files into memory in 4 K blocks, this means that most of the read and write activity has a 4 K block size. Windows 7 does try and aggregate sequential writes to a larger block size to make writing a more efficient process. It will try and aggregate the writes to up to 1 MB in size. The reason for this is that again Windows is expecting a local, dedicated spindle and spinning disks are very good at writing large blocks."

Windows operating systems constantly read and write information in blocks with different disk placement, and the native user interaction is another reason for the behavior. Random access with small blocks is a time consuming task for mechanic disk drives and the number of operations per second (IOPS) for each drive is very limited. For this reason, it is important to utilize RAID groups to achieve the required I/O throughput.

SSDs provide excellent read performance, but also have limitation for small, random write I/Os. Nonetheless, they provide better performance over disk drives, at a much higher cost.

The following diagram shows an illustration showing the difference between sequential access and random access:



Storage tiering and I/O distribution

Previously in this chapter, we discussed VMware View tiered storage and the ability to assign different datastores or exports to different types of disks. Now, we will discuss how those tiers interact with the storage infrastructure from an I/O perspective.

VMware View 4.5 introduced the ability to select a dedicated datastore, where replica disks are stored. The VMware View Architecture Guide recommends that this datastore should be served by a pool of SSDs. SSDs generally provide a larger amount of Input/Outputs Per Second and throughput.

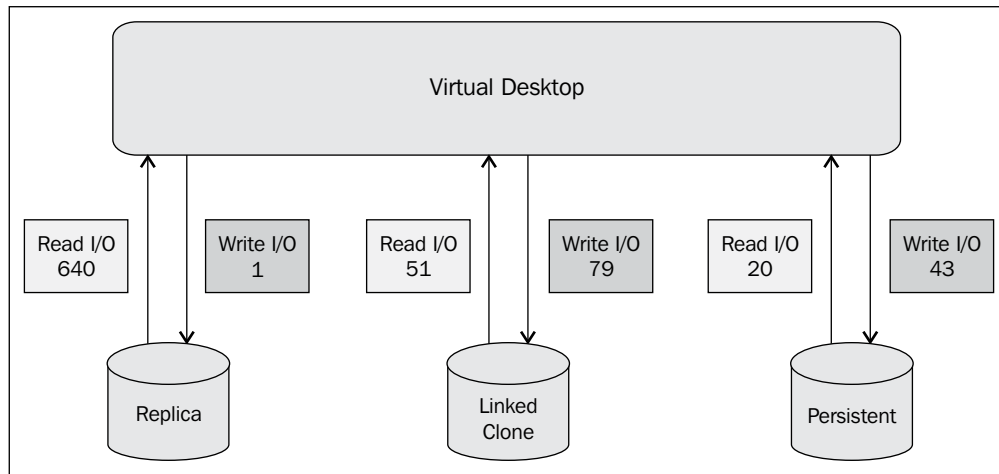
As mentioned earlier, common scenarios where high performance and high throughput are required are during boot and login storms, and during large scale application deployment to users or AV updates. As an example, Windows 7 can generate up to 700 IOPS during boot time. Another good example is if the "Refresh on logoff" option is used in conjunction with floating pools, it is common to see utilization peaks at the end of the work shift, when business users start to logoff.

The total amount of IOPS generated by a virtual desktop is a combination of the number of read I/Os in the replica disk plus read and write I/Os on all other disks. During the different utilization phases, the virtual desktop performs differently and requires a different number of I/Os and a different read/write I/O pattern for each individual tier.

The best way to understand how many I/Os are required for power on, customization, and first boot is to find out the averaged maximum I/O per datastore. The reason for this is that each storage tier will have different performance requirements.

The following diagram is an illustration showing the IOPS breakdown. It demonstrates the number of IOPS generated by a virtual desktop from the first power on operation to its first boot. The operations involved are as follows:

1. Power on.
2. Customization.
3. First boot.



The same numbers from the preceding diagram can be demonstrated in a percent style per storage tier. The following diagram shows a table that provides great visibility of what is happening with the virtual desktop during its creation process:

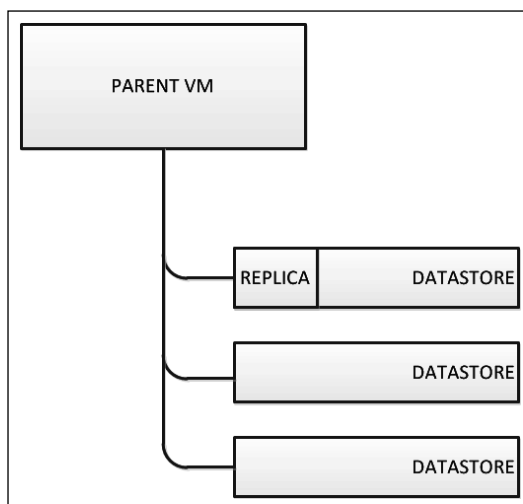
Replica		Linked Clone		Persistent Disk	
641 IOPS		130 IOPS		63 IOPS	
Read	Write	Read	Write	Read	Write
99.8%	0.2%	39%	61%	32%	68%
640	1	51	79	20	43



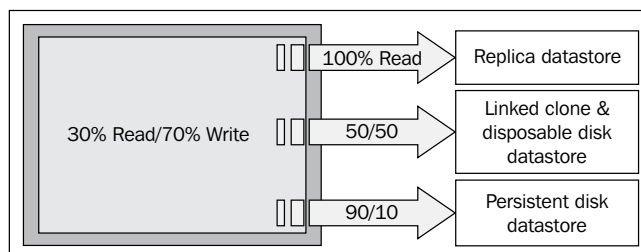
Important: Please remember that those numbers may be completely different in your VDI environment.

Understanding how many virtual desktops will be booting, logging on, or working simultaneously is critical to correctly design the tier supporting replica disks. If the Dedicated Replica Datastore option is selected, it is even more critical that this single disk tier supporting the replica disks can efficiently deliver the performance required, considering that up to 1,000 virtual desktops may be using that single replica disk simultaneously.

The following diagram is an illustration showing the use of a Dedicated Replica Datastore option:



Getting to the exact number of IOPS required for each tier of the storage solutions can be a tireless task. To provide you with an insight into how complex this can get, imagine a linked clone virtual desktop making use of a disposable disk and persistent disk. Assume that the replica disk is hosted in the dedicated replica datastore, the linked clone and the disposable disk on a different datastore, and the persistent disk on yet another datastore. The following diagram demonstrates this scenario. It is an illustration showing the breakdown of I/O when using the various virtual disks of VMware View:



As can be observed in the preceding diagram, in this scenario the virtual desktop is generating 30 percent read I/O and 70 percent write I/O. Let's assume that the total I/O for the virtual desktop is 20; then we have 6 read I/Os and 14 write I/Os.

We know replicas are 100 percent read; however, unless we scrutinize the replica disk it's not possible to know how many of the 14 read I/Os are actually being issued against the disk. Read I/Os could also be issued to the linked clone, disposable or persistent disk.

For the tier supporting linked clones and disposable disks, and for the tier supporting the persistent disk, we will also have a different number of read and write I/Os that are a small percentage from the 6 read I/Os and 14 write I/Os produced by the virtual desktop.

As you can see, the total number of operations produced by a virtual desktop will often be split across multiple tiers and datastores. The best time to gather those numbers is during the VDI pilot. VMware has the ideal tools for the job—esxstop and vscsiStats.

The truth is that there is no magic formula to help you to get to the exact I/O profile other than analyzing an existing environment's usage patterns. Plan your storage for performance and whenever possible utilize real workload data to calculate the environment. Pre-trended data from white papers and reference architecture guides may give you a baseline; however, they may not apply to your workload and you may end up with an undersized or oversized infrastructure.

The most common formulae for IOPS calculation are as follows:

Replica tier (read I/O only) = (Concurrent Boot VMs * Peak Boot IOPS) +
(Concurrent VMs - Concurrent Boot VMs) * (Replica Steady State IOPS)

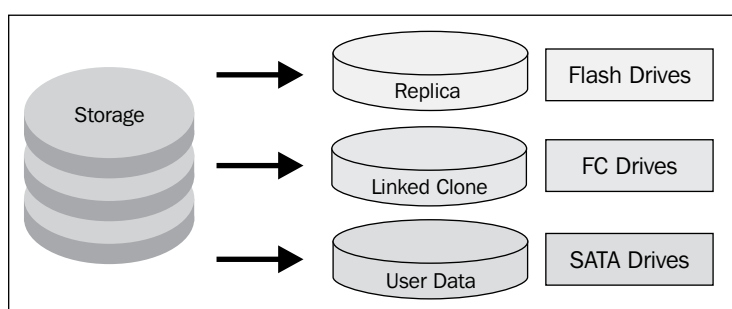
All other tiers = (VM Read I/O + (VM Write I/O * RAID Penalty)
* concurrent VMs

Paul Wilson from Citrix has created an attention-grabbing complex model based on peak IOPS, steady state IOPS, and estimated boot IOPS that takes into consideration the launch rate and desktop login time. His article can be found at <http://blogs.citrix.com/2010/10/31/finding-a-better-way-to-estimate-iops-for-vdi>.

Disk types

A common question is related to the type of disk that should be utilized for each storage tier. That's a complex discussion that you will need to have with your storage administrator or vendor. Most intelligent storage arrays provide some type of acceleration or caching mechanism that will potentially reduce the storage backend requirements. With the reduction of the requirements, disks with higher capacity and lower performance may be used.

The following diagram is an illustration showing the virtual disk-to-disk type relationship typically used by storage providers:



The most common type of disks used for VMware View deployments are as follows:

- **SSDs:** They provide the best throughput performance and may be leveraged for the replica tier requiring bursting capabilities during boot and login storms.
- **Fibre Channel and SAS:** They provide the best relationship between costs versus performance. Nowadays, Fibre Channel and SAS disks are probably the most common type of disks in enterprise environment and may be leveraged to host Linked Clone disks.
- **Serial Advanced Technology Attachment (SATA):** They provide the highest capacity and lowest cost, but with lowest performance. A common use for SATA disks is for persistent or user profile disk placement.

It's important to note that these are just conventional recommendations and your environment may pose different challenges or features. As an example, some scale-out NAS appliances may use very large pools of SATA disks and perform as well as Fibre Channel disks when the matter is throughput.

Capacity sizing exercises

Sizing the storage infrastructure correctly might be the difference between succeeding or failing a VDI rollout. Many deployments that have excellent performance during the pilot and initial production quickly start to run into storage contention issues because of the lack of understanding of the storage layer.

In the next section, we demonstrate a few sizing exercises for different VMware View implementations.

Sizing full clones

Let's see two scenarios for sizing full clones.

Scenario 1

The following are the parameters:

- **Desktops:** 1,000
- **Pool type:** Dedicated (full clones)
- **Guest OS:** Windows 7
- **RAM:** 2 GB
- **Disk size:** 40 GB
- **Disk consumed:** 22 GB
- **Overhead:** 10 percent

Parent VM

The Parent VM may be thin or thick provisioned and is usually powered off. If the Parent VM is thick provisioned, its size is similar to the creation of the disk plus log files. As an example, if the Parent VM disk size is set to 40 GB, this will be the approximate size of the Parent VM.

If the Parent VM is created using thin provisioning, its size is equal to the amount of storage utilized by the Windows operating system at the NTFS plus log files. As an example, if the Parent VM disk size is set to 40 GB but only 10 GB is used, the total size of the Parent VM will be approximately 10 GB plus log files.

There is no considerable performance improvement using thick provisioning over thin provisioning for the Parent VM, given that these are master images and won't be used unless a new replica disk is required.

Overhead

The VMware recommendation on storage overhead per datastore is at least 10 percent.

The following table explains the features and requirements:

Feature	Requirement	Reasoning								
VMs per datastore	32 VMFS	Recommended limit of full clones per datastore								
VM datastore size		Size based on the following calculations: <table><tr><td>Raw file size</td><td>40,960 MB</td></tr><tr><td>Log file size</td><td>100 MB</td></tr><tr><td>Swap file size</td><td>2,048 MB</td></tr><tr><td>Free space allocation</td><td>10 percent overhead</td></tr></table> Minimum allocated datastore size = (VMs * (raw + swap + log) + overhead) = (32 * (40,960 MB + 2,048 MB + 100 MB) + 137 GB) = 1.44 TB	Raw file size	40,960 MB	Log file size	100 MB	Swap file size	2,048 MB	Free space allocation	10 percent overhead
Raw file size	40,960 MB									
Log file size	100 MB									
Swap file size	2,048 MB									
Free space allocation	10 percent overhead									
Number of datastores	1 per 32 virtual desktops	Number of VMs/VMs per datastore = 1,000/32 = 32								
Total storage		Number of datastores * datastore size = 32 * 1.44 TB = 46 TB								

Comments

The following is a list of comments:

- As a safety number, it's an assumption that all full clone raw files will eventually achieve the full size (40 GB). Some administrators may prefer to use a fraction of the full utilization size to cut storage costs during calculation.
- In addition to the storage allocation required to support all full clone virtual desktops, it is important to set aside at least another one datastore per VMware View cluster to host the Parent VM and ISO images.

Scenario 2

Here are the parameters:

- **Desktops:** 2,000
- **Pool type:** Dedicated (full clones)
- **Guest OS:** Windows 7

- **RAM:** 2 GB
- **Disk size:** 32 GB
- **Disk consumed:** 22 GB
- **VM memory reservation:** 50 percent (1,024 MB)
- **Overhead:** 10 percent

The following table shows the features and requirements:

Feature	Requirement	Reasoning
VMs per datastore	32 VMFS	Recommended limit of full clones per datastore
VM datastore size		Size based on the following calculations: Raw file size 32,768 MB Log file size 100 MB Swap file size 1,024 MB Free space allocation 10 percent overhead Minimum allocated datastore size = (VMs * (raw + swap + log) + overhead) = (32 * (40,960 MB + 1,024 MB + 100 MB) + 108 GB) = 1.13 TB
Number of datastores	1 per 32 virtual desktops	Number of VMs/VMs per datastore = 2,000/32 = 33
Total storage		Number of datastores * datastore size = 32 * 1.12 TB = 36 TB

Comments

The following is a list of comments:

- The disk size for this scenario has been changed to 32 GB, reducing the overall storage footprint required. It's important to size virtual desktops appropriately for what the users require, not adding extra fat to the infrastructure.
- This scenario introduces 50 percent VM memory reservation, reducing the size of the `.vswp` file to half of the virtual desktop memory. In this scenario, the `.vswp` file is 1,024 MB.

Sizing linked clones

Sizing linked clone virtual desktops is to a certain extent more complex than sizing full clones due to the number of variables involved in the calculation. As mentioned earlier in this chapter, linked clone virtual desktops introduce new files and may work differently when the Dedicated Replica Datastore option is selected.

Parent VM

The Parent VM used with linked clones is similar to the one used with full clones, however, it includes VM snapshots that are used by View Composer to determine what baseline image is used to create replica disks.

Replica

Replica disks are created as thin provisioned clones from the Parent VM. If the Parent VM is set to a 40 GB disk size, the replica is equal to the amount of storage utilized by the Windows operating system at the NTFS plus the snapshot selected. As an example, if the Parent VM disk size is set to 40 GB but only 10 GB is used, the total size of the replica is approximately 10 GB.

Without making use of the Dedicated Replica Datastore option, for any desktop pool, a unique replica is created in each datastore assigned to the pool. If multiple snapshots are in use in a desktop pool, multiple replicas per datastore may be created if VMware View decides to use the datastore. It is common to have two or more snapshots in use at the same time in a single datastore, especially during Recompose operations.

Desktop pools * snapshots * datastores = number of replicas

$2 * 1 * 32 = 64$ replicas (2 per datastore)

$2 * 2 * 32 = 128$ replicas (4 per datastore)

If the Dedicated Replica Datastore option is in use, VMware View uses a single datastore to create all replicas for the desktop pool. The calculation of the number of replicas is also subject to the number of snapshots concurrently in use.

Desktop pools * snapshots * datastores = number of replicas

$2 * 1 * 1 = 2$ replicas

$2 * 2 * 1 = 4$ replicas

Scenario 1

Here are the parameters:

- **Desktops:** 5,000
- **Pool type:** Floating (linked clones)
- **Guest OS:** Windows 7
- **RAM:** 2 GB
- **Disk size:** 32 GB
- **Disk consumed:** 22 GB
- **Refresh on logoff:** 10 percent
- **Overhead:** 10 percent
- **VAAI:** Enabled

The following table explains the features and requirements:

Feature	Requirement	Reasoning
VMs per datastore	140 VMFS (VAAI)	Recommended limit of full clones per datastore
VM datastore size		Size based on the following calculations: Raw file size 3,277 MB Log file size 100 MB Swap file size 1,024 MB Free space allocation 10 percent overhead Minimum allocated datastore size = (VMs * (raw + swap + log) + overhead) = (140 * (3,277 MB + 1,024 MB + 100 MB) + 60 GB) = 661 GB
Number of datastores	1 per 140 virtual desktops	Number of VMs/VMs per datastore = 5,000/140 = 36
Total storage		Number of datastores * datastore size = 36 * 661 GB = 23 TB

Comments

The following is a list of comments:

- The desktop pool type is floating and that means that whenever the user has logged off, the virtual desktop will be refreshed. The important information here is to know for how long, on average, the users will remain connected to the virtual desktop and how much data they will generate on the delta disk during their usage.

If the desktops are designated to classes that will last 45 minutes, the chances are that the delta disk will present marginal growth. However, if the virtual desktop is used for a whole day, the chances are that the delta will grow a few hundred megabytes.

For this exercise, we are assuming that the delta disk will grow to a maximum size of 10 percent of the Parent VM, that is, 3,277 MB.

- VAAI is enabled in this scenario, enabling higher virtual desktop consolidation per datastore. The maximum number of virtual desktops per datastore supported with VAAI is 140.

Scenario 2

The following are the parameters:

- **Desktops:** 10,000
- **Pool type:** Persistent (linked clones)
- **Guest OS:** Windows 7 (64 bit)
- **RAM:** 4 GB
- **Disk size:** 32 GB
- **Disk consumed:** 22 GB
- **Refresh:** Never
- **Overhead:** 10 percent

The following table explains the features and requirements:

Feature	Requirement	Reasoning
VMs per datastore	100 VMFS	Recommended limit of full clones per datastore
VM datastore size		Size based on the following calculations:
		Raw file size 32,768 MB
		Log file size 100 MB
		Swap file size 4,096 MB
		Free space allocation 10 percent overhead
		Minimum allocated datastore size = (VMs * (raw + swap + log) + overhead) = (100 * (32,768 MB + 4,096 MB + 100 MB) + 360 GB) = 3.9 TB
Number of datastores	1 per 100 virtual desktops	Number of VMs/VMs per datastore = 10,000/100 = 100
Total storage		Number of datastores * datastore size = 100 * 3.9 TB = 390 TB

Comments

This scenario explores the idea of using linked clone but not having an internal policy to refresh virtual desktops so often. The result is similar to implementing full clones as the delta disks will grow to its full capacity, 32 GB in this case.

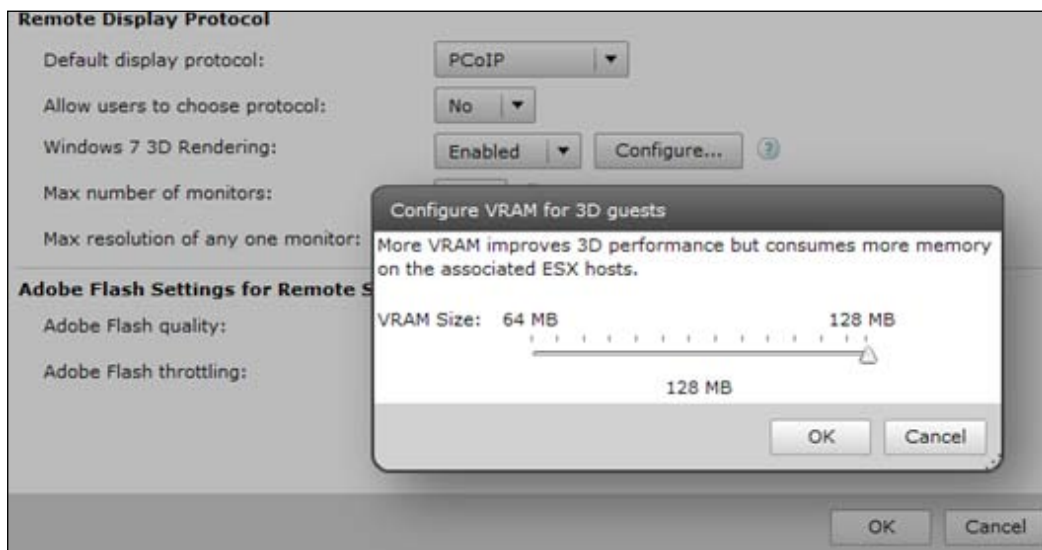
With 64-bit Windows 7 and 4 GB RAM without any VM memory reservation, the swap file is responsible for consuming 4 GB of storage capacity per virtual desktop. In total, the .vswap file will be consuming 40 TB; however, with only 20 percent memory reservation, this total would go down to approximately 31 TB of used storage space.

vSphere 5.0 video swap

VMware View has always automatically calculated video RAM based on resolution and color depth. Up to VMware View 4.6, only 24-bit color depth was supported and VMware published the vRAM overhead that each resolution type would require. vRAM overhead is part of the VM memory overhead in virtual machines running on ESXi. The other part of the overhead comes from the number of vCPUs and amount of RAM.

VMware View 5.0 introduces a 32-bit color depth and makes it the default option. On top of that, to allow 3D support, VMware introduced a new feature in View 5.0 that allows administrators to select how much video RAM should be assigned to virtual desktops.

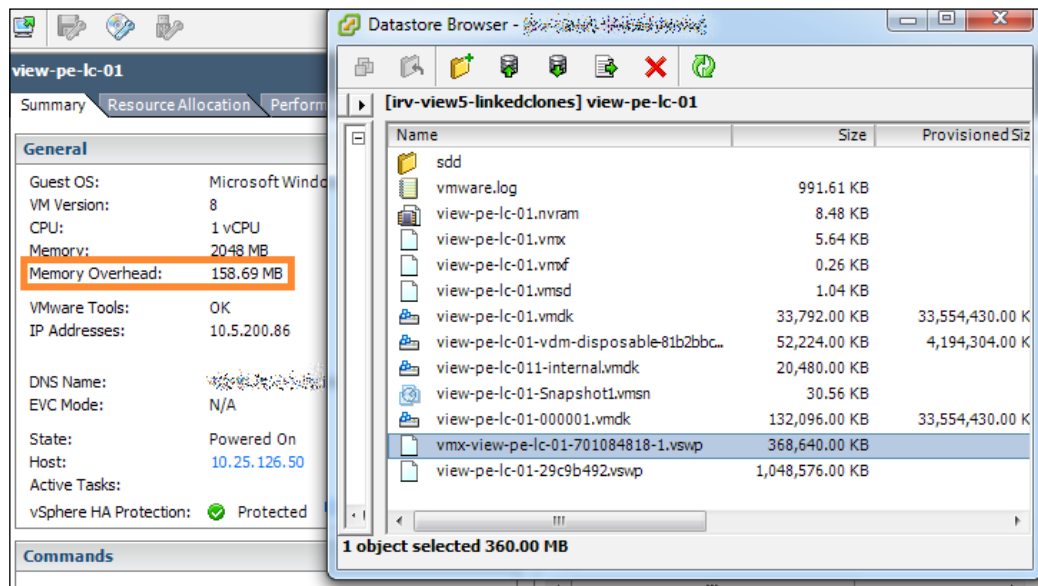
The following screenshot shows the configuration of vRAM for vDesktops requiring 3D capabilities:



The VMware View explanation of how to configure vRAM for 3D support is not very helpful and essentially says: the more vRAM, the more 3D performance will be available to vDesktop(s).

To support the new 3D option, vSphere 5.0 implements a second `.vswp` file for every virtual desktop created either using hardware version 7 or 8. This second `.vswp` file is dedicated to video memory overhead and will be used when the virtual desktop is under video resource constraint.

The following screenshot shows the second `.vswp` file; this is used for video memory:

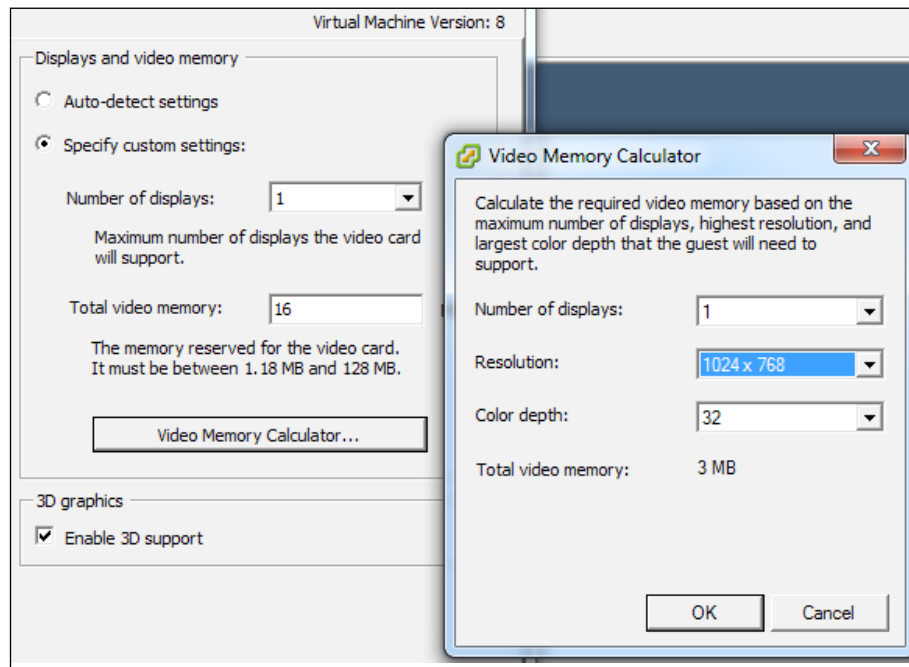


The total memory overhead is defined by the following factors:

- Number of virtual CPUs
- Amount of RAM
- Amount of vRAM (defined by number of displays, screen resolution, and color depth)
- 3D support

Memory overhead is nothing new to VMware administrators, as they used to calculate overhead based on vCPU, RAM, and vRAM. However, with the introduction of a video memory calculator, vSphere Client 5.0 provides an easy way to define the amount of vRAM required for a given video configuration.

The following screenshot shows advanced **Video Memory Calculator**:



The new video overhead `.vswp` file will affect storage footprint and datastore sizing. In order to understand the real impact, we have reverse-engineered the new video support option. The **swap (MB)** column demonstrates the total storage utilized by the video swap file, and the **overhead (MB)** column demonstrates the amount of RAM overhead utilized for each combination of vCPU, vRAM, and color depth:

vCPU	vRAM	video	overhead (MB)	swap (MB)
1vCPU	1GB	8	71.24	49.00
1vCPU	1GB	16	79.35	57.00
1vCPU	1GB	64	128.00	104.00
1vCPU	1GB	128	192.88	104.00
1vCPU	2GB	8	90.34	49.00
1vCPU	2GB	16	98.45	57.00
1vCPU	2GB	64	147.10	104.00
1vCPU	2GB	128	211.98	104.00
1vCPU	4GB	8	128.52	49.00
1vCPU	4GB	16	136.63	57.00
1vCPU	4GB	64	185.29	104.00
1vCPU	4GB	128	250.16	104.00
2vCPU	1GB	8	107.62	50.00
2vCPU	1GB	16	115.79	58.00
2vCPU	1GB	64	164.82	105.00
2vCPU	1GB	128	230.20	105.00
2vCPU	2GB	8	137.75	50.00
2vCPU	2GB	16	145.92	58.00
2vCPU	2GB	64	194.95	105.00
2vCPU	2GB	128	260.33	105.00
2vCPU	4GB	8	197.99	50.00
2vCPU	4GB	16	206.16	58.00
2vCPU	4GB	64	255.20	105.00
2vCPU	4GB	128	320.57	105.00



When 3D support is enabled, a 256 MB overhead is added to the secondary .vswp file. Therefore, if you are planning to use 3D, you should size datastores appropriately to accommodate this difference. This additional 256 MB will help virtual desktops not to run into video performance issues when executing 3D display operations. The 256 MB overhead is independent of how much vRAM you assigned to the virtual desktop in VMware View 5.0.

A datastore with 100 desktops will require additional 25 GB with 3D support enabled:

$$100 \text{ VMs} * 256 \text{ MB} = 25 \text{ GB}$$

The following screenshot shows a table that demonstrates the .vswp file (swap) resulting from 3D support enabled:

vCPU	vRAM	video	overhead (MB)	swap (MB)
1vCPU	1GB	8	71.24	305.00
1vCPU	1GB	16	79.35	313.00
1vCPU	1GB	64	128.00	360.00
1vCPU	1GB	128	192.88	360.00

When sizing for 3D support, you will need to ensure that datastores are appropriately sized for the amount of virtual desktops that will reside in the datastore, plus any additional 3D swap overhead.

Summary

Storage for virtualized environments already offers significant complexity from a design perspective. By adding VDI on top of a classic server virtualization solution, the additional storage technologies that were (for example, View Composer) potentially utilized, makes the storage design exponentially more complex. This chapter covered both the high-level aspects of storage design for VMware View solutions as well as the subtle intricacies that can often make or break a solution. Storage design, especially for solutions that are intended to scale over time, can require significant effort. It is important to not only understand fundamental storage principles before embarking on a VMware View storage design, but also understand the various types of virtual disks, as well as how the underlying guest uses its disk.

Now that all of the major design concepts have been covered, the next chapter will focus on backup and recovery. While a robust VMware View solution should be able to mitigate most outage scenarios, there may be times where a recovery action needs to be taken; as such, understanding the points of interest from a backup perspective as well as the recovery process is important as a design is implemented and handed over to an operations team.

9 Security

Whether deployed at a hospital, college, corporation, federal agency, or non-profit organization, security of the end device has become a critical component of any organization's data loss prevention and information assurance policies. With data loss events, for example, such as WikiLeaks or stolen laptops with social security numbers from organizations such as the U.S. Census Bureau, Ireland Department of Social and Family Affairs, or Anheuser-Busch, ensuring that sensitive data stays within the confines of the corporate infrastructure has gained much visibility.

In a traditional physical desktop model end users are issued desktops or laptops that contain writeable media (hard drives). These end devices store data such as the user's profile, copies of data from file shares, browser cache, plain text documents, images, spreadsheets, and other business and personal data.

Even with encryption of the hard drive on the end device sensitive data can still reside on the laptop (for example). With the availability of high-powered compute instances with processing power ideal for password cracking algorithms, such as Amazon EC2 GPU instances, cracking passwords and encryption algorithms can be offloaded to a public cloud. Therefore, the safest end device is a device that does not store any sensitive information, whether encrypted or not. For this reason, PCoIP zero clients (end devices that have a PCoIP chip from Teradici) are arguably more secure than thin clients (with a locked down operating system). Both are exponentially more secure than thick clients (traditional laptop or desktop).

The inherent security of VDI

With a properly designed VDI solution, all of the sensitive data resides in a secured data center versus living on hard drives in devices such as laptops and desktops. While it is possible to copy data within the vDesktop to, for example, a USB thumb drive plugged into the end device, it is also possible to prevent USB redirection of such devices.

In secure VDI implementations the only data that is typically transmitted is the visual and audio stream to deliver the desktop experience to the end device. This means that if an end user is using Microsoft Word to manipulate a document while connected to their vDesktop, the document does not live on his/her end device (for example, a laptop). Instead, it completely resides within the vDesktop running, ideally, within the data center. The visual representation of the desktop, including the visual display of Microsoft Word and the document are streamed down to the end device via the secured PCoIP protocol.

In a properly designed VDI solution, if an end device is broken, stolen, lost, or misplaced, the end user simply needs a new end device to connect back to their vDesktop. For example, if Lily has a PCoIP zero client that is no longer working, she can be issued a new zero client and can immediately resume working in the VDI. There is no re-imaging process for zero clients and Lily can quickly return to productive tasks.

Without VDI, she may have to wait for days for an end device to be repurposed, procured, or provisioned before she can return to productivity.

In addition, there are no data salvage actions that need to be performed because no data exists on the end device. In environments that employ the use of hot desking, or the practice of providing unassigned workspaces without reservations in an office environment, Lily could simply walk to an available workspace, log in, and reconnect to the VDI. Again, all of Lily's data resides in the data center.

Firewalls, zones, and antivirus

The basic fundamentals of securing a VMware View environment involve only allowing the specific ports and protocols absolutely necessary for a functioning VDI. In addition, it also involves the use of **Secure Sockets Layer (SSL)** (as opposed to unencrypted traffic over port 80) when available. In addition, requiring the use of PCoIP, as opposed to also allowing RDP connections, can further increase security in the environment.

Within a given VDI solution there are potentially several firewalls that come into effect. These firewalls include:

- **Windows OS firewall:** This firewall is used to restrict inbound and outbound traffic at the operating system layer
- **Network firewall (internal):** These firewalls are used to restrict traffic within the internal LAN environment

- **Network firewall (external/DMZ):** These firewalls are used to restrict traffic (typically) generated from the Internet
- **Virtual firewall:** These firewalls are used to restrict traffic across virtual port groups and virtual switches within the virtual infrastructure

The calculated use of firewalls helps create physical and virtual security enclaves known as **zones**. A virtual security zone is a group of network configurations, security policies, virtual machines, and other virtual infrastructure components allowed to freely communicate with each other according to the defined policies.

Virtual security zones have the following possibilities for cross-zone communication:

- **Permitted:** Virtual machines in Zone_A and Zone_B are able to freely communicate with each other based on a mutual trust relationship (not to be confused with technologies such as Active Directory trusts and relationships)
- **Restricted:** Virtual machines in Zone_A and Zone_B are able to communicate with each other along predefined ports and protocols only
- **Prohibited:** Virtual machines in Zone_A and Zone_B are not permitted to communicate with each other

One of the final pieces of the security matrix is antivirus protection for vDesktops. Antivirus protection ensures that malware does not penetrate and proliferate the physical and/or virtual desktop environment.

The fundamentals – firewall rules

For a more detailed list of ports and protocols, please see Christoph Harding's excellent article – *Firewall settings for a VMware View environment* on ThatsMyView.net found at: <http://www.thatsmyview.net/2011/04/24/firewall-settings-for-a-vmware-view-environment/>.

Source IP	Direction	Destination IP	Transport protocol	Port	Application protocol	Description
End user device	Inbound	View Security Server	TCP	443	HTTPS	Authentication and other communications
End user device	Both	View Security Server	TCP and UDP	4172	PCoIP	PCoIP handshake and data transfer
View Security Server	Inbound	View Connection Server	TCP	8009	AJP13	AJP-Data Traffic

Source IP	Direction	Destination IP	Transport protocol	Port	Application protocol	Description
View Security Server	Inbound	View Connection Server	TCP	4001	JMS	Java
View Security Server	Inbound	View Transfer Server	TCP	443	HTTPS	Communication with View Transfer Server
View Security Server	Both	View Agent	TCP and UDP	4172	PCoIP	PCoIP handshake and data transfer
View Security Server	Both	View Agent	TCP	32111		USB Redirection (if applicable)
View Connection Server	Outbound	Active Directory	TCP and UDP	389	LDAP	Active Directory Authentication and ADAM
View Connection Server	Both	View Connection Server	TCP	4100	JMSIR	Internal View Connection Server communication
View Connection Server	Both	View Connection Server	TCP	636	LDAPS	AD LDS
View Connection Server	Both	View Connection Server	TCP	1515		Microsoft Endpoint Mapper
View Connection Server	Both	View Connection Server	TCP	4001	JMS	Java
View Connection Server	Both	View Connection Server	TCP	8009	AJP13	AJP-Data Traffic
View Connection Server	Both	View Transfer Server	TCP	8009	AJP13	AJP-Data Traffic
View Connection Server	Outbound	View Transfer Server	TCP	443	HTTPS	Communication with View Transfer Server
View Connection Server	Outbound	View Transfer Server	TCP	4001	JMS	Java
View Connection Server	Outbound	View Transfer Server	TCP	4100	JMSIR	Internal communication

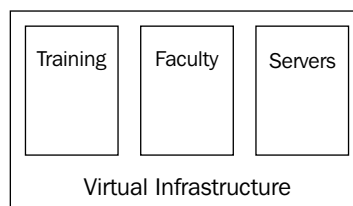
Source IP	Direction	Destination IP	Transport protocol	Port	Application protocol	Description
View Connection Server	Outbound	vCenter Server	TCP	18443	SOAP	View Composer communication
View Connection Server	Outbound	vCenter Server	TCP	443	HTTPS	vCenter communication
View Connection Server	Both	View Agent	TCP	4001	JMS	Java
End user device	Outbound	View Connection Server	TCP	443	SSL	Communication with View Connection Server for authentication and other activities
View Security Server	Inbound	View Connection Server	TCP	8009	AJP13	AJP-Data Traffic
View Security Server	Inbound	View Connection Server	TCP	4001	JMS	Java
End user device	Inbound	View Transfer Server	TCP	443	HTTPS	Communication with View Transfer Server
View Security Server	Inbound	View Transfer Server	TCP	443	HTTPS	Communication with View Transfer Server
View Security Server	Inbound	View Transfer Server	TCP	8009	AJP13	AJP-Data Traffic
View Security Server	Inbound	View Transfer Server	TCP	4100	JMSIR	Internal communication
View Security Server	Inbound	View Transfer Server	TCP	4001	JMS	Java
View Connection Server	Inbound	View Transfer Server	TCP	8009	AJP13	AJP-Data traffic
End user device	Both	View Agent	TCP and UDP	4172	PCoIP	PCoIP connection and data

Source IP	Direction	Destination IP	Transport protocol	Port	Application protocol	Description
End user device	Both	View Agent	TCP	32111		USB redirection (if applicable)
View Agent	Outbound	View Connection Server	TCP	4001	JMS	Java
End user device	Both	View Agent	TCP and UDP	4172	PCoIP	PCoIP connection and data
End user device	Inbound	View Agent	TCP	32111		USB redirection (if applicable)
End user device	Inbound	View Connection Server	TCP	443	HTTPS	
End user device	Inbound	View Connection Server	TCP	443	HTTPS	
End user device	Both	View Connection Server	TCP and UDP	4172	PCoIP	PCoIP connection and data

Virtual enclaves

A **virtual enclave** is a defined group of virtual machines, virtual port groups, resources (if using resource pools), and potentially underlying datastores. The notion of a virtual enclave is to provide segmentation within the VDI, separating one group from another.

The following diagram is an illustration showing three separate enclaves:



In the preceding diagram, three classifications of vDesktops exist within the overall virtual infrastructure. These classifications are composed of desktop pools of the same name. They are as follows:

- **Training:** This enclave is used by training rooms to provide vDesktops for training purposes

- **Faculty:** This enclave is used by faculty members at the organization for their primary vDesktop
- **Servers:** This enclave is used by all of the virtual machines running a server-based operating system

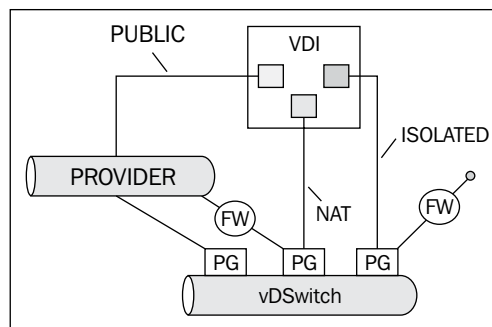
From within the virtual infrastructure, there are ways to isolate the three enclaves as follows:

- VLAN tagging
- Separate vSwitch/vDSwitch uplinks
- Enabling vSwitch/vDSwitch security settings
- Using resource pools to isolate compute consumption
- Using separate datastores to isolate data and I/O
- Using separate clusters

All of the preceding methods are available with VMware vSphere without additional software components.

However, with solutions such as VMware vShield TM, Reflex Systems vTrust TM with vmTagging TM, and other security products, it's possible to provide virtual air gaps from within the virtual infrastructure.

The following diagram shows different segmentation options with VMware virtual networking:



The preceding diagram showcases several virtual networking technologies:

- A virtual distributed switch (vDSwitch)
- Three separate virtual distributed port groups
- Two separate software firewalls (provided through VMware vShield technology or Reflex Systems vTrust technology)

- A provider network connection; this connection has direct access to the Internet (in this example)
- Three separate vDesktop groups

The red enclave contains vDesktops that support within a given organization that require direct access to the Internet for functions that support external clients.

This enclave's vDesktops are connected to the provider port group, which has direct access to the Internet from within the virtual infrastructure. The connection could be filtered further upstream at the physical layer, however.

The blue enclave contains vDesktops used by the majority of the work staff within a given organization.

This enclave's vDesktops are connected to a port group that has access to the provider network; however, instead of direct access to the provider network, it uses **network address translation (NAT)** to mask the actual IP addresses of vDesktops within the blue enclave.

The green enclave contains vDesktops used by the training rooms within a given organization.

This enclave's vDesktops are connected to a port group that does not have access to the provider network (note the fact that the connection terminates as it exits the firewall). Its virtual enclave provides access for the vDesktops and resources within the green enclave to communicate freely. However, there is no mechanism for virtual machines within the green enclave to communicate with resources outside of the enclave. This enclave is described as being isolated.

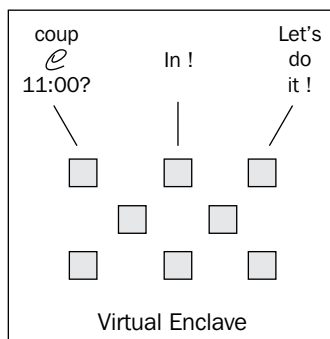
In addition to performing network segmentation, both the VMware vShield and Reflex Systems solutions can provide software-based firewall protection between the various enclaves.

For example, vDesktops in the blue enclave may only be allowed to communicate with vDesktops in the green enclave over port 443 (HTTPS).

The jailbreak scenario

The **jailbreak scenario**, pulled from a real-world solution, involves preventing communications between vDesktops within the same desktop pool. Desktop pools are used to define several key settings of all of its vDesktops; one of these settings is the specific port group (standard or distributed) assigned to one or more vDesktop's virtual NICs. This setting is defined at a desktop pool level; therefore, all vDesktops within a given desktop pool will be on the same port group.

In the jailbreak scenario, the IT staff at a detention center has implemented a VDI to allow inmates to perform various training exercises. While the vDesktops have been locked down to prevent connectivity to the Internet, the fact that all of the vDesktops are on the same port group could pose a threat.



The biggest threat in the prison break scenario is that the various vDesktop users, while segregated from any other network connectivity (including connectivity to the Internet or to the production network of the detention center) still have the ability to send data to one another. The threat is that multiple vDesktop users will leverage the fact that all of their vDesktop virtual machines are on the same port group and send messages to coordinate a revolt on the prison staff at a specific time.

For example, if thirty inmates are logged into a VDI and start trading discrete messages to assault the prison staff at 11:00 a.m., that could pose a huge risk for the prison staff in terms of their own personal safety, the safety of the facilities, and of the nearby community.

While there is no out of the box solution to prevent this type of communication (arguably, the built-in Windows firewall can be of use in this scenario), Reflex Systems does offer the ability to segregate individual virtual machines from one another. In addition, VMware vShield could potentially be used to provide this virtual segmentation. For environments with a high rate of volatility (expansion, contraction, View Composer refreshes, and so on), this solution, regardless of whether VMware, Reflex Systems, or another security solution is used, will require a significant amount of customization, scripting, and integration work.

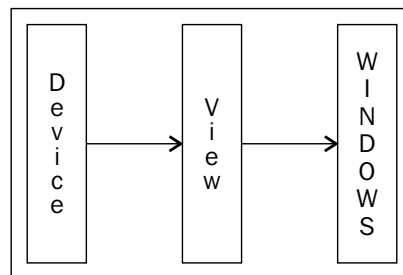
USB redirection and filtering

Throughout a growing number of organizations, USB hard drives are prohibited. This is because of the great risk of data leakage from end users copying sensitive data to a USB hard drive and then misusing the device or using the USB hard drive's data maliciously. However, simply blocking all USB devices may disallow the following perfectly accepted USB devices such as:

- Pointing devices
- Audio headsets
- Transcription playback pedals
- Medical equipment, for example, a patient monitoring device
- Scientific equipment, for example, a metering device
- Photographic equipment, for example, a video recorder
- Audio equipment, for example, USB MIDI interfaces
- Authentication devices, for example, a card reader

Therefore, it's important to not simply block all devices, but instead, build a white list of allowed USB devices. This is known as **USB filtering**.

The following diagram is an illustration showing the three main levels available for USB filtering:



There are three integration points where USB filtering can be applied, they are as follows:

- End device
- View Connection Server
- Windows desktop operating system

In addition, USB Filtering can be applied to:

- An entire ClassID to allow or disallow an entire class of devices (for example, USB mass storage device)
- A **VendorID (VID)** and **ProductID (PID)** to allow or disallow a specific device (for example, Kingston mass storage device)

USB filtering at the end device

One of the benefits of using PCoIP zero clients is the ability to create device profiles and apply them to all zero clients in an environment. In this manner, a single USB filtering device profile could be created and applied to all zero clients in an environment. By locking the devices down with a complex password, the device's profile would only be controlled by, for example, the Teradici Management Console. Therefore, any policy defining USB filtering would not be able to be overridden. By managing USB filtering at the end device, such as with the Teradici Management Console, permissions can be granted on a device ID or device class, allowing great flexibility in management (for example, USB thumb drives are disallowed unless they are made by IronKey TM).

The drawbacks of applying USB device filtering at the device level is that it strongly discourages **bring-your-own-device (BYOD)** programs. These programs encourage end users to use the device that they are most comfortable with; as the device will ultimately be connecting to an organization-owned vDesktop and all work will be performed on the vDesktop, it is of less or no concern whether the end device is in such a scenario.

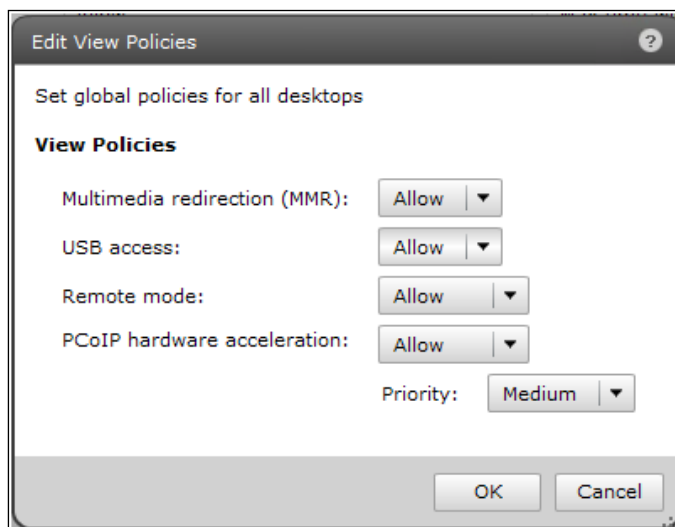
Using device profiles for USB filtering means that a profile must be built for each device, and each device must be managed.

In many organizations, the move to a VDI solution is to get out of the business of managing end devices and instead, enabling the end user workforce to use their preferred method of computing.

USB filtering via View Connection Server

VMware View Connection Server also provides a mechanism for performing USB filtering.

The following screenshot shows the USB access policy setting in the View Admin console:



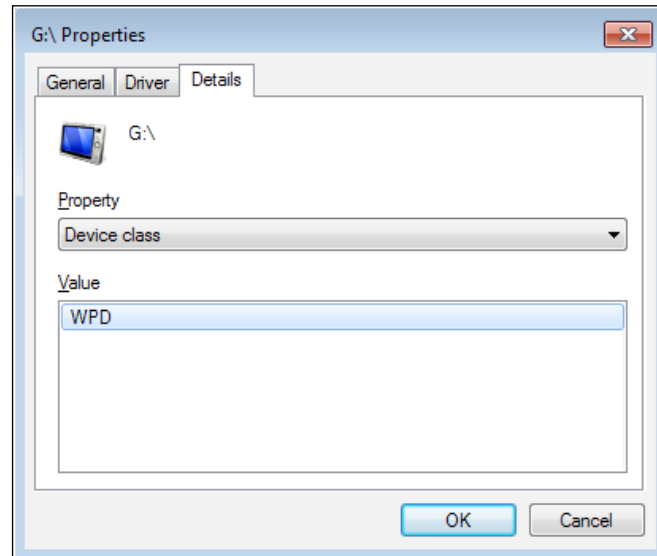
From within the **Policies | Global Policies** section of the View Admin console, **USB access** can be set to the all-encompassing, **Allow** or **Deny**. This does not allow fine-toothed management of only granting access to specific devices. Instead, this allows or disallows USB redirection for all devices.

Another method that is similar in the all or nothing approach, is to not install the USB redirection component of the VMware View Agent from within the vDesktop template or Parent VM. This is not recommended as it is limiting future capabilities within the environment.

USB filtering via the Windows operating system

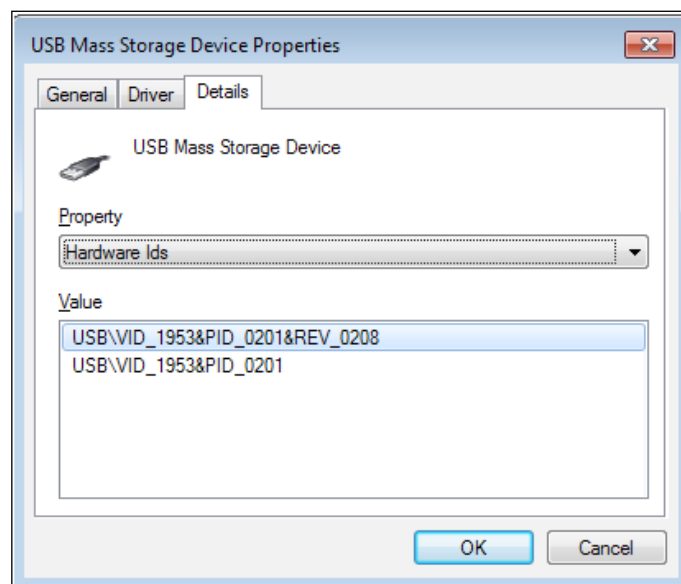
For example, an IronKey encrypted USB drive has a device class of **Windows Portable Device (WPD)**.

The following screenshot shows a USB device from within the guest operating system:



This information can be found by opening the **Properties** tab from within **Device Manager** with the applicable device highlighted.

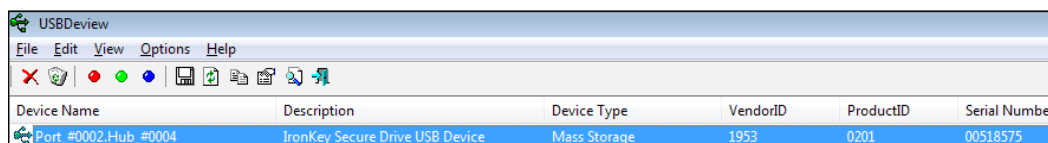
The following screenshot specifically shows a USB device ID:



Under **Hardware Ids** of the given device (for example, IronKey thumb drive), the PID and **firmware revision (REV)** can be found. In the preceding example, **VID** is **1953**, **PID** is **0201** and **REV** is **0208**.

NirSoft makes a free product called USBDeview © that is a handy utility to quickly find the PID, VID, serial number, and other information about a specific USB device as well. It also shows the information in a more user friendly manner.

The following screenshot shows the use of USBDeview to identify the ID of a USB product:

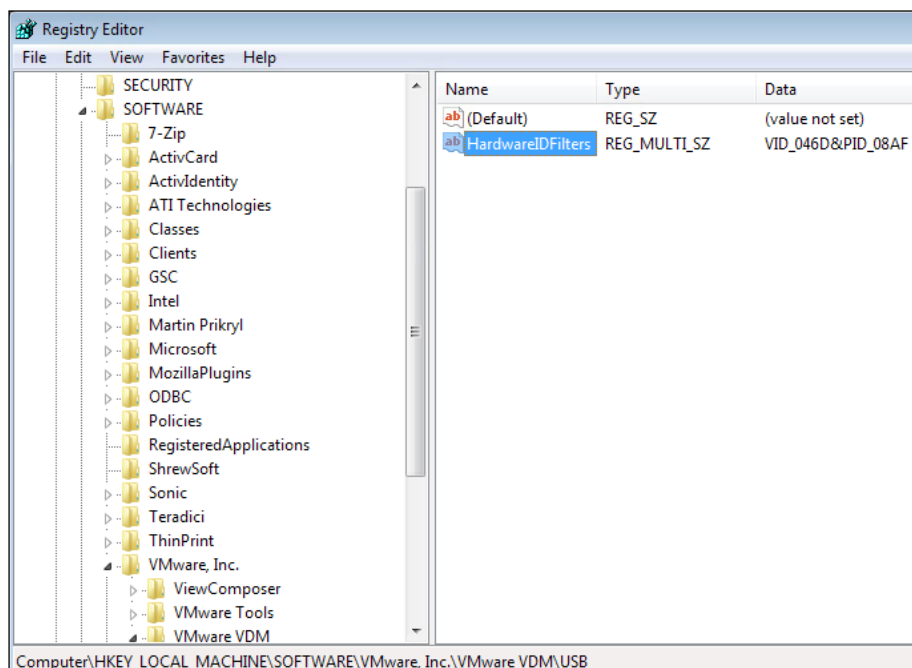


Device Name	Description	Device Type	VendorID	ProductID	Serial Number
Port_#0002.Hub_#0004	IronKey Secure Drive USB Device	Mass Storage	1953	0201	00518575

In the preceding example, the IronKey thumb drive has a **VendorID** of **1953** and a **ProductID** of **0201**.

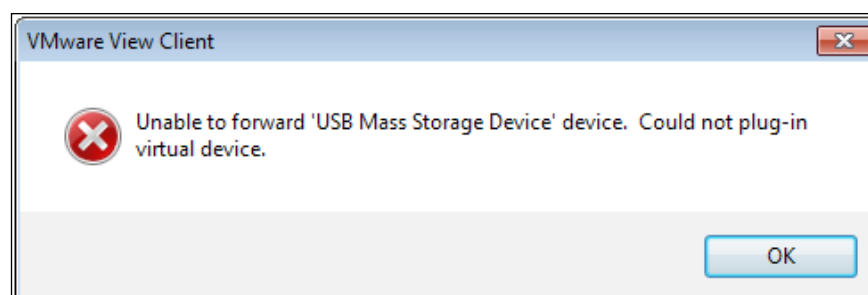
The DeviceClassGUID is also needed to configure USB filtering. It can also be found under the **Properties** tab from within **Device Manager**.

The following screenshot shows the registry key for hardware filters:



For example, to disallow all IronKey encrypted thumb drives (VendorID 1958, ProductID 0201), **VID_1958&PID_0201** would be added to the **HardwareIDFilters** key at the location **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\VMware, Inc.\VMware VDM\USB**.

The registry change takes effect immediately and does not require a reboot. Now, when an end user attempts to connect their IronKey encrypted thumb drive they will receive the error. The following screenshot gives an example of the error message a user may receive when attempting to use a USB device in an environment where that USB device is prohibited:



There are several methods of allowing and disallowing USB devices (especially mass storage devices). However, the techniques outlined in this section fundamentally apply to all devices and should be used as a best practice.

The most secure way of performing USB filtering is to block all devices except those defined on the white list.

Smart card authentication

Smart card authentication is a mechanism by which a plastic card, typically with gold plated contact pads, is used to store certificates used by the end user to authenticate. Smart cards are used throughout many industries, including military, healthcare, education, retail, and the scientific community. The advantages of smart card authentication are as follows:


- It requires the end user to have the authentication mechanism on his person
- It requires the end user to successfully provide the answer to a challenge (PIN)

Smart card authentication is a two-factor authorization mechanism that requires the end user to physically possess a smart card as well as enter in a PIN successfully. The PIN does not authenticate the end user to the domain; instead the PIN authenticates the end user to the certificate on the smart card. The certificate on the smart card is then used to authenticate the end user to the domain.

Smart card authentication is already a standard practice within hospitals, education facilities, the scientific community, and military organizations.

Smart card authentication requires the following things:

- One or more certificates
- Middleware (for example, ActivClient from ActivIdentity)

 Middleware should be installed before the VMware View Agent to avoid any GINA chaining issues; the proper installation order is VMware Tools, then Smart Card Middleware, and then VMware View Agent. It may also be beneficial to set smart card removal behavior to lock the workstation for persistent solutions or log off for non-persistent solutions.

- Smart card
- Smart card reader (for example, SCR331)

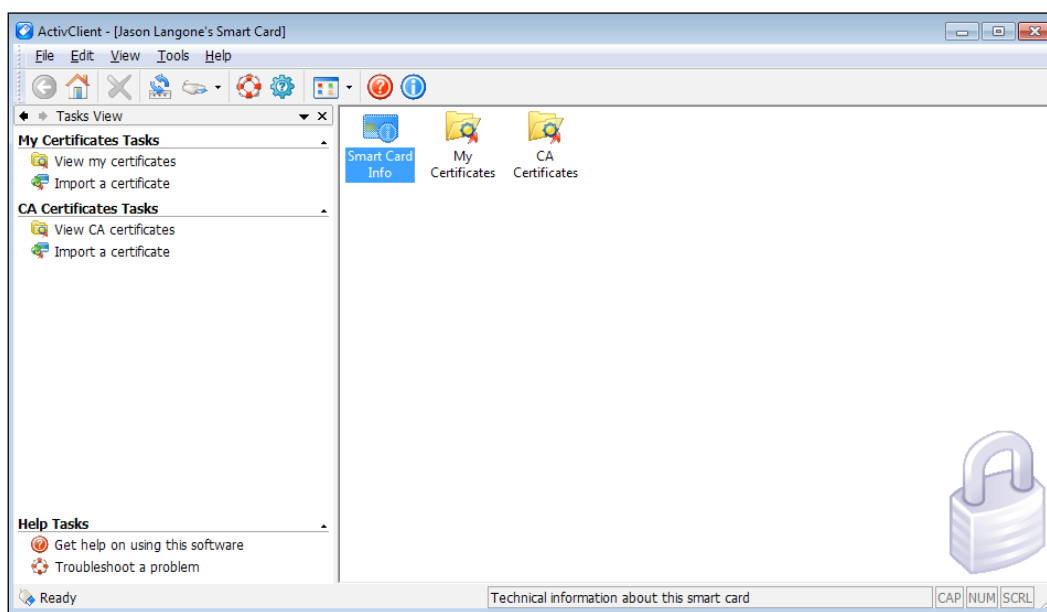
In addition, the following prerequisites must be met:

- The smart card option is installed during the VMware View Connection Server installation process
- The VMware View Connection Server is configured to allow smart card authentication
- The middleware (for example, ActivClient) is functioning properly and is configured with the necessary certificates
- The `locked.properties` file on all of the VMware View Connection and Security Servers in the environment has been configured to use the master keystore holding one or more **Certificate Authority (CA)** certificates for the respective user certificates in use on the smart cards

The smart card configuration should be nearly identical for any organization (with only the certificates being the differentiator). It is important to note that while most smart cards may look the same, there are approximately a dozen or so smart card models on the market.

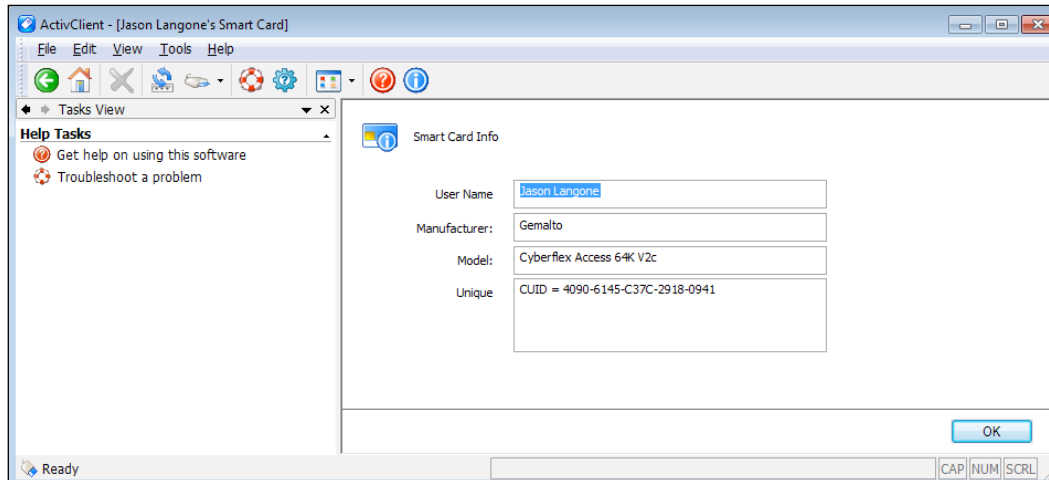
The make and model of the smart card can typically be discovered via the smart card middleware in use, for example, ActivClient.

The following screenshot shows a given smart card from within the ActivClient console:



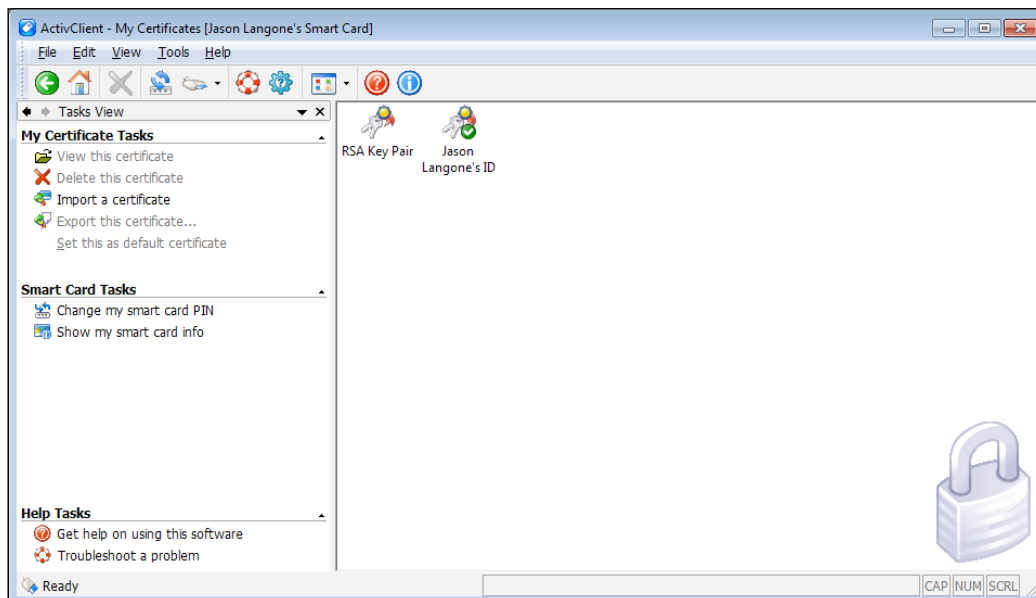
In the preceding screenshot, a card has been inserted into an approved card reader (for example, SCR331). There are three options on the home screen within ActivClient. Clicking on the **My Certificates** folder opens the user certificates stored on the smart card. Clicking on **CA Certificates** opens the CA certificates stored on the smart card and are used to validate the user certificates.

Opening the **Smart Card Info** object brings up the following screenshot that shows smart card information from within ActivClient:



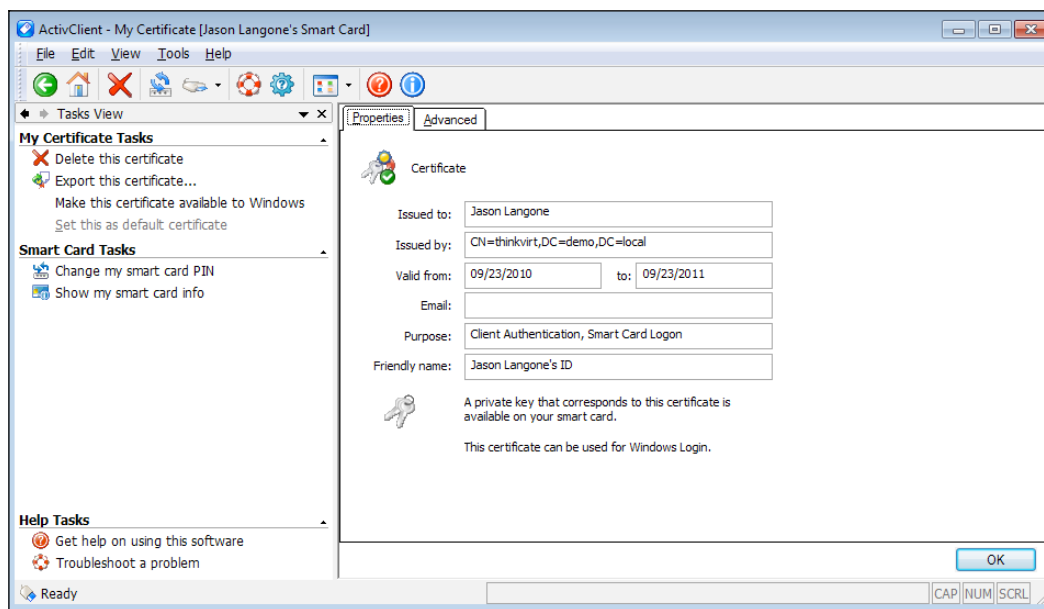
As shown in the preceding screenshot, the **Manufacturer** of the smart card in question is **Gemalto** and the **Model** is **Cyberflex Access 64K V2c**. In addition, the username (typically associated with an Active Directory user account of the same name) is also displayed in the **User Name** field.

The following screenshot shows the certificates on a given smart card:



While only one user certificate is shown in the preceding screenshot, it is possible to have multiple user certificates stored on a smart card. VMware View will filter through the user certificates and prompt the end user to select which certificate to use for authentication. Only valid certificates that have the client authentication and smart card logon role will be displayed.

The following screenshot shows certificate details of a smart card via ActivClient:



The preceding screenshot shows the **My Certificate** screen from within ActivClient. The principal name of the issuing CA for the user certificate (for example, `thinkvirt.demo.local`) is displayed. It is important that the name resolution to the issuing CA be fully functional. Malfunctioning DNS resolution can impact smart card authentication times.

Additional smart card information from VMware can be found in the *Smart Card Certificate Authentication with VMware View 4.5/4.6* whitepaper.

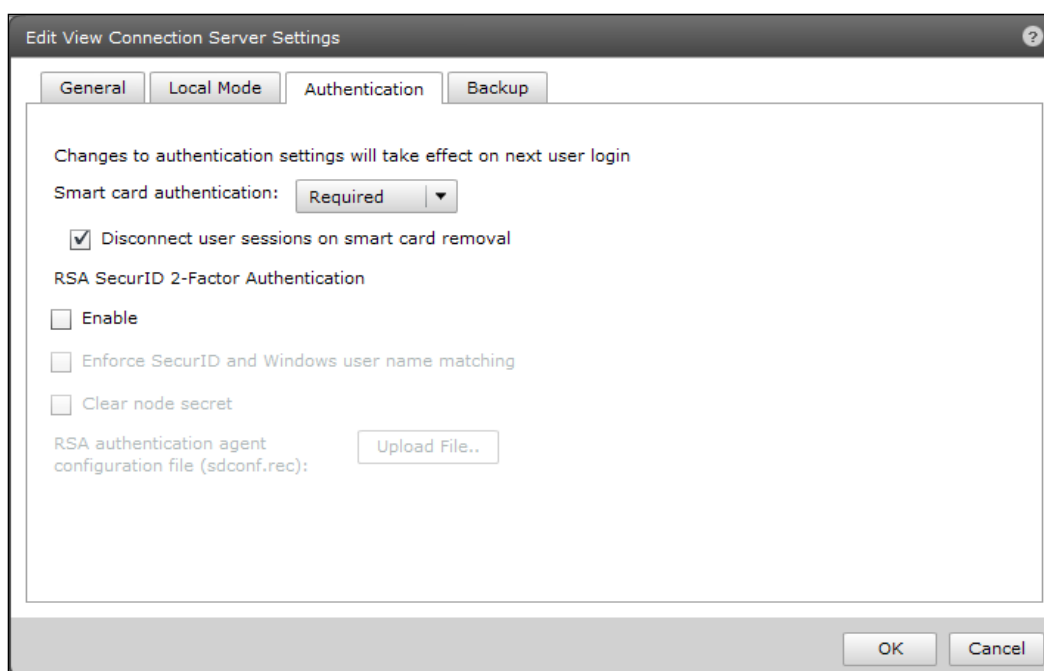
Configuring smart card authentication for VMware View Connection Servers

Smart card authentication is supported by PCoIP zero clients, thin clients, and thick clients. It is important to verify that the exact card reader model, card model, and certificates in use are supported in the VMware View/Teradici PCoIP support matrix. These documents are available at <http://www.vmware.com/> and <http://www.teradici.com/> respectively.

In addition, smart cards are often used to provide a secure mechanism for Single Sign-On, which is the ability to log in to the VMware View environment once (and not for every time a session is connected or reconnected to a vDesktop).

To mandate the use of smart card authentication, go to the **Edit View Connection Server Settings** tab found under **View Configuration | Servers**.

The following screenshot shows advanced smart card configuration options:



Smart card authentication can be set to not **Allowed**, **Optional**, or **Required**. In addition, sessions can be set to disconnect upon smart card removal by selecting the **Disconnect user sessions on smart card removal** checkbox.

In many secured environments, the required option will be configured to enforce that any incoming requests to access a vDesktop in the VDI are authenticated by the use of a user's smart card.

In addition to configuring smart card authentication in the VMware View Admin console, there is a main file of importance during configuration – `locked.properties`.

The `settings.properties` file, also located in the `\sslgateway\conf` subdirectory contains configuration for the certificate used by the VMware View Admin console for HTTPS encryption; the value for which certificate to use is stored in the `keyfile` string. In addition, the `settings.properties` file contains the hashed password necessary to use the certificate; the value for the password is stored in the `keypass` string.

Preparing the environment for smart card authentication

To prepare the environment for smart card authentication, perform the following steps:

1. The first step is to verify a fully functioning DNS and NTP environment as certificate-based authentication is very sensitive to the time drift or difficulty in resolving servers within the **public key infrastructure (PKI)**.
2. Next, the CA certificate must be downloaded. This can be done by opening a browser and pointing it to `http://<CA_SERVER>/certsrv`, where `<CA_SERVER>` is the **fully qualified domain name (FQDN)** or IP address of the CA Server.
3. Select the **Download CA Certificate** link with **DER encoding** (default) selected. Save the certificate to the VMware View Connection Server under `\VMware View\Server\jre\bin`.
4. Launch the command prompt with administrative permission (for example, right-click on the **Command Prompt** icon and select **Run As Administrator**). Navigate to `\VMware View\Server\jre\bin`.
5. Next, type the following command to generate the keystore:

```
keytool -import -alias view4ca -file certnew.cer -keystore trust.  
key
```
6. The `certnew.cer` file is the CA certificate that was downloaded in a previous step. The `trust.key` file is the generated keystore that will be used by the VMware View Connection Server to verify end user certificates stored on their smart card.

7. The keytool utility will then prompt for the CA certificate's password as well as to whether the certificate should or should not be trusted.
8. Once the keystore has been successfully generated, copy the file (for example, `trust.key`) to `\VMware View\Server\sslgateway\conf` subdirectory.
9. Next, create a text file named `locked.properties` within the `\VMware View\Server\sslgateway\conf` subdirectory with Notepad (or a similar tool).
10. Enter the following text:

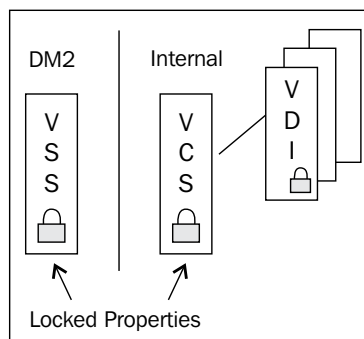
```
trustKeyFile=trust.key
trustStoreType=JKS
UseCertAuth=true
```
11. Restart the VMware View Connection Server service.

Configuring smart card authentication for VMware View Security Servers

For environments that leverage one or more VMware View Security Servers, it is important to configure the View Security Server to also utilize smart card authentication (as well as configuring the appropriate certificates). Otherwise, only internal users or users bypassing the VMware View Security Server, will be able to leverage their smart card for authentication.

The steps are identical to those listed in the *Configuring smart card authentication for VMware View Connection Servers* section.

The following diagram shows an illustration showing the location of the `locked.properties` file:



Therefore, the easiest way to configure a VMware View Security Server (shown as VSS in the preceding diagram) is as follows:

1. Copy the `trust.key` file (or other appropriate keystore file) to the `\VMware View\Server\sslgateway\conf` subdirectory.
2. Copy the `locked.properties` file to the `\VMware View\Server\sslgateway\conf` subdirectory.
3. Restart the VMware View Security Server service.

Notice the simplicity of the preceding steps if a VMware View environment is leveraging the same PKI and same keystore for the CA certificate, which is likely almost all VMware View solutions using smart cards. Therefore, it is quite possible to script the copying of the files from one server to the other, as well as restarting the appropriate service (for example, VMware View Security Server service) when necessary.

Configuring U.S. Department of Defense CAC Authentication

U.S. Department of Defense Common Access Card (DoD CAC) (smart card) authentication is a **Homeland Security Presidential Directive 12 (HSPD-12)** approved mechanism used by U.S. military installations for authentication to IT assets. CACs also serve as a general identification card pursuant to the Geneva Conventions.

Personnel log in to their physical desktop by entering their CAC into a USB smart card reader, laptop smart card reader, a thin or zero client with an integrated card reader, or a keyboard with an integrated card reader. Once the card has been read, the end user is prompted to enter his PIN.

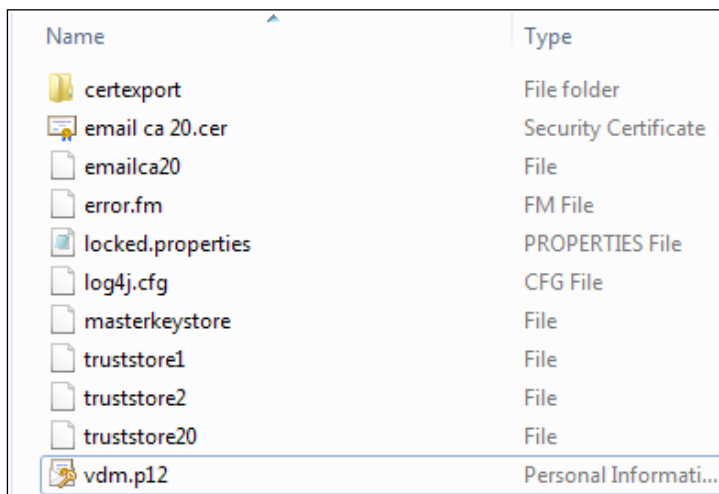
It is entirely possible to use CAC authentication inside the vDesktop without ever configuring it as an acceptable authentication mechanism to get to the vDesktop.

For example, if ActivClient is installed and configured properly within a vDesktop but smart card authentication has not been configured on the View Connection Server environment, then the smart card cannot be used to connect to the vDesktop. However, once connected to the vDesktop, the smart card can be successfully used to authenticate within the VDI (for example, an RDP connection). In normal smart card operation, authentication using a smart card will prompt the end user for his/her PIN. While this scenario is possible, it is far from the preferred solution, as connecting into the VDI is not done via smart card authentication.

Configuring CAC authentication encompasses the techniques used in standard smart card authentication configuration and adds a few minor considerations.

Configuring CAC authentication is explained as follows:

1. On the VMware View Connection Server, navigate to the \VMware View Server\Server\sslgateway\conf subdirectory.
2. Create a subdirectory named certexport. Within \certexport, place all of the .cer files that are applicable. This directory is used to generate a current or future master keystore.
3. The next step of configuring CAC authentication is to generate or obtain a master keystore, which contains all of the U.S. DoD and intermediate CA certificates. The master keystore file should be placed in the \sslgateway\conf subdirectory found within the VMware View installation directory on the View Connection Server. Instructions on how to generate a master keystore are outlined later in this chapter.
4. Copy the truststore files to the \sslgateway\conf subdirectory.



Name	Type
certexport	File folder
email ca 20.cer	Security Certificate
emailca20	File
error.fm	FM File
locked.properties	PROPERTIES File
log4j.cfg	CFG File
masterkeystore	File
truststore1	File
truststore2	File
truststore20	File
vdm.p12	Personal Informati...

The preceding screenshot is a representative screen capture of the \\sslgateway\conf subdirectory of working VMware View Connection Server configured for CAC authentication.

5. Now that all of the certificates are in the proper location, VMware View Connection Server must be configured to use the certificates.

6. Next, open the `locked.properties` file, which can also be found in the `\sslgateway\conf` subdirectory. If the file does not exist, it should be created using Notepad or a similar utility.
7. The contents of the `locked.properties` file should be similar to:

```
trustKeyFile=masterkeystore
trustStoreType=JKS
useCertAuth=true
```

The preceding text assumes that the master keystore file is actually named as `masterkeystore`. The `trustStoreType = JKS` defined that the trust store is a Java keystore generated with the **Java Runtime Environment (JRE)** `keytool.exe` or similar utility. The `useCertAuth = true` enabled the use of the certificate.

Once the settings have been applied, restart the VMware View Connection Server service or the VMware View Security Server service. At this point, it's also important to verify that the View Admin console is still functional, as a malformed `locked.properties` file can prevent the View Admin console from loading properly.

Certificate revocation configuration

A **Certificate Revocation List (CRL)** is used to prevent users whose end user certificate has been revoked (for example, the end user is an employee who has been terminated) from successfully authenticating to the environment. VMware View supports CRLs and **Online Certificate Status Protocol (OCSP)** to check the certificate revocation status of a given certificate. If both OCSP and CRL are configured on a VMware View Connection Server or VMware View Security Server, VMware View will attempt to use OCSP first and then fall back to the use of a CRL if OCSP fails. VMware View will not fall back to OCSP from the use of a CRL if the CRL check fails.

Configure the use of a CRL

The use of a CRL is configured by editing the `locked.properties` file and adding the following lines:

```
enableRevocationchecking=true
allowCertCRLs=true
crlLocation=<URL_OF_CRL>
```

The `enableRevocationchecking` and `allowCertCRLs` strings enables VMware View to perform certificate revocation checking. The `crlLocation` string is used to define the location of the CRL.

An example of a value for `crlLocation` is `http://cert.demo.local/certEnroll/ocsp-ROOT_CA.crl`.

Configure the use of OCSP

The use of a OCSP is configured by editing the `locked.properties` file and adding the following lines:

```
enableRevocationchecking=true
enableOCSP=true
allowCertCRLs=true
ocspSigningCert=<OCSP_Signing_Cert>
ocspURL=<URL_OCSP>
```

The `enableRevocationchecking` and `allowCertCRLs` strings enables VMware View to perform a certificate revocation check. The `enableOCSP` string enables OCSP. The `ocspSigningCert` is used to define the certificate used by the OCSP authority and the `ocspURL` is used to define the location of OCSP responder.

Configure the use of both a CRL and OCSP

To configure the use of both a CRL and OCSP, insert all of the preceding fields and their appropriate values into the `locked.properties` file. Please note that the `allowCertCRLs=true` string only needs to be listed once.

In addition, the following should be added to the `locked.properties` file:

```
ocspCRLFailover=true
```

The `ocspCRLFailover` string allows the VMware View Connection Server or VMware View Security Server to use a CRL if OCSP fails.

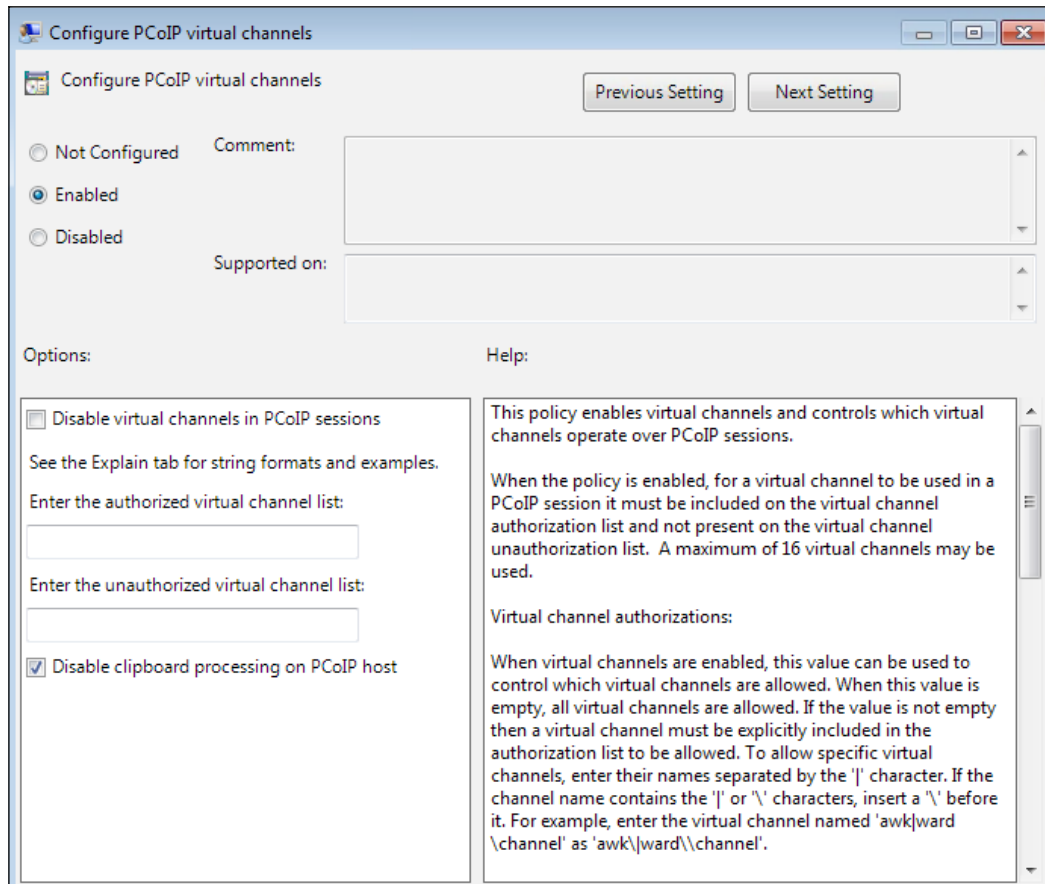
Prohibiting the use of Copy and Paste functions

In some environments, administrators may want to prevent end users from copying and pasting between their vDesktop and their thick or thin client. The proper way to prevent the Copy and Paste functions is via Group Policy of the vDesktops.

This is defined in the `PCoIP.ADM` template available on any View Connection Server in the `\extras` subdirectory. This setting can be found in `Computer Configuration\Admin Templates\PCoIP Session Variables\Not Overridable Admin Settings\Configure PCoIP Virtual Channels`.

Within this setting, there is an allowed and disallowed list. If a virtual channel is listed on both the allowed and disallowed list, it will be disallowed. In View 4.6 and later, the virtual channel responsible for the clipboard (mksvchan) no longer needs to be explicitly mentioned. Instead, the administrator can simply check the disable clipboard processing on the PCoIP host and enable the policy.

The following screenshot shows the settings of the clipboard process:

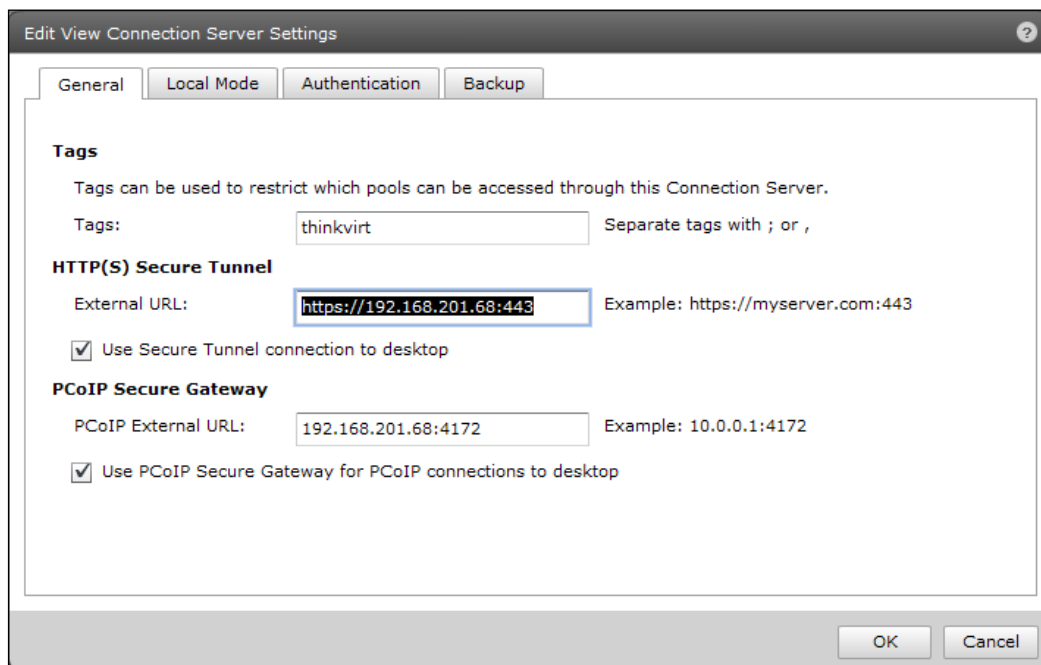


The **Disable clipboard processing on PCoIP host** setting is read at the time of connection or reconnection. Therefore, changing the setting from **Not Configured** to **Enabled**, for example, will go into effect on the next log in and not the existing session.

View Connection Server tags

VMware View Connection Server uses tags to control the access to specific desktop pools in an environment with multiple View Connection Servers. Any given VMware View Connection Server can have no tags, one tag, or many tags. Tags are defined under **View Configuration | Servers | Edit View Connection Server Settings** in the View Admin console.

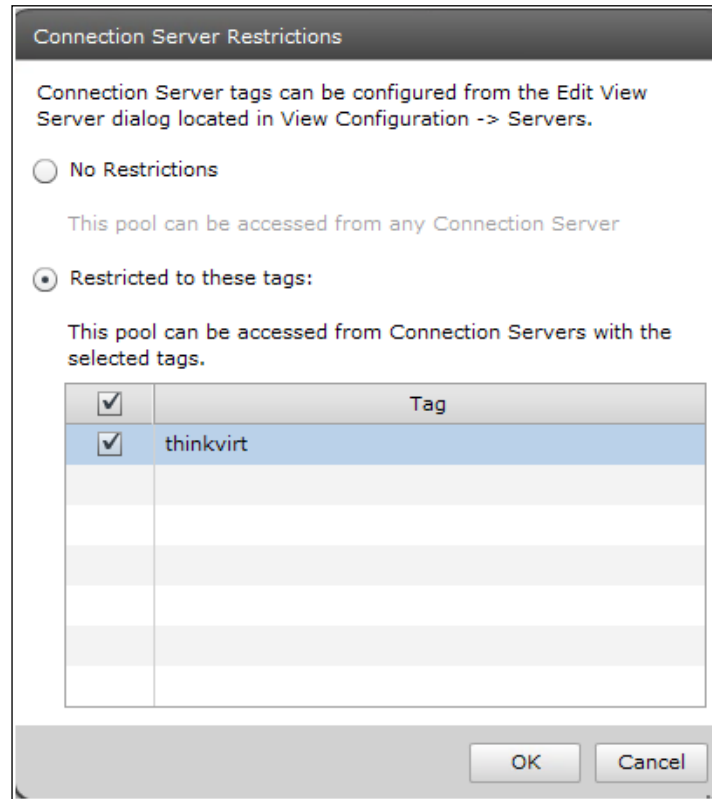
The following screenshot shows the use of a Connection Server tag (**thinkvirt**, in this case):




In the preceding example, a specific VMware View Connection Server has been assigned the **thinkvirt** tag. To assign multiple tags to a VMware View Connection server, separate the tags by either semicolons or commas.

Then, from within the configuration of a desktop pool, on the **Pool Settings** tab select **Browse** for configuration tagging.

The following screenshot shows the use of a restriction tag:



The preceding screenshot would show multiple tags if multiple tags were in use and would allow the administrator to select none, some, or all of the available tags.


 The **Tag** field will only be populated if at least one View Connection Server within the environment has a defined tag.

In the **Connection Server Restrictions** dialog box, there are two options:

- **No restrictions:** This pool can be accessed from any VMware View Connection Server
- **Restricted to these tags:** This pool can be accessed from one or more VMware View Connection Servers with the defined tags

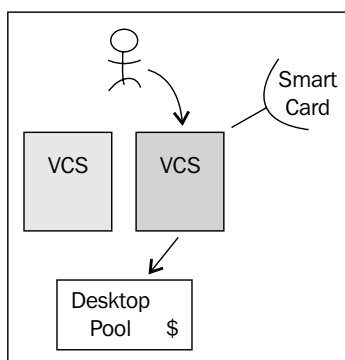
The following is a matrix of tag connection permissions:

Connection Server has defined tag?	Desktop pool is configured to use tags?	Result
No	No	Able to connect
No	One or more	Not able to connect
One or more	No	Able to connect
One or more	One or more	Able to connect only if one or more tags match

One example of when this may be useful is if an organization has two separate inbound VPN environments. VPN_A is used by consultants and visitors. VPN_B is used by employees. If the organization wanted to restrict users of VPN_A to a desktop with limited capabilities and minimal applications installed, one or more separate View Connection Servers could be set up for VPN_A and VPN_B, respectively. The View Connection Servers would be tagged VPNA and VPNB, respectively. Then, the limited desktop pool would only allow connections from VPNA, whereas the fully functional desktop pool would only allow connections from VPNB.

It's important to note that a VMware View solution can leverage more than one vCenter Server. Therefore, not only could tagging limit the pools an inbound user has access to but the backend desktop pools could live on a completely separate virtual infrastructure.

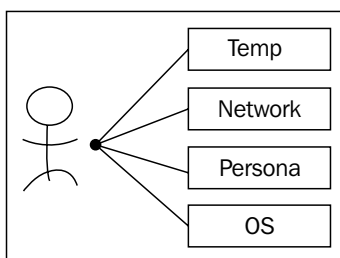
In addition, you could use View Connection Server tagging to identify which users were forced to use two-factor authentication and which were not.



In the previous example, the end user has more than one View Connection Server (VCS) available to him. The Green VCS requires smart card authentication. For example, this VCS could be used by military staff or doctors who have been issued a common access card (CAC), whereby the Blue VCS could be used by civilians, interns, or temporary staff (who are not issued smart cards). Within the VDI there exists a Desktop Pool that has several sensitive financial applications as part of its base image. By using VMware View Connection Server tagging, the financial desktop pool can be configured to only allow incoming connections from the Green VCS, thereby enforcing the use of smart card authentication for incoming users.

Forensics

Forensics, in terms of Information Technology, typically relates to the extraction of legal evidence from computer systems to support legal events. Forensics involves identifying, preserving, recovering, analyzing, and presenting collected data from a computer environment. Forensics is also a required component for many sensitive computing environments looking to leverage VDI solutions.



To understand how forensics is impacted by VDI, it's first important to understand where user-authored or user-manipulated data may reside.

The primary locations for user-authored or user-manipulated data are as follows:

- **Operating system:** For VMware View solutions that do not leverage View Composer or do not leverage redirection of the user's persona, user data will reside within the operating system partition.
- **Persona:** For VMware View solutions that leverage Microsoft roaming profiles, a persona management solution, for example, Liquidware Labs ProfileUnity TM, or VMware View's persistent user data drives, user data will reside within the persona partition. For Microsoft roaming profiles, AppSense, or ProfileUnity, the user data will be stored on a network share. Therefore, ensuring that the network share that stores the user data is backed up according to the organization's policy is imperative as the location where forensics analysis will occur.

For solutions that use persistent user data drives, it is important to preserve these virtual disk files so that they can be attached to other virtual machines if the need to perform forensics arises. When user data resides in the persona layer, virtual machine volatility is of far less concern.

- **Network resources:** For network resources, such as file shares, web-based collaboration resources, the scope for preserving these data points is outside the scope of this book and relies more on understanding various platforms and how they provide auditing and data restoration capabilities.
- **Temporary location:** For solutions that leverage redirecting the user's profile, it's possible that configuration of the solution may miss user data due to misconfiguration and therefore, would be discarded during a vDesktop or desktop pool View Composer.

The biggest challenge for VDI that requires forensics capabilities is the use of non-persistent desktop pools. Persistent desktop pools are automatically assigned once and therefore data, versioning, state, and so on is able to be maintained.

Summary

While a VDI solution is inherently secure in nature, as the end user's data typically resides in a secure data center, it is still important to understand an organization's security posture, policies, and attack vectors and take appropriate measures where necessary. With end users connecting from any location, such as an unsecured Wi-Fi connection at a coffee shop, an Apple iPad over the AT&T 3G network, a corporate LAN, or a home cable ISP, it is important to protect corporate data and intellectual property. The use of smart card authentication—a solution rapidly gaining in popularity—is one strong approach for protecting the authentication entry point. Sound networking policies limiting traffic to defined ports, protocols, sources, and destinations is another key component of a secure VDI.

Finally, understanding basic fundamentals of data forensics to ensure compliance, if necessary, is an important skill to have within the VDI solution team. While potentially the majority of VDI solutions will not require in-depth forensic capabilities, understanding the data points to preserve, monitor, and collect are significant.

The next chapter focuses on the process of migrating from a physical desktop solution to a virtual desktop solution. There are many different approaches that can be taken, and the advantages and disadvantages of each will be covered.

10

Migrating from Physical Desktops to Virtual Desktops

This chapter analyzes the strategies and techniques used to migrate a user population from a physical desktop environment to a virtual desktop solution. While many VDI solutions will be part of net new construction and not involve the migration of users, the majority of VDI solutions to be implemented will involve some component of user migration.

To help ensure success of the overall VDI endeavor, it is important to minimize the perceived impact of the transition to the end users. Part of this impact minimization is understanding how to properly migrate user-specific data, also known as the user's persona.

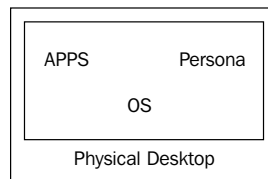
A user's persona consists of the user preferences, application settings, themes, shortcuts, favorites, printers, and other unique configurations. In order to decouple a user's persona from a desktop, the persona must ultimately reside outside of the desktop operating system. Typically, user personas are stored on a classic network file share or a distributed filesystem share. By storing personas on a network share, a consistent end user experience can be delivered no matter which vDesktop resource a user connects to, as the persona is not bound to a specific vDesktop.

There are several solutions on the market that help with the migration of a user's persona, including everything from Microsoft roaming profiles and folder redirection, to AppSense, to Liquidware Labs ProfileUnity.

Migration of the user persona

In order to migrate a user's persona, it must first be decoupled from the desktop operating system. In a completely coupled scenario, the user's persona resides inside the operating environment of the physical desktop.

The following diagram is an illustration that shows the characteristics of a physical desktop:

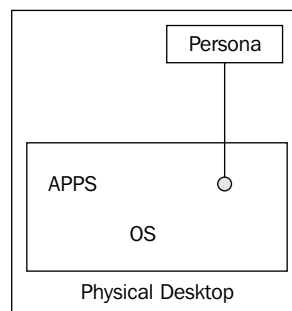


In the preceding diagram, the operating system, applications, and user persona, all reside within the same environment. There is no application virtualization (for example, VMware ThinApp) or persona management solution (for example, ProfileUnity) in place. The first step to successfully migrate the physical desktop in the preceding scenario to a fully functioning vDesktop, is to separate the persona from the operating environment.

Separating the persona from the operating environment

By separating a user's persona from the underlying desktop, it can be freely migrated to another physical desktop, or ideally, to a virtual desktop. This is the same approach used in application virtualization where a given application is packaged via ThinApp, for example, and is now untethered from the underlying operating system.

The following diagram is an illustration that shows the decoupling of the user's persona from the physical desktop:



Three of the easiest options for untethering the persona from the operating environment of the physical desktop are as follows:

- Microsoft roaming profile + folder redirection
- Liquidware Labs ProfileUnity
- AppSense

In this state, a physical desktop still contains installed applications, but customizations and other details that comprise the user's persona reside outside of the guest OS.

Folder redirection

Folder redirection works by redirecting the path of a folder (for example, `\My Documents`) to a new location, typically a network share, unbeknownst to the user. An end user who has his `\My Documents` redirected to a network share will continue to open, save, and manipulate files in his `\My Documents` while the user is opening, saving, and manipulating files on a network share as opposed to the local drive in the background. The advantages of folder redirection are as follows:

- A user's data is accessible from any desktop resource, assuming appropriate network connectivity exists
- Group policy can be leveraged to enforce disk quotas to minimize the space of a user's persona
- A user's data that has been redirected is likely to have a greater chance of recovery from a desktop failure, as production network shares are often backed up more frequently than desktops

With the native Microsoft solution, the `My Documents`, `Application Data`, `Desktop`, and `Start Menu` parent folders can be redirected. The subfolders of the aforementioned parent folders will also be redirected.

`My Documents` is a folder that a user will have read/write access to, and is used as a place to save documents, pictures, media, and other data. `My Documents` is the default save-to location for many Microsoft applications.

The `Application Data` folder is used by applications to save customized user settings relevant to a given application.

The `Desktop` folder is the folder that contains all of the items that reside on a user's desktop.

The `Start Menu` folder contains items found in a desktop's start menu list.

Profiles

In order to understand how roaming profiles work, it is important to understand what makes up a profile in a Windows environment. In Windows, a profile consists of the following:

- **Registry hive:** The registry hive, stored as `NTuser.dat`, stores the contents of `HKEY_CURRENT_USER`
- **Profile folder:** (for example, `C:\Users\User4`)

Within the registry hive and profile folder are the configuration settings for things such as mapped printers, desktop shortcuts, drive mapping, unique processes, and logging.

In Windows, there are several types of profiles as follows:

- **Local profile:** This is the typical type of profile used and is created upon first login of a user to a desktop
- **Roaming profile:** This type of profile makes a local copy of the network-based master copy during login; at logoff, changes are copied back to the network-based master copy
- **Mandatory profile:** This type of profile is used by administrators to specify settings for users; changes made by users are lost during logoff

In many VDI solutions, especially those that are nonpersistent, a roaming profile or other profile management solution will be used. This is because roaming profiles allow any user to access any available vDesktop and still maintain their own unique personalization settings.

How a profile is built: first login

To understand how a Windows profile is built, it is first important to understand the folder directory structure of `C:\Users`.

Under `C:\Users` there are several folders as follows:

- `All Users`: Settings in this folder apply to anyone who logs in to the desktop
- `Default User`: Settings in this folder are used as a template for any new users who log in to the workstation, meaning that they do not already have a profile folder on the desktop
- `Username`: Settings in this folder are unique to the specific user

When a user logs in to a desktop, whether physical or virtual, for the first time, that user has his own unique profile folder created under `C:\Documents and Settings` (for example, Windows XP) or `C:\Users` (for example, Windows 7). The contents of this folder are based off of the contents in `Default User`. In addition, any contents in `All Users` are loaded as part of the profile.

Subsequent logins

Once a user has his/her own unique profile folder on a desktop, they no longer use the `Default User` folder. This means that any settings or shortcuts that have been placed in `Default User` after the user has already created a profile will not be reflected in the user's profile. However, shortcuts placed in the `All Users` folder will be reflected.



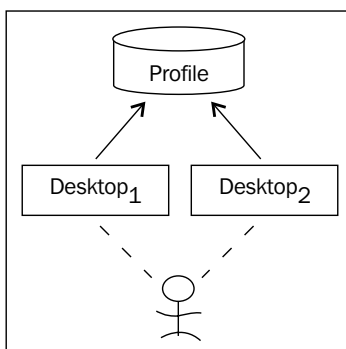
If a user has administrative access to a machine and deletes a shortcut or file that originates from `All Users`, that shortcut or file will be deleted for all users on the machine.

A shortcut or file placed in `All Users` will immediately be displayed to any user logged in to the desktop.

Roaming profiles

Roaming profiles are profiles that are stored in a central repository and accessed on demand at the time of logging in to a desktop operating system.

The following diagram is an illustration that shows a user's ability to log in to either of the desktops and still receive his profile settings:



Using roaming profiles is a technique to store the user's profile folder on a network share, thereby decoupling the user profile from the actual desktop.

In an example scenario without roaming profiles, a user, Dwayne, walks up to a physical desktop, Desktop1. Dwayne works on a document, changes his wallpaper, maps a printer, and then logs out. If Dwayne then walks over to a different physical desktop, Desktop2 and logs in, he would not have any of the work, settings, or mappings he just made on Desktop1. This is because Dwayne's profile physically resides on the local drive of Desktop1.

In the same scenario with roaming profiles enabled, Dwayne's documents, wallpapers, printer mappings, and other settings would be copied to a central network location upon logging off from Desktop1. Therefore, when Dwayne logs in to Desktop2, all of the settings, documents, and mappings would be downloaded from the central network location.

One of the drawbacks of roaming profiles is that it is possible to enter into a scenario where a user's profile is extremely large and the log on and log off tasks are crippled as a profile is synced with the network share. For example, if a user has a 5 GB roaming profile and logs in to a machine for the first time, the entire 5 GB worth of data will be downloaded from the network location before the user is presented with a working desktop. Therefore, it's important to minimize the data that resides in the roaming profile to ensure a positive end user experience.

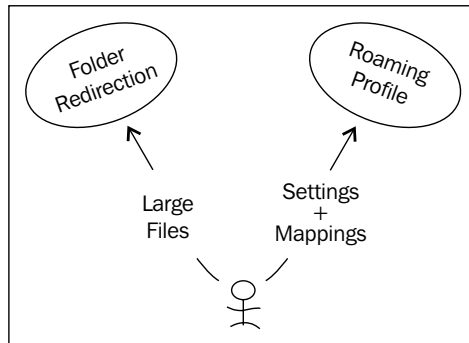


Ensure that "allow offline files" and/or "allow offline caching" is disabled. If using a non-persistent VDI solution and offline caching is allowed, it's possible that a user could log in to a vDesktop and not download the current version of his own profile (because a cached copy exists).

Roaming profiles + folder redirection: increased performance

As the majority of large files in a user's profile are likely to be in locations such as \My Documents, redirecting such a folder to a network location can ensure a user's profile is not overly bloated.

The following diagram is an illustration that shows the segmentation of a user's persona with folder redirection and roaming profiles:



By combining folder redirection and roaming profiles, large files, for example, documents stored typically in `\My Documents` can be redirected to a network location while settings and configuration files can be synced via roaming profiles.

Other third-party solutions: Liquidware Labs ProfileUnity

While there are several profile management solutions on the market, Liquidware Labs ProfileUnity is a cost-competitive solution that maintains settings and configurations in native Windows format versus storing them in a proprietary database.

In addition, ProfileUnity also provides additional benefits such as the ability to:

- Manage user profiles and folder redirection from one console
- Easily configure MAPI profiles for use with Microsoft Exchange Server
- Filter the execution of a script based on rules, machines class, OS, connection type, and so on
- Speed up log on times through the use of compression and profile corruption reduction technologies

In addition, for system administrators not overly comfortable with advanced group policy management, ProfileUnity has a fairly intuitive user interface for management.

Cutting over from physical to virtual

Once the profile has been decoupled from the desktop, a user can log in to a physical desktop on Tuesday and log in to a vDesktop on Wednesday, and maintain all settings. A few things to consider are as follows:

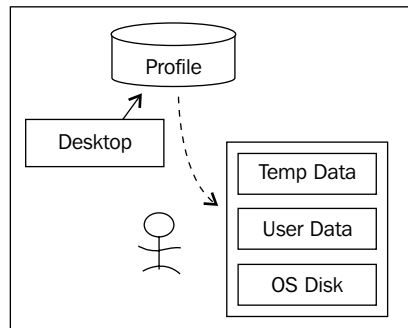
- If upgrading operating systems as part of the migration, ensure all of the settings from the older OS apply to the newer OS.
- If planning to provide the ability to go back and forth between an older and newer OS, special considerations may need to be made to ensure that settings apply. For example, wallpaper file types are different between Windows XP and Windows 7.

Also, considering what type of desktop pool is implemented (persistent or non-persistent) as well as what type of profile management solution is chosen (native Microsoft, Liquidware Labs ProfileUnity, and so on), the first login may take a significant amount of time.

The use of VMware View User Data Disks

VMware View provides the ability to store a user's profile in a User Data Disk. The User Data Disk is tethered to a specific vDesktop in the VDI.

The following diagram is an illustration that shows the user of a User Data Disk for profile management:



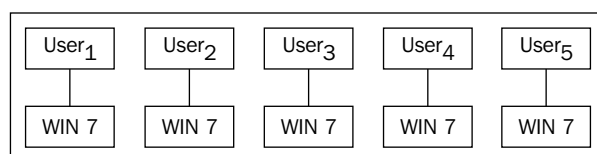
VMware View provides the ability to redirect a user's profile to a persistent **User Data Disk (UDD)**. This disk is separate from the other disks that make up a user's vDesktop; however, a user's UDD can only be attached to one vDesktop at a time. Furthermore, UDDs can only be used with persistent desktop pools.

User profiles can be migrated to a UDD through the use of standard Microsoft tools or third-party solutions. Once a user profile has been completely and successfully migrated to the UDD, it no longer resides on the network share and its contents are only accessible after attaching the UDD (a .vmdk file) to a virtual machine.

Operational considerations with user data

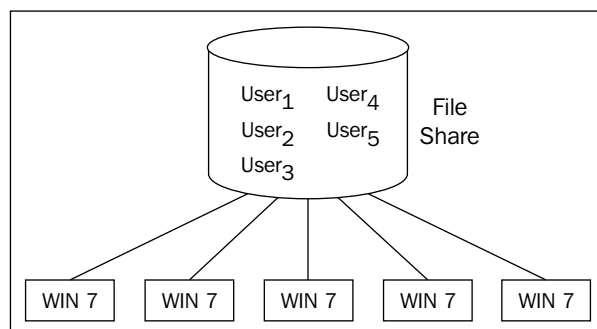
In addition to technical considerations that need to be made in a VMware View solution, there are also operational considerations to be made. One such consideration is the management of user data as it relates to human resource activities. Such activities include the hiring and termination of employees. For example, when an employee is terminated, the user data must be typically archived and stored for a defined period of time.

The following diagram is an illustration that shows the management of User Data Disks in a VMware View environment:



If User Data Disks are part of the solution, the User Data Disk must be detached from the vDesktop and likely moved to a separate data store dedicated to historical data, when an employee is terminated. At some point, if the historical data is to be analyzed, the User Data Disk must first be attached to an existing virtual machine. This can become cumbersome for organizations with seasonal turnover, for example, groups managing election campaigns.

The following diagram is an illustration that shows the management of user profiles residing in a central file share:



By using a central file share, all of the user data across an organization resides in a single place. Access to the file share is often controlled by a user's Active Directory account, therefore disabling a user's account (due to termination, for example) also disables his/her access to the profile directory; however, the profile directory still resides on the file share until an administrator takes action (if necessary).

Summary

For brand new VDI environments (for example, a classroom facility) that don't need to import user data, migrating user persona data is of no concern. However, for many organizations, the migration of user data from a physical desktop to a virtual desktop will be an important part of the implementation process. By decoupling the user's data from the desktop operating system, the user's settings can be maintained while the actual desktop is transitioned from physical to virtual. In addition, it is important to have an understanding of how the user profile solution may impact business processes, especially those related to human resources.

The next chapter will focus on backing up the VDI as well as recovering during an outage. While redundant design is covered in this book, there are times when an unforeseen or unscheduled outage could cause a potential issue in the VDI. Therefore, it is important to understand how to protect and recover from such outages.

11

Backing Up the VMware View Infrastructure

While a single point of failure should not exist in the VMware View environment, it is still important to ensure regular backups are taken for quick recovery in times of failure. Also, if a setting becomes corrupted or is changed, a backup could be used to restore to a previous point in time. The backup of the VMware View environment should be performed on a regular basis in line with an organization's existing backup methodology. A VMware View environment contains both files and database.

The main backup points of a VMware View environment are as follows:

- VMware View ADAM database
- VMware View Composer database
- VMware vCenter database
- VMware View Connection Server settings

With a backup of all of the preceding components, the VMware View Server infrastructure can be recovered during a time of failure.

To maximize the chance of success in a recovery environment, it is advised to take backups of the View ADAM, View Composer, and vCenter database at the same time to avoid discrepancies. Backups can be scheduled and automated or can be manually executed; ideally, scheduled backups will be used to ensure that they are performed and completed regularly.

Proper design dictates that there should always be two or more View Connection Servers. As all View Connection Servers in the same replica pool contain the same configuration data, it is only necessary to back up one View Connection Server. This backup is typically configured for the first View Connection Server installed in standard mode in an environment.

Backing up the VMware View Connection Server environment

View Connection Server backups can be configured from the VMware View Admin console. These backups dump the configuration files and database information to a location on the VMware View Connection Server, and must then be backed up through normal mechanisms, such as a backup agent and scheduled job. The workflow for a VMware View Connection Server backup is as follows:

1. Schedule VMware View Backup runs and exports to C:\View_Backup\.
2. Third-party backup solution runs on the View Connection Server and backs up the System State, Program Files, and C:\View_Backup\ folders.

From within the VMware View Admin console, there are three primary options that must be configured to back up VMware View Connection Server settings:

- **Automatic backup frequency:** This is the frequency at which backups are automatically taken
 - **Recommendation – every day:** As most server backups are performed daily, if the automatic View Connection Server backup is taken before the full backup of the Windows server, it will be included in the nightly backup; adjust as necessary
- **Max number of backups:** This is the maximum number of backups that can be stored on the View Connection Server; once the maximum number has been reached, backups will be rotated out based on age, with the oldest backup being replaced by the newest backup
 - **Recommendation – 30 days:** This will ensure that approximately one month of backups is retained on the server; adjust as necessary
- **Folder location:** This is the location on the View Connection Server, where the backups will be stored
 - Ensure that the third-party backup solution is backing up this location

Once a backup has been either automatically run or manually executed, there will be two types of files saved in the backup location as follows:

- **LDF files:** These are the LDIF exports from the VMware View Connection Server ADAM database and store the configuration settings of the VMware View environment
- **SVI files:** These are the backups of the VMware View Composer database

The backup process of the View Connection Server is fairly straightforward. While the process is easy, it should not be overlooked.

Security server considerations

Surprisingly, there is no option to backup the VMware View Security Server via the VMware View Admin console. For View Connection Servers, backup is configured by selecting the server, selecting **Edit**, and then **Backup**. Highlighting the View Security Server provides no such functionality.

Instead, the security server should be backed up via normal third-party mechanisms. Of primary concern, is the installation directory, which is `C:\Program Files\VMware\VMware View\Server` by default.

In the `...\sslgateway\conf` directory is the `.config` file, which includes the following settings:

- `pcoipClientIPAddress`: This is the public address used by the security server
- `pcoipClientUDPPort`: This is the port used for UDP traffic (default: 4172)

In addition, the `settings` file is located in this directory, which includes settings such as:

- `maxConnections`: This is the maximum number of concurrent connections the View Security Server can have at one time (default: 2000)
- `serverID`: This is the hostname used by the security server

In addition, custom certificates and log files are stored within the installation directory of the VMware View Security Server. Therefore, it is important to back up the data regularly if the log file data is to be maintained (and is not being ingested into a larger enterprise log file solution).

Transfer server and ThinApp repository considerations

There are two components to backing up the VMware View Transfer Servers in an environment. The first component is that of the transfer server's VMware installation directory and the server's registry. The second component, which is significantly more important, is the actual repository storing the published vDesktops.

If the View Transfer Server itself fails but the repository is online, the checkout and check-in tasks will be unavailable, but the main data, the published vDesktops, will still be preserved. Once the View Transfer Server is rebuilt and restored, it should function as normal. In most production scenarios, there should be two transfer servers that are likely pointed to the same redundant file share.

If the View Transfer Server is online but the repository fails, any transfer-related activity, such as checkout, update, or check-in, will fail.

The ThinApp repository is similar in nature to the transfer server repository in that it should reside on a redundant file share that is backed up regularly. If the ThinApp packages are configured to preserve each user's sandbox, the ThinApp repository should likely be backed up nightly.

Restoring the VMware View environment

The steps to perform a View Connection Server restore can primarily be found in the VMware KB article, *Performing an end-to-end backup and restore for View Manager 3.x or 4.x*.

Backing up the gold templates

For environments using full cloning as the provisioning technique for the vDesktops, the gold template should be backed up regularly. The gold template is the mastered vDesktop that all other vDesktops are cloned from. The VMware KB article — *Backing up and restoring virtual machine templates using VMware APIs* covers the steps to both back up and restore a template. In short, most backup solutions will require that the gold template is converted from a template to a regular virtual machine at which time it can then be backed up.

Backing up the Parent VM

Backing up the Parent VM can be tricky as it is a virtual machine, often with many different point-in-time snapshots. The most common technique is to collapse the virtual machine snapshot tree at a given point-in-time snapshot, and then back up or copy the newly created virtual machine to a second datastore. By storing the parent VM on a redundant storage solution, it is quite unlikely that the Parent VM will be lost. What's more likely is that a point-in-time snapshot may be created of the Parent VM while it's in a non-functional or less-than-ideal state.

Summary

As expected, it is important to back up the fundamental components of a VMware View solution. While a resilient design should mitigate most types of failure, there are still occasions when a backup may be needed to bring an environment back up to an operational level. Some of the additional tools that could prove useful to a VDI architect designing a VMware View solution are discussed in *Appendix, Additional Tools*.

12

VMware View 5.1

At the launch of this book, VMware View 5.0 was being superseded by the VMware View 5.1 release, and despite being a .1 release, the VMware View team added a significant number of features that improve the VMware View performance, scalability, and user experience.

This chapter splits the new features into five main areas as follows:

- Platform
- User experience and client
- Management and administration
- Persona management
- Security

Platform features

The platform feature enhancements in VMware View 5.1 are focused on making the storage requirements less dramatic. As a significant portion of this book has been dedicated to streamlining and optimizing the storage requirements, the improvements in VMware View 5.1 are quite welcome.

Content-Based Read Cache (also known as View Storage Accelerator)

The **Content-Based Read Cache (CBRC)** feature is native to VMware vSphere 5 and is managed by VMware View. CBRC helps to address some of the typical VDI performance bottlenecks, as well as help to decrease the overall storage cost for VDI.

CBRC is a RAM-based caching solution on a given ESXi host that helps to reduce the number of read I/Os issued to the storage subsystem. By reducing the number of read I/Os issued to the storage subsystem, improved scalability of the storage subsystem and overall performance can be realized. CBRC is completely transparent to the guest OS (vDesktop).

VMware announced that during their tests with CBRC, there was an approximate reduction of boot storm as follows:

- 80 percent of peak IOPS
- 45 percent of average IOPS
- 65 percent of peak throughput
- 25 percent of average throughput

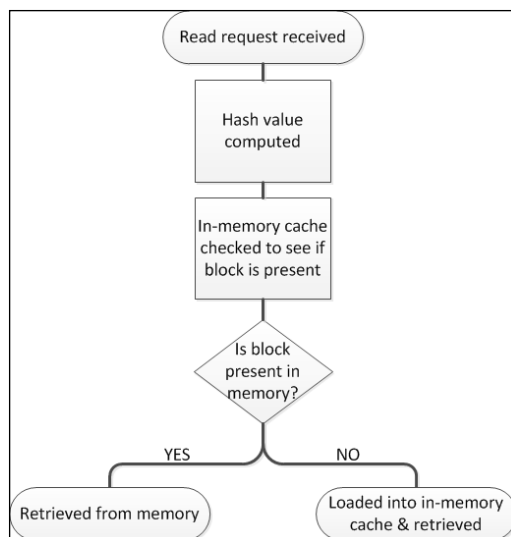
These are significant savings of the storage subsystem and should be carefully considered during a VMware View design. It should be noted that CBRC is a feature for read I/Os only.

There are two components of the cache as follows:

- **In-memory cache:** This is configured by the administrator and has a fixed maximum size of 2 GB and default memory reservation of 400 MB
- **Dynamic cache:** It loads blocks on demand and manages the cache based on access patterns of the various blocks on the VMDK

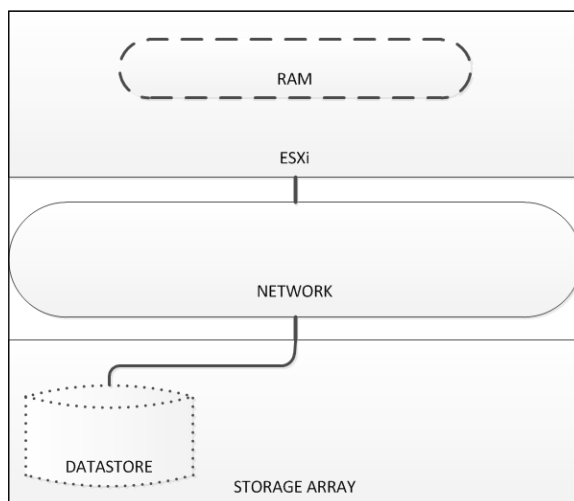
A digest/metadata table that is maintained on disk for each VMDK disk on the host. The metadata holds information about the various blocks on the VMDK. It can be imagined as a hash table with each hash entry pointing to a particular block.

Putting the preceding two components together, if there is a read request to a particular block on the VMDK, a hash value is computed, and the in-memory cache is checked to see if the block is present. If it is not present, the hash table is accessed and the appropriate block is loaded into the in-memory cache. If the block is already in the in-memory cache, it is returned back to the user.



The additional memory taken up by CBRC itself is treated as a **regression** as far as memory consumption goes. As memory requirements are not as high for CBRC for steady state workload, vSphere characterizes and reduces memory consumption.

CBRC will benefit VDI environments without intelligent arrays and cache management. However, for arrays with read or read/write cache management, CBRC will also help to reduce I/O latency in the storage fabric. As read I/Os are served from in-host RAM, there is no requirement to go out to the network to retrieve data blocks. Additionally, data blocks are retrieved to the guest in terms of microseconds, instead of milliseconds.



The preceding diagram highlights the direct access the ESXi host has to its memory, where the CBRC resides. If data must be retrieved from a typical storage array, the request must traverse any paths to have the request fulfilled. The I/O performance improvement delivered by CBRC is clearly noticed by end users while using their desktops. However, it should be duly noted that the majority of I/Os during a steady-state workload are write I/Os, not read I/Os.

View Storage Accelerator allows host caching of OS Disks and user-persistent disks on linked clone desktops; and administrators also have the ability to specify how often the cache metadata should be regenerated.

CBRC storage sizing

When View Storage Accelerator is enabled (OS Disk, or OS and persistent disk), a per-VMDK digest file is created to store hash information about the VMDK blocks.

The estimated size of each digest file is roughly:

- 5 MB per 1 GB of the VMDK size when hash collision detection is Off (default value)
- 12 MB per 1 GB of the VMDK size when hash collision detection is On

[esx01-datastore-SSD-01] replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3			
Name	Size	Provisioned Size	Type
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3.vmdk	9,136,128.00 KB	25,165,820.00 KB	Virtual Disk
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3.nvram	8.48 KB		Non-volatile me...
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3.vmx	3.24 KB		Virtual Machine
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3.vmx	0.29 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3.vmsd	0.50 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-digest-flat.vmdk	124,992.00 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-digest.vmdk	0.52 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-Snapshot1.vmsn	28.19 KB		Snapshot file
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-000001-digest-delta.vmdk	16,388.00 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-000001-digest.vmdk	0.42 KB		File
replica-3bbc1760-b007-4bd5-b8b2-a998098e5fc3-000001.vmdk	1,024.00 KB	25,165,820.00 KB	Virtual Disk

The preceding screenshot shows the existence of the CBRC digest files in a given datastore.

Name	Target	Status
Configure virtual disk digest	vcenter.lab.local	In Progress

Within VMware vCenter's task pane, the creation of a virtual disk digest will also be displayed.

The digest file creation for a large replica disk can take a reasonable amount of time and IOPS, therefore it is recommended not to run the operation, create new desktop pools, or recompose existing pools during production hours.

As an example, a 25 GB Windows VM replica will consume about 125 MB of storage space for the digest file. For snapshots (deltas) or persistent disks, a snapshot is created for the digest file as well. If a VMDK is cloned, the digest file is copied.

Due to the fact that Windows-based desktops will have a significant percentage of identical blocks between them, it is safe to assume that performance gains can be realized when using CBRC with full desktop clones. At the time of writing, however, full desktop clones were not supported by CBRC.

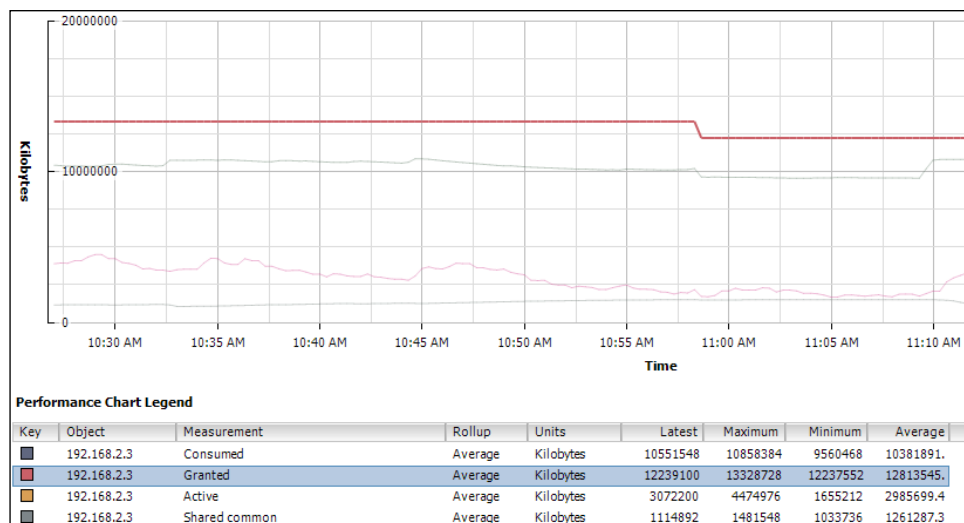
For 5 GB persistent disks, the digest file will be approximately 24 MB.

Host memory sizing

CBRC uses a RAM cache to manage the cached disk blocks. The per-VMDK digest file is also loaded in the memory.

CBRC should not be enabled under memory-overcommit environments. If a host is memory overcommitted and CBRC is enabled, the memory pressure is increased as CBRC also uses memory for the cache. In such cases, the host could experience increased swapping and the overall host performance could be impacted. In this scenario, enabling CBRC could actually make the performance worse.

The following screenshot is a graph demonstrating the moment when CBRC with 512 MB cache is enabled on the host:



As CBRC consumes host memory as part of its architecture, host disk swap and performance degradation can be mitigated by reducing VM density on the hosts themselves. A more favorable approach is to size the hosts appropriately to include the additional RAM required to support the CBRC functionality.

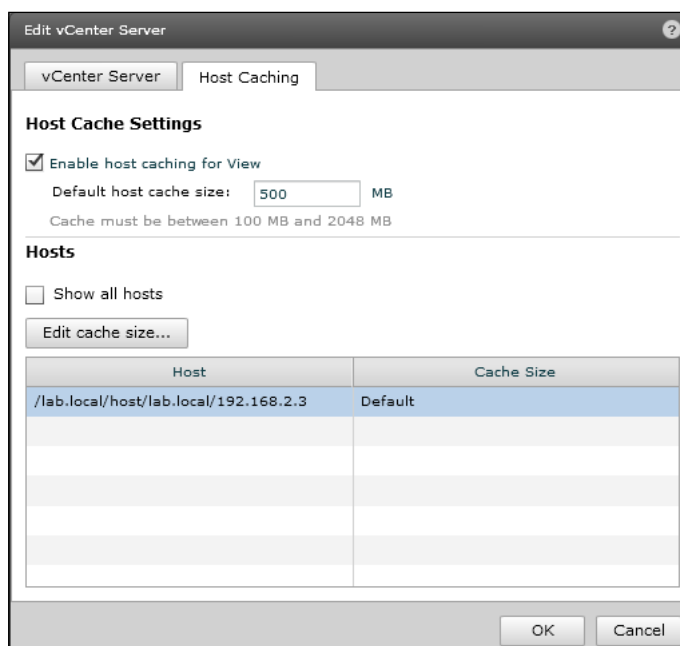
For each VMDK digest created, approximately 24 MB of RAM is consumed in addition to the defined CBRC cache. As an example, if only one system disk is being hashed and the host cache is 500 MB, then $500 \text{ MB} + 24 \text{ MB} =$ total of 524 MB memory will be used.

It is important to remember that it is possible to create digests for system and persistent disks as well.

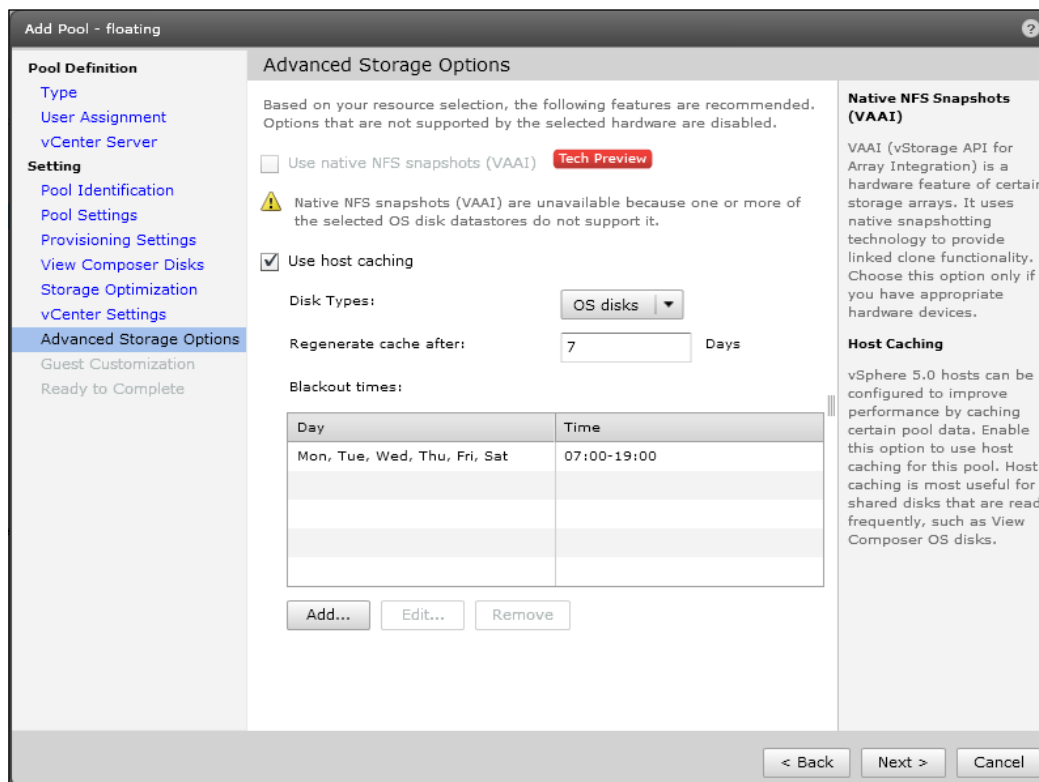
In another example, if 64 VMs were in use, there would be 64 persistent disks, plus 1 replica disk. In this case, we would have 65 VMDK to be hashed. Assuming that host cache is using 2 GB RAM (maximum size), then $2048 \text{ MB} + (65 * 24 \text{ MB}) =$ total of 3.5 GB memory will be used.

Managing CBRC

In VMware View, CBRC is under the **Host Caching** tab of **vCenter Server** configuration. It is possible to enable and define the total amount of RAM cache assigned for the host. Each host may have a different cache size, although it is recommended to maintain consistency across the vSphere cluster.



During the desktop pool creation process, administrators may define that the pool should use CBRC, the types of disks to have the digest file created for, and how often the digest file should be regenerated.



The following options are exposed through `/config/CBRCFilter/intOpts` and is visible through the VMware vSphere Client Advanced Configuration. VMware View has built-in capabilities to manage the following options, and it's recommended not to manually modify any of these items:

- `/config/CBRC/intOpts/DCCacheMemReserved`: Memory consumed by CBRC data cache (in MB).
- `/config/CBRC/intOpts/DCCacheSize`: Size of CBRC data cache (in MB). This cannot be changed if **CBRC.Enable** is set to **1**.
- `/config/CBRC/intOpts/DigestJournalBootInterval`: Interval (in minutes) for which Digest Journal is temporarily disabled to avoid interfering with the boot process.
- `/config/CBRC/intOpts/Enable`: Enable Content-Based Read Cache.

It is important to note that View Storage Accelerator is not supported under certain conditions, including:

- With View Composer APIs for Array Integration, which is a Tech Preview feature of View 5.1
- For use with desktops with the Local Mode feature turned on
- When VMware View replica tiering is enabled.

View Composer Array Integration

View Composer Array Integration (VCAI) is a Tech Preview feature in VMware View 5.1, which allows administrators to take advantage of the storage-native snapshot feature within the normal administrative workflow of VMware View and View Composer.

VCAI integrates with **Network-attached storage (NAS)** partner's native cloning capabilities using **vSphere vStorage APIs for Array Integration (VAAI)**. VCAI speeds up provisioning of linked clone virtual desktops in automated pools, helping to offload CPU consumption and network bandwidth.

Add Pool - VCAITEST

Pool Definition

- Type
- User Assignment
- vCenter Server

Setting

- Pool Identification
- Pool Settings
- Provisioning Settings
- View Composer Disks
- Storage Optimization
- vCenter Settings
- Advanced Storage Options**
- Guest Customization
- Ready to Complete

Advanced Storage Options

Based on your resource selection, the following features are recommended. Options that are not supported by the selected hardware are disabled.

☒ Use native NFS snapshots (VAAI) **Tech Preview**

☐ Use host caching

⚠ Host caching is disabled in vCenter settings.

Disk Types: OS disks

Regenerate cache after: 7 Days

Blackout times:

Day	Time

Add... Edit... Remove

Native NFS Snapshots (VAAI)

VAAI (vStorage API for Array Integration) is a hardware feature of certain storage arrays. It uses native snapshotting technology to provide linked clone functionality. Choose this option only if you have appropriate hardware devices.

Host Caching

vSphere 5.0 hosts can be configured to improve performance by caching certain pool data. Enable this option to use host caching for this pool. Host caching is most useful for shared disks that are read frequently, such as View Composer OS disks.

< Back Next > Cancel

It is recommended to read *VMware End-User Computing Blog* at <http://blogs.vmware.com/euc/2012/05/view-composer-array-integration-tech-preview.html>.

Support 32 (up from 8) hosts in a cluster on NAS

Until VMware View 5.1, any vSphere cluster supporting VMware View deployments with linked clones would only allow for clusters with a maximum of 8 hosts.

The reason behind the limitation is a VMFS limit on the number of hosts that can concurrently execute I/O operations against a single file; the replica disk.

This was never much of an issue when talking about NFS exports; however, the limitation was hardcoded into View Composer—the tool responsible for creating the linked clones.

With VMware View 5.1, this limitation has been removed and View Composer will support a cluster with 32 hosts if the underlying storage filesystem and protocol is NFS. This change completely modifies the architecture of many VMware View deployments with NFS-based clusters. As this was a late addition to the book, its increased in-host support (from 8 to 32) for NFS will be addressed in a future blog post or addendum.

Standalone View Composer Server

VMware View Composer is the software responsible for creating linked clones and may now be installed in a server other than vCenter Server. This move is aiming towards a highly scalable VMware View architecture.

Edit vCenter Server

View Composer Settings

☐ Do not use View Composer

☒ View Composer co-installed with vCenter Server
Choose this if View Composer is installed on the same server as vCenter

Port: 18443

☐ Standalone View Composer Server
Choose this if View Composer is installed on a separate server from vCenter

Server address:

User name:

Password:

Port: 18443

Domains

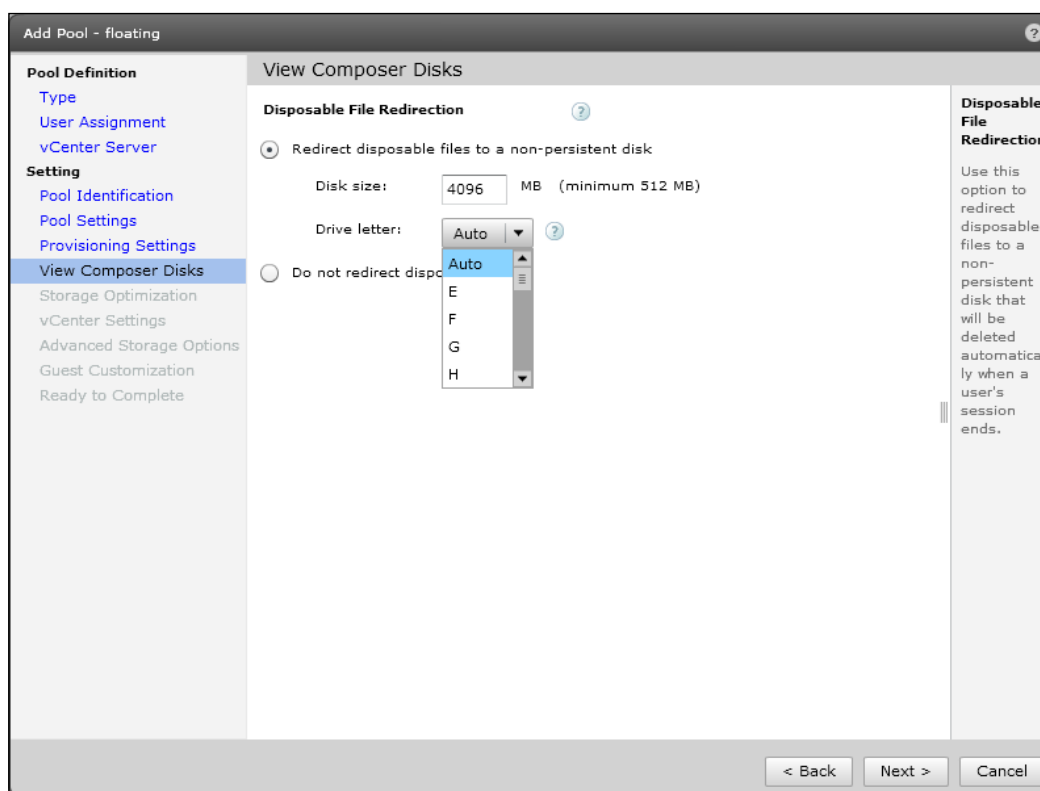
Verify Server Information

OK Cancel

The preceding screenshot shows the configuration tab where a standalone View Composer Server can be configured. This is ideal for large environments where the resilience and performance of both the VMware vCenter Server(s) and VMware View Composer Server(s) must be protected.

Customizable disposable disk drive letter

VMware View 5.1 has added the ability to specify the drive letter for disposable disks. In the past, the disposable disk would utilize the first available drive letter in the desktop. VMware View can still autoselect the drive letter by leaving the **Drive letter** option set to **Auto** mode, as shown in the following screenshot:



The preceding screenshot shows the configuration tab for the disposable disk drive letter.

User experience and client features

VMware View 5.1 has a large number of improvements that directly affect the user experience. Both VMware View and Teradici PCoIP continue to evolve with every release. Comparing the end user experience of VMware View 4.x and VMware View 5.1 shows a very significant improvement.

VMware View 5.1 delivers enhancement in several user experience and client areas, including Local Mode and USB redirection.

The enhancements for VMware View Local Mode are as follows:

- Multi-monitor support
- Disk I/O performance improvements and reduced deduplication I/O cost
- NAT support for DNS over TCP
- Local Mode disk consistency validations
- Virtual Hardware Version 8 support
- Improved NAT, DNS resolution performance, link state propagation
- One click to send the *Ctrl + Alt + Delete* keystrokes
- Automation Support for Point-of-Sale operations
- Data integrity and security hardening

VMware also reworked the USB stack for Windows clients with VMware View 5.1. The enhancements for USB are as follows:

- Broader device support
- New filtering mechanism for better management of devices on Client, configurable via Group Policies
- Multi-platform support for USB View Client
- New filtering mechanism for better management of devices on Agent, allowing blocking of unwanted devices and blocking of devices that are forwarded by other means (for example, keyboards/smartcards), configurable via Group Policies
- The device driver for a device no longer needs to be installed on the client machine

There are also a small number of enhancements for PCoIP. These enhancements include reduction in CPU utilization to decode PCoIP frames on the client side, ultimately improving the protocol performance.

Management and administration

While VMware View has proven to be relatively easy to manage and administer, there have been definite areas for improvement, including basic UI features such as right-click. In VMware View 5.1, there are several improvements to the UI and management of VMware View.

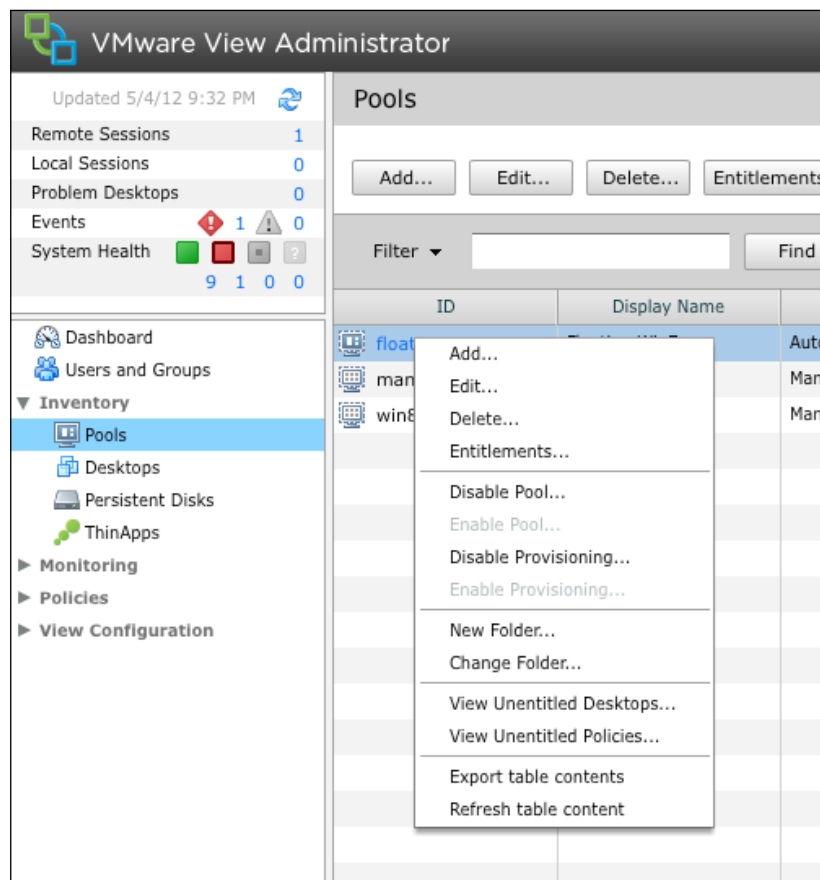
UI enhancements and localization

The user interface in VMware View 5.1 has a new look and feel; it's sleeker and faster. VMware View has also been localized to five different foreign languages (French, German, Japanese, Korean, and Simplified Chinese).

The screenshot displays the VMware View Administrator interface. The left sidebar contains a navigation tree with categories like Inventory, Monitoring, Policies, and View Configuration. The main area is divided into several panels. The 'System Health' panel shows a tree of components including Connection Servers, Event database, View Composer Servers, vSphere components (Datastores, ESX hosts, vCenter Servers), and Other components (Domains). The 'Desktop Status' panel shows a table of desktops with columns for status and count. The 'Datastores' panel at the bottom shows a table of datastores with columns for Datastore, vCenter Server, Path, Capacity (GB), and Free Space (GB).

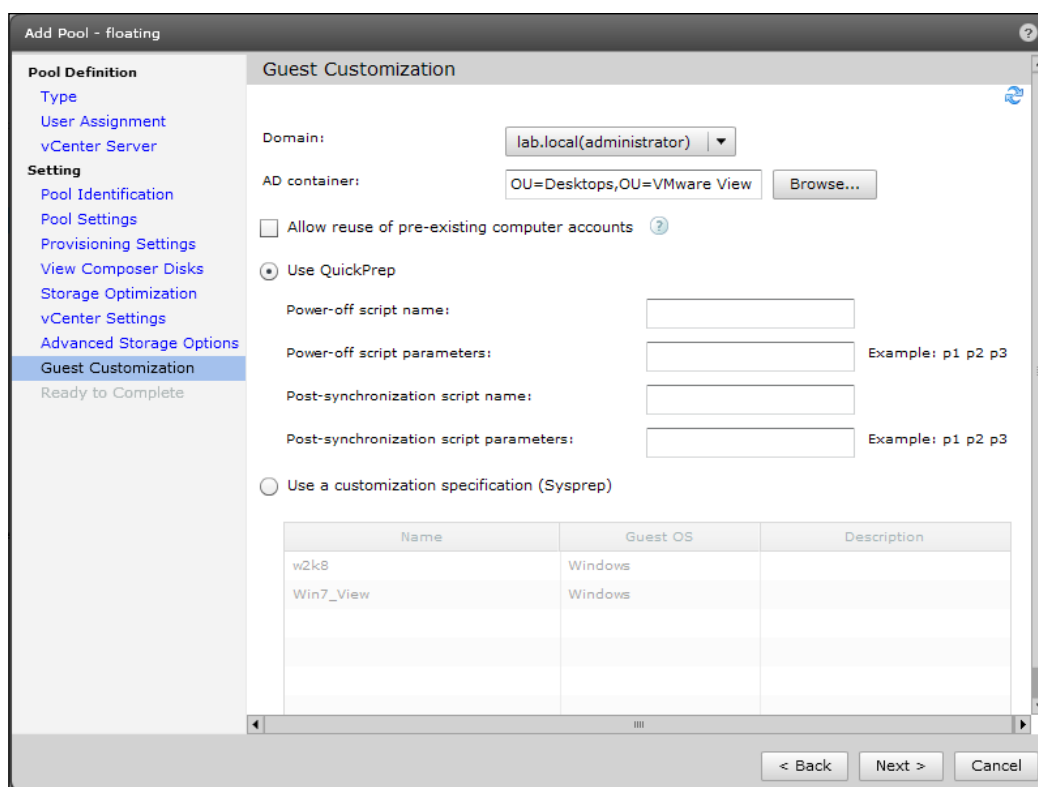
Datastore	vCenter Server	Path	Capacity (GB)	Free Space (GB)
esx01-datastore2	192.168.2.5	/lab.local/esx01-datastore2	465	464
esx01-datastore-SSD-01	192.168.2.5	/lab.local/esx01-datastore-SSD-01	114	40
datastore-nexenta_pool_01	192.168.2.5	/lab.local/datastore-nexenta_pool_01	318	255
datastore-NFS-public	192.168.2.5	/lab.local/datastore-NFS-public	1,391	789

A new right-click functionality has been added to the user interface to help streamline the process of managing desktop pools, entitlements, and desktops.



Support of pre-created Active Directory Machine Accounts

The ability to utilize pre-created Active Directory computer objects is a great addition for organizations that need to create their own Active Directory computer accounts due to security guidelines, or because of automation processes in place to ensure that Active Directory objects are created upon the user joining the organization, as an example.



The preceding screenshot shows the configuration tab for allowing the use of pre-existing Active Directory computer accounts.

VMware vCenter and View Composer Advanced Settings

The VMware View user interface now allows administrators to specify the maximum concurrent number of provisioning and maintenance operations. Previously, only power and vCenter concurrent operations were available for configuration via the user interface.

It is recommended not to change the default settings in the production environment as it could affect the user experience; this is because an environment under heavy provisioning or maintenance tasks could generate significant IOPS, impacting all of the current users.

The screenshot shows a dialog box titled "Edit vCenter Server". It contains two main sections: "vCenter Server Settings" and "Advanced Settings".

vCenter Server Settings

- Server address: 192.168.2.5
- User name: lab.local\administrator
- Password: *****
- Description: (empty text box)
- Port: 443

Advanced Settings

Specify the concurrent operation limits.

- Max concurrent vCenter provisioning operations: 8
- Max concurrent power operations: 5
- Max concurrent View Composer maintenance operations: 12
- Max concurrent View Composer provisioning operations: 12

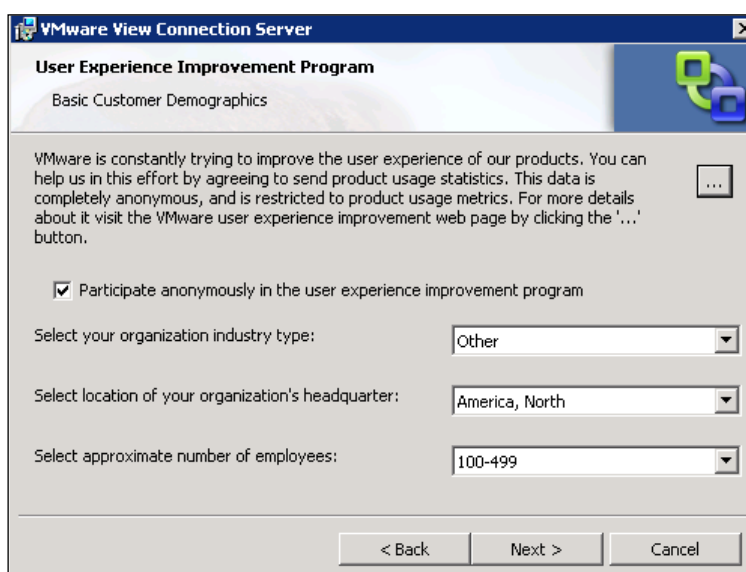
At the bottom right, there are "OK" and "Cancel" buttons.

The preceding screenshot shows the configuration tab for specifying advanced operation maximums.

Phone home

This is an opt-in option during install time for anonymous VMware View usage statistics collection. All data is made anonymous and untraceable, and phone home will collect information on versions, features used, system architecture choices, and deployment scale.

VMware aims to use this information to provide better support and more enhancements to the most popular features. In addition, VMware believes that this data collection will allow for better alignment of View product R&D priorities to match the customer use out in the field.



The preceding screenshot shows the configuration tab for enabling phone-home support (**Participate anonymously in the user experience improvement program**).

Persona management

Persona management has also received enhancement in VMware View 5.1. While many organizations will continue to use third-party solutions such as Liquidware Labs ProfileUnity, the native persona management in VMware View is starting to evolve into a more acceptable solution. The enhancements in View 5.1 to persona management include:

- Allowing virtual profile management of physical machines
- A one-time Windows XP to Windows 7 migration capability

VMware View Persona Management now offers profile support on physical machines to help a user's transition from physical to VMware View desktops. As mentioned earlier in this book, extracting the user profile from a vDesktop (as is typical in a non-persistent solution) allows for an easier migration from a physical desktop to a vDesktop.

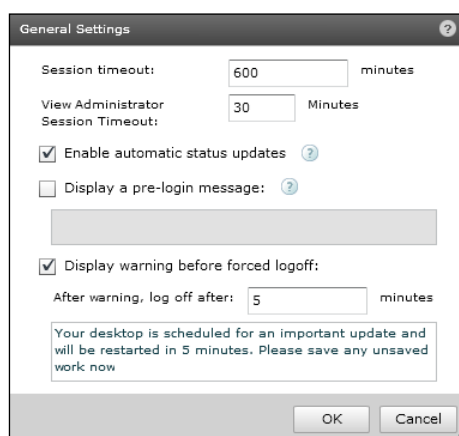
During a physical to virtual migration, the administrator can first install View Persona Management on the physical desktop, and when the user uses a virtual desktop with persona management enabled, user data and settings are automatically synchronized.

VMware also provides support for a one-time Windows XP to Windows 7 migration.

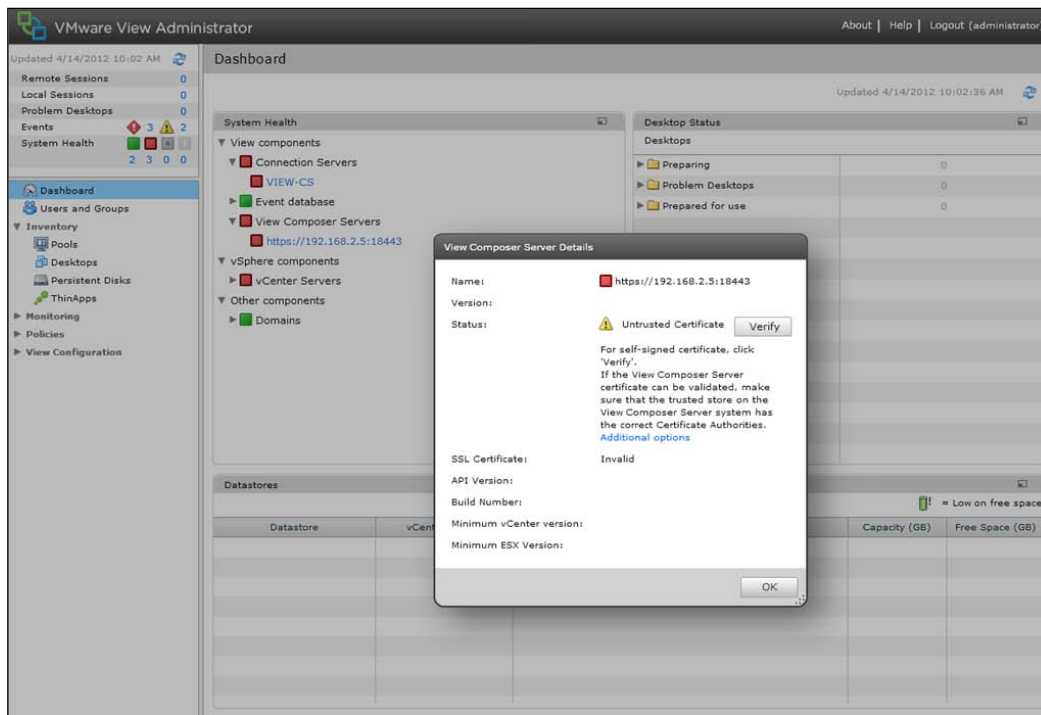
Security

Security was a big topic addressed in VMware View 5.1, including security hardening and fixes. Security is a big driver for many organizations to adopt VDI. The continued improvements in the security area of the product highlight VMware's commitment to highly-secured environments. The release of View 5.1 includes the following security enhancements:

- Support for multiple two-factor authentication with RADIUS Support, including vendors such as RSA SecurID, VASCO DIGIPASS, SMS Passcode, SafeNet, and others.
- **VMware View Administrator session timeout:** The session timeout option was always there, but now it is an option that can be configured by the administrator.
- **SSL certificate security enhancements:** VMware View 5.1 now alerts when self-signed certificates are used. Administrators will have to validate the use of self-signed certificates. VMware recommends that trusted **certification authority (CA)** services should be used.



The preceding screenshot shows the configuration tab for enabling the VMware View Administrator **Session timeout**.



The preceding screenshot shows an error generated from using self-signed certificates.

Summary

VMware continues to evolve the View product through acquisition, internal R&D, customer feedback, support from partners such as Teradici, and involvement from the community. As this book was already completed and in the final publishing process when View 5.1 was released, the authors felt it was important to have an overview of the new content. For additional reading, please refer to the blogs <http://myvirtualcloud.net/> and <http://www.thinkvirt.com/>.

Additional Tools

Every VDI architect has his/her own preferred toolset. This toolset may include I/O stress test tools such as Iometer or certain Visio stencils. This appendix provides a few recommendations for tools that can be used to help in the design process. While no design requires the use of these tools, they may prove helpful to many.

VMware RAWC

VMware RAWC (now known as VMware Desktop Planner) is a tool used to stress a VMware View environment both in terms of PCoIP traffic as well as generated workload on the vDesktops. This tool is available to VMware Partners only and is made available through the VMware Partner Central website. However, individual consultants should be able to sign themselves up as low-level VMware Partners and receive access to the tool. You can get this tool at <http://www.vmware.com/>.

VDI Fox

VDI Fox is an iOS-based application created by the authors of this book. VDI Fox uses the logic found in the myvirtualcloud.net VDI calculators that have become the industry standard, and made them available in the palm of a VDI engineer's hand. In addition, VDI Fox steers its user through a VDI design with a questionnaire to help VDI engineers of all levels navigate their way through VDI.

VDI Fox is expected to be available in the iTunes AppStore in the June 2012 timeframe. For the latest on VDI Fox, please check <http://www.redfoxllc.com/> or follow the Red Fox twitter account at @RedFoxLLC.

Websites and social media

The Internet has a wealth of information related to VMware View, thanks to a very active community both on websites, Twitter, and the VMTN forums.

The following websites provide fantastic content around the VDI technology:

- <http://itbloodpressure.com/>
- <http://www.vladan.fr/>
- <http://blog.simonbramfitt.com/>
- <http://communities.vmware.com/index.jspa>

In addition, using social media is a great way for organizations to understand what others are doing in the VDI space (use the #VDI hashtag to filter based on VDI-related topics). The following people are recommended on Twitter:

- @dlink7
- @simonbramfitt
- @brianmadden
- @eternjanin
- @vladan
- @vdiinfo
- @vmwjourney
- @thinapp_pso
- @terastu
- @roidude
- @ronoglesby
- @vmwareview

In addition, VDI-related conferences, for example, BriForum, should provide a wealth of information around the current trends in VDI.

For an updated list of recommended websites, iOS app info, and Twitter followers, please visit <http://www.redfoxllc.com/> and browse the section dedicated to this book.

Index

Symbols

.log file 147, 148
.nvram file 147
.vmsd file 147
.vmsn file 147
.vmss file 147
.vmxf file 147
.vmx file 147
.vswp file 147
-flat.vmdk file 147
-internal.vmdk file 148

A

actions, linked clones
 Rebalance 21
 Recompose 20, 21
 Refresh 20
 Reset 20
affinity 123
anti-affinity 123
architecture types, pod concept
 traditional 103, 104
 traditional in modular form 104
assessment
 components, metric collection 32
 components, questionnaire 29
**assessment worksheet, Desktop
 Virtualization Solutions**
 desktop pool inputs 29, 30
 landscape inputs 30
 network inputs 31
 profile management inputs 32
 storage inputs 30
 success criteria 32

Atomic Test and Set. *See* ATS
ATS 161
audio video interleave (AVI) 81
auto storage tiering 148

B

Bill of materials. *See* BOM
BOM 101
boot storm 50
bring-your-own-device (BYOD) 197

C

C: \Users folders
 All Users 222
 Default User 222
 Username 222
cache, components
 dynamic cache 236
 in-memory cache 236
CBRC
 about 236
 host memory sizing 239, 240
 managing 240, 241
 storage sizing 238, 239
Certificate Authority (CA) 202
Certificate Revocation List. *See* CRL
cloud 64
Cluster 8
compute
 about 97, 99
 cluster maximums 99
 maximums 99
 memory maximums 99
 single processor with 6 cores 98

Connection Server Restrictions dialog box

- No restrictions option 215
- Restricted to these tags option 215

Continuity of Operations. *See* COOP**converged virtualization appliances 105****COOP 132****Copy function**

- use, avoiding 212, 213

core components, VMware View

- about 8
- vCenter Server 8, 9
- View Agent 12
- View Client 12
- View Connection Server 10

CPU Ready 37**CPU Wait 37****CRL**

- use, configuring 211, 212

cross-zone communication

- permitted 189
- prohibited 189
- restricted 189

Ctrl + Alt + Delete keystrokes 245**D****DAS 138****data, processing**

- CPU usage 36
- disk activity 38
- disk read/write percentage 38
- disk throughput 38
- graphics intensity 39
- memory usage 36
- monitor count 39
- network latency 37
- network throughput 37
- unused applications 39

Dedicated Attached Storage. *See* DAS**Dedicated Replica Datastore option**

- using 171

delta disk 19, 145, 151**design overview**

- about 41, 42
- connection infrastructure 47
- data store level 43
- data store level, benefits 44

end devices 48**networking 44****people 48****storage 42, 43****user persona management 46****View desktop pool Infrastructure 45, 46****VMware vSphere infrastructure 45****DFS 132****DFS-R 132****DHCP****about 32, 92, 93****reallocation 93****workflow 92****disk****OS Disk 17****types 16****User Data Disk 17****disk I/O profile. *See* IOPS****disk types****about 173****diagram 18****for VMware View deployments 173****virtual disk-to-disk type relationship 173****disk types, VMware View deployments****Fibre Channel and SAS 173****Serial Advanced Technology Attachment (SATA) 173****Solid State Drives (SSDs) 173****disposable disk****about 17, 152, 153****temporary internet files 153****Windows paging files 153****Distributed File System. *See* DFS****Distributed File System Replication. *See* DFS-R****DoD CAC****about 209****configuring 209, 210****DRS 7, 87****dvPortGroup 95****dvSwitch 94****dynamic binding****about 96****advantage 96****disadvantage 96****Dynamic Host Configuration Protocol. *See* DHCP**

E

end device

- about 67
- choosing 73
- other clients 73
- thick clients 68
- thin clients 70
- zero clients 71

ephemeral binding

- about 96
- advantage 97
- disadvantage 97

F

FCoE 160

Fibre Channel over Ethernet. *See* FCoE

firewalls

- about 188
- Network firewall (external/DMZ) 189
- Network firewall (internal) 188
- rules 189-191
- Virtual firewall 189
- Windows OS firewall 188

firmware revision (REV) 200

Folder 9

Forensics

- about 217
- user-manipulated data, primary locations 217, 218

FQDN 207

frames per second (FPS) 79

FT

- about 128
- using, design impact 129-131

full clone 16

full clones, sizing

- scenarios 174, 175

full virtual machine 131

fully qualified domain name. *See* FQDN

G

gold templates

- backing up 232

Graphical User Interface (GUI) 131

H

HA 7, 115

hard disk drives (HDDs) 163

hard PCoIP host 79

High Availability (HA) 9

Homeland Security Presidential Directive

- 12. *See* HSPD-12

host 116

host video decoding 80

HSPD-12 209

I

ICMP 113

Input/Output Operations Per Second. *See*

IOPS

installation configuration, View Connection Server

- full option 11
- replica option 11
- security option 11

internal disk 150, 151

Internet Control Message Protocol. *See*

ICMP

Internet Protocol (IP) 92

I/O distribution 169, 171, 172

IOPS

- about 33
- baselines 164

IOPS calculation

- formulae 172

I/O storage profile

- about 163, 165
- sizing, metrics 163

I/O testing

- screenshot 166

J

jailbreak scenario 194, 195

Java Runtime Environment. *See* JRE

JRE 211

K

key phases, VMware View solution

- assessment 28

- design overview 41
- validation 48, 49

kiosk 32

L

LAN 89

latency 137

LDAP 11

Lightweight Directory Access Protocol. *See* **LDAP**

linked clone 14

linked clones, sizing

- Parent VM 177

- replicated disk 177

- scenarios 178, 180

linked vCenter servers

- about 105

- prerequisites 106, 107

load balancing 127, 128

Local Area Network. *See* **LAN**

locked.properties file 212

login storm 50

lossless representation

- importance 78

M

MAC 15

Media Access Control. *See* **MAC**

metric collection

- about 32-36

- data, processing 36-39

- goal 33

- list 34

Microsoft Network Load Balancer. *See* **NLB**

Microsoft's Remote Display Protocol. *See* **RDP**

Microsoft Windows XP Embedded. *See* **XPe**

MMR

- about 80, 81

- benefit 81

- disadvantages 81

- supported file types 80

Multimedia redirection. *See* **MMR**

multisite VDI solutions

- about 60, 61

- profiles 63, 64

N

NAT 194

network address translation. *See* **NAT**

Network-attached storage. *See* **NAS**

network considerations

- bandwidth 90

- bandwidth provisioning 90

- characteristics 91

- consumables 89

- network 89

- task worker, characteristics 89

Network File System. *See* **NFS**

Network Interface Controller. *See* **NIC**

Network Time Protocol. *See* **NTP**

NFS 161

NIC 95

NLB 128

non-persistent desktop 54

non-persistent vDesktop

- about 57, 58

- advantages 59

- benefits 59

- costs 58

- drawback 60

- examples 58

- for multisite VDI 62

- integrating, with persistent vDesktop 64

- related areas 58

- selecting 65

NTP 106

O

OCSP

- about 211

- use, configuring 212

ODBC 14

one-cable zero client solution 74, 75

Online Certificate Status Protocol. *See* **OCSP**

Open Data Base Connectivity. *See* **ODBC**

OS Disk

- about 17

- full clones 18

- linked clones 18

other clients

- about 73
- drawback 73

P

parent 14

parent vDesktop

- about 131
- snapshots, protecting 132

Parent VM

- about 138
- backing up 232

Paste function

- use, avoiding 212, 213

PCoIP

- about 77, 78
- capable hosts 79
- clients 79
- connection types 79, 80
- connection types, hard PCoIP 79
- connection types, soft PCoIP 79
- features 77
- network fundamentals 78

PCoIP bandwidth floor 91

PCoIP maximum bandwidth 91

PCoIP software host 79

persistent desktop 53, 54

persistent disk 154-156

persistent vDesktop

- about 54, 55
- cost 56
- examples 55, 56
- integrating, with non-persistent vDesktop 64
- related areas 55
- selecting 65
- solution concept 119

persona management

- about 250
- View 5.1 enhancements 250

PET

- importance 78

PKI 207

platform features, VMware View 5.1

- 32 hosts in cluster 243

- CBRC 236

- disposable disk drive letter 244

- Standalone View Composer Server 243, 244

- VCAI 242, 243

pod concept

- architecture types 103
- components 103
- linked servers 105
- linked vCenter servers 105
- vCenter Servers 107
- VMware Update Manager Servers 111

Pool Settings tab 214

port binding

- about 95
- dynamic binding 96
- ephemeral binding 96
- static binding 95
- types 95
- VMware View Composer 97

Positron Positron Emission Tomography.

- See* PET

ProductID (PID) 197

profiles

- building, steps 222
- Liquidware Labs ProfileUnity 225
- local profile 222
- mandatory profile 222
- profile folder 222
- Registry hive 222
- roaming profile 222-225
- subsequent logins 223

Proof-of-Concept (PoC) 14

proper device

- one-cable zero client solution 74
- selecting 73, 74

public key infrastructure. *See* PKI

Q

questionnaire 29

R

RAID type

- using 167

random access 169

RAWC *See* also VMare View Planner

RAWC 49

- RDP** 77
- read/write I/O ratio**
 - about 165-167
 - diagram 166, 167
- Recompose action**
 - about 20
 - using 21
- Recompose option** 21
- recovery time objective.** *See* RTO
- Reference Architecture Workload Simulator.** *See* RAWC
- Refresh action** 20
- regression** 237
- Remote Procedure Call.** *See* RPC
- replica**
 - about 15
 - installation types 126, 127
- replica disk** 149, 150
- replica-GUID.vmdk file** 148
- replication** 23
- Reset action** 20
- Resource pool** 9
- rollback method**
 - about 24
 - actions 24
- RPC** 106
- RTO** 12, 124

S

- SAN** 121
- secondary OS Disk** 17
- Secure Sockets Layer.** *See* SSL
- security** 251, 252
- sequential access** 169
- sizing exercises**
 - full clones, sizing 174
 - linked clones, sizing 177
- sizing process** 87
- SKU** 8
- slot fragmentation** 117
- smart card authentication**
 - about 201-205
 - ActivClient console 203
 - advantages 201
 - configuration, for VMware View Connection Servers 206, 207

- configuration, for VMware View Security Servers 208, 209
- CRL 211
- DoD CAC, configuring 209
- requirements 202
- via ActivClient 205
- smart card authentication configuration, for VMware View Connection Servers**
 - about 206
 - advanced options 206
 - environment, configuring 207, 208
- smart card authentication configuration, for VMware View Security Servers** 208, 209
- Smart Card Info object** 204
- soft PCoIP** 79
- Solid State Drives.** *See* SSD
- solution design example**
 - formulae, using 113
 - networking configuration 102
 - physical server requirements 101
 - pod concept 103
 - View Connection Servers 112
- specific files, VMware View**
 - internal.vmdk 148
 - .log 148
 - about 148
 - replica-GUID.vmdk 148
 - VDM-disposable-GUID.vmdk 148
 - VM-s000[n].vmdk 148
- SSD** 137
- SSL** 188
- static binding**
 - about 95
 - advantage 95
 - disadvantage 96
- Stock Keeping Unit.** *See* SKU
- Storage Area Network.** *See* SAN
- storage overcommit**
 - about 156
 - Always option, using 157
 - At y percent of disk utilization, using 157
 - Every x number of days, using 157
 - need for 156
 - Never option, using 157
- storage overcommit level**
 - about 157

- options 158, 159
- storage protocols 160**
- storage tiering feature**
 - about 148
 - delta disk 151
 - disposal disk 152, 153
 - internal disk 150
 - persistent disk 154-156
 - replica disk 149

T

- T1 93**
- T2 93**
- Tag field 215**
- TCP 91**
- TDP 91**
- temp data**
 - non-persistent disk 18
 - OS Disk 18
- template 16, 131**
- Teradici APEX offload card**
 - about 82, 83
 - benefits 82
 - components 82
 - decision tree diagram 84
 - design considerations 85, 86
 - offload tiers, defining 85
 - process, steps 84
- Teradici PCoIP-powered devices 71**
- thick client**
 - about 68, 69
 - advantages 68
 - drawbacks 68, 69
 - examples 68
 - repurposing 69
- thick provisioning 19**
- thin clients**
 - about 68, 70
 - advantages 70
 - drawbacks 71
 - examples 70
- thinkvirt tag 214**
- thin provisioning 19**
- third-party add-on solutions**
 - selecting 61
- traditional architecture**

- about 103
- diagram 104
- traditional in modular form architecture**
 - about 104
 - diagram 104
 - implementing 105
- transfer server**
 - about 21
 - replication 23
 - requirements 22
 - rollback 24
 - virtual desktop, checking in 23
 - virtual desktop, checking out 22
- Transmission Control Protocol. *See* TCP**

U

- UDD**
 - about 17, 154
 - diagram 226
 - OS Disk 18
 - persistent disk 18
 - use 226
- UDP 91**
- UNC 22**
- Universally Unique IDentifier. *See* UUID**
- Universal Naming Convention. *See* UNC**
- UUID 15**
- USB devices**
 - about 196
 - at end device 197
 - via the Windows operating system 198
 - via View Connection Server 198
- USB filtering**
 - about 196
 - applying to 197
 - End device 197
 - level diagrams 196
 - View Connection Server 197
 - Windows desktop operating system 197
- use case**
 - defining 40
 - design 40
 - determining 40
- User Data Disk *See* UDD**
- User Datagram Protocol. *See* UDP**

- user's personas**
 - about 132, 133, 135
 - migrating 220
 - separating, from operating environment 220-226
 - vCenter Database, component 133
 - vCenter Server, component 133
 - View Composer, component 135
 - View Connection Server, component 134
- users per core**
 - need for 37
- user's persona separation, from operating environment**
 - about 220
 - folder, redirecting 221
 - operational considerations 227, 228
 - profiles, working 222
 - UDD, using 226

V

- VAAI 242**
- VCAI 161, 242**
- validation**
 - about 48, 49
 - VMware View Planner tool 49
- vCenter Server**
 - about 8
 - Cluster 8
 - DRS 8
 - folder 9
 - HA 9
 - Resource pool 9
 - vMotion 8
 - vSphere Client 9
- vCenter Servers**
 - about 107, 110
 - naming conventions, using 107, 109
- vCSH**
 - about 125
 - application 125
 - data 125
 - network 125
 - performance 125
 - server 125
 - using 126

- VDI**
 - about 7, 27
 - inherent security 187, 188
- VDI Fox 253**
- VDI solution**
 - sizing layers, diagram 87, 88
- VDI technology**
 - websites 254
- VDM-disposable-GUID.vmdk file 148**
- VendorID (VID) 197**
- Video Memory Calculator 183**
- View 5.1**
 - enhancements 251
 - security enhancements 251, 252
- View Admin console**
 - data store selection 142
- View Agent**
 - PCoIP connectivity 12
 - persona management 12
 - Single Sign-On (SSO) 12
 - USB redirection 12
 - View Composer support 12
 - virtual printing, via ThinPrint technology 12
- View Client**
 - tasks 13
 - types 12
- View Composer**
 - about 13
 - full clone 16
 - linked clones 14, 15
 - snapshots 14
 - templates 16
 - vCenter's SQL Express Installation, using 14
- View Composer Array Integration. See VCAI**
- View Composer linked clones option 131**
- View Connection Server**
 - about 10
 - installation configuration 11
 - installed services, on full installation 11
 - tags 214-217
 - types 11
 - USB filtering, performing 198
- View Connection Servers 112, 113**

View Storage Accelerator. *See* CBRC
Virtual Desktop Infrastructure. *See* VDI
virtual desktop (vDesktop) 8
virtual enclave
 about 192
 blue enclave 194
 diagram 192
 faculty 193
 green enclave 194
 jailbreak scenario 194, 195
 red enclave 194
 servers 193
 training 192
 virtual networking technologies 193
Virtual Infrastructure (VI) 8
virtualization
 about 7
 benefits 7
Virtual Local Area Networks. *See* VLANs
Virtual Machine File System. *See* VMFS
Virtual Private Network. *See* VPN
virtual switch considerations
 about 94
 distributed vSwitch 94
 port binding 95
 standard vSwitches 95
VLANs 95
VMFS 19, 121
VM LockStatus parameter
 about 139
 screenshot 139
vMotion 8
VM-s000[n].vmdk file 148
VMware Desktop Planner 253
VMware Distributed Resource Scheduler.
 See DRS
VMware Distributed Resource Scheduling.
 See VMware DRS
VMware DRS
 about 122
 Affinity 123
 Anti-affinity 123
VMware Fault Tolerance. *See* FT
VMware HA
 about 115, 116
 isolated vDesktops, actions 116
 local storage, using 121
 need for 116-119
 non-persistent example 119, 120
VMware High Availability. *See* HA
VMware RAWC. *See* VMware Desktop Planner
VMware Update Manager Servers
 about 111
 VMware vCenter Server Heartbeat 111
VMware vCenter
 tasks 10
VMware vCenter Server
 about 124
 components 124
VMware vCenter Server Heartbeat. *See* vCSH
VMware View
 about 126
 backup 229
 core components 8
 designing, missteps 9
 FT 128
 load balancing 127, 128
 multiple virtual disks, using 149
 replica 126, 127
 specific files 148
 storage protocols 160
 storage tiering feature 148
 VMware View Composer 138
 vSphere files 147
VMware View 5.0 181
VMware View 5.1
 about 235
 Active Directory Machine Accounts support 248
 administration 246
 client features 245
 localization 246, 247
 management 246
 phone home 250
 platform features 235
 UI enhancements 246, 247
 user experience 245
 View Composer Advanced Settings 249
 VMware vCenter 249
VMware View Composer
 about 138-146
 management features 138

- Parent VM 138
 - VMware View Composer tasks 94**
 - VMware View Connection Server environment backup**
 - LDF files 230
 - primary options 230
 - security server considerations 231
 - SVI files 230
 - ThinApp repository considerations 231, 232
 - transfer server 231, 232
 - workflow 230
 - VMware View design**
 - storage layer 137
 - VMware View environment**
 - restoring 232
 - VMware View maximums**
 - 1,000 clones per replica 162
 - 8 hosts per vSphere cluster 162
 - 32 full- clones desktops per data store(VMFS) 162
 - 64 to 140 linked clones per data store(VMFS) 161
 - 250 linked clones per data store (NFS) 161
 - VMware View Planner tool**
 - 7-Zip 50
 - about 49, 50
 - Adobe Acrobat Reader 50
 - controller virtual machine 49
 - E-mail server virtual machine 49
 - Internet Explorer 50
 - Java Runtime 50
 - McAfee AntiVirus 50
 - Microsoft Excel 49
 - Microsoft Outlook 50
 - Microsoft PowerPoint 49
 - Microsoft Word 49
 - session launcher virtual machine 49
 - storage platforms, comparing 50
 - Target vDesktops 49
 - Windows Media Player 50
 - VMware View solution**
 - key phases 27
 - VMware vSphere files**
 - flat.vmdk file 147
 - .log file 147
 - .nvram file 147
 - .vmsd file 147
 - .vmsn file 147
 - .vmss file 147
 - .vmxf file 147
 - .vmx file 147
 - .vswp file 147
 - about 147
 - VMware vSphere maximums**
 - working with 100
 - Voice over IP (VoIP) 90**
 - VPN 70**
 - vRAM for vDesktops configuration 181-185**
 - vSphere Client 9**
 - vSphere vStorage APIs for Array Integration. *See* VAAI**
- W**
- Windows operating system**
 - USB filtering, performing 198-201
 - Windows Portable Device. *See* WPD**
 - WPD 198**
- X**
- XPe 70**
- Z**
- Zagg Mate 73**
 - zero clients**
 - about 71
 - advantages 71, 72
 - drawbacks 72
 - examples 71
 - zones 189**



Thank you for buying VMware View 5 Desktop Virtualization Solutions

About Packt Publishing

Packt, pronounced 'packed', published its first book "Mastering phpMyAdmin for Effective MySQL Management" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.packtpub.com.

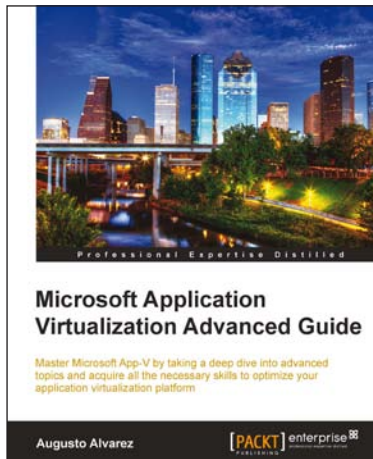
About Packt Enterprise

In 2010, Packt launched two new brands, Packt Enterprise and Packt Open Source, in order to continue its focus on specialization. This book is part of the Packt Enterprise brand, home to books published on enterprise software – software created by major vendors, including (but not limited to) IBM, Microsoft and Oracle, often for use in other corporations. Its titles will offer information relevant to a range of users of this software, including administrators, developers, architects, and end users.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

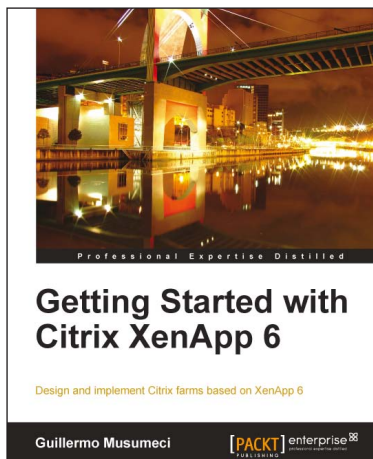


Microsoft Application Virtualization Advanced Guide

ISBN: 978-1-84968-448-4 Paperback: 474 pages

Master Microsoft App-V by taking a deep drive into advanced topics and acquire all the necessary skills to optimize your application virtualization platform

1. Understand advanced topics in App-V; identify some rarely known components and options available in the platform
2. Acquire advanced guidelines on how to troubleshoot App-V installations, sequencing, and application deployments
3. Learn how to handle particular applications, adapting company's policies to the implementation, enforcing application licenses, securing the environment, and so on



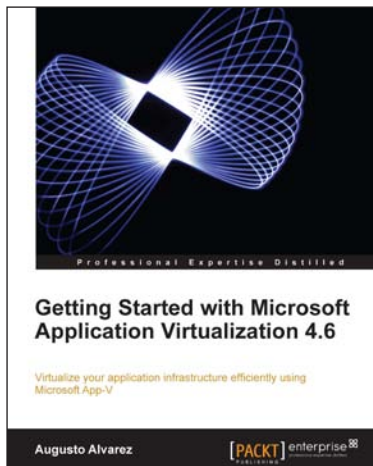
Getting Started with Citrix XenApp 6

ISBN: 978-1-84968-128-5 Paperback: 444 pages

Design and implement Citrix farms based on XenApp 6

1. Use Citrix management tools to publish applications and resources on client devices with this book and eBook
2. Deploy and optimize XenApp 6 on Citrix XenServer, VMware ESX, and Microsoft Hyper-V virtual machines and physical servers
3. Understand new features included in XenApp 6 and review Citrix farms terminology and concepts

Please check www.PacktPub.com for information on our titles



Getting Started with Microsoft Application Virtualization 4.6

ISBN: 978-1-84968-126-1

Paperback: 308 pages

Virtualize your application infrastructure efficiently using Microsoft App-V

1. Publish, deploy, and manage your virtual applications with App-V
2. Understand how Microsoft App-V can fit into your company
3. Guidelines for planning and designing an App-V environment
4. Step-by-step explanations to plan and implement the virtualization of your application infrastructure



Oracle JRockit: The Definitive Guide

ISBN: 978-1-847198-06-8

Paperback: 588 pages

Develop and manage robust Java applications with Oracle's high-performance Java Virtual Machine

1. Learn about the fundamental building blocks of a JVM, such as code generation and memory management, and utilize this knowledge to develop code you can count on
2. Realize the full potential of Java applications by learning how to apply advanced tuning and analysis
3. Work with the JRockit Mission Control 3.1/4.0 tools suite to debug or profile your Java applications

Please check www.PacktPub.com for information on our titles