



**Analysis and recommendations for VDI system with
increased automation on the example of VMware Horizon
View**

By

Student Name: Bartosz Bernat

Department of Computing

Supervisor: Dr Aisling O'Driscoll

February, 2015

Submitted in partial fulfilment of the requirement of the MSc in Cloud Computing

I hereby certify that this material which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work. I understand that my project documentation may be stored in the library at CIT, and may be referenced by others in the future.

ACKNOWLEDGEMENTS

I would like to thank my wife for lots of support, patience, and all the sacrifices she had to make along my whole MSc studies. It was indeed family effort.

A big thank you to all the VMware staff and managers that helped me with guidance and support along the way. Special thanks to Ramesh Subramaniam and Maeve Molumphy.

Last but not least I would like to thank my supervisor Dr Aisling O'Driscoll for guidance and excellent support during the whole course of this project.

ABSTRACT

Desktop virtualization has been the subject of many studies in recent years. It can help reduce total cost of end user system ownership, improve data security and integrity, simplify endpoint management, and provide robust disaster recovery solution by shifting end user data and workloads back to the data center. As capable as desktop virtualization solutions today are, there is significant space for improvement. Missing functionality, poor virtual machine template management, lack of interoperability between systems or need for use of multiple management interfaces are the main culprits that can impede maximizing benefits drawn from the technology.

One of the aims of this project is to analyze the current state of the desktop virtualization solution provided by VMware, its design and current capabilities of multiple virtual machine deployment and management. By analyzing the gaps and bottlenecks in current functionality, management processes, and system scalability areas of improvement are identified and explored. Provided overall set of recommendations can help with building system with improved automation supporting bigger deployments managed by single management console. Additional analysis of the market trends will help understand current technology drivers dictating product development, and likelihood of new features and improvements being implemented in the product.

Secondly, an implementation of a “proof of concept” virtual machine template management automation solution is presented. By overcoming some of the technology and design restrictions of VMware Mirage, improved automation level for multiple images maintenance and update was achieved. The implementation results are compared against benchmarks set using current processes, and show real life benefits of recommendations proposed.

ABBREVIATIONS AND ACRONYMS

AD – Active Directory
ADAM – Active Directory Application Mode
API – Application Program Interface
BR – Branch Reflector
BYOD – Bring Your Own Device
CS – Connection Server
CVD – Centralized Virtual Desktop
DMZ – Demilitarized Zone
EFD – Enterprise Flash Disk
GDL – Global Data Layer
GB – Giga Byte
GPO – Group Policy Object
GPU – Graphical Processing Unit
I/O – Input / Output
IOPS - Input / Output operations Per Second
LAN – Local Area Network
LDAP – Lightweight Directory Access Protocol
MKS – Mouse, Keyboard, Screen
NIC – Network Interface Card
POC – Proof of Concept
RDP – Remote Desktop Protocol
URL – Uniform Resource Identifier
VDI – Virtual Desktop Infrastructure
VIPA – View Inter-Pod API
VPN – Virtual Private Network
WAN – Wide Area network
QoS – Quality of Service

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1. An Introduction	1
1.2. A Thesis Contribution	2
1.3. A Thesis Outline	2
 Chapter 2: Literature Review	 3
2.1. Virtualization	3
2.1.1. Benefits of Virtualization	3
2.1.2. Virtual Machines	4
2.2. Hypervisors	4
2.2.1. Type 1 hypervisor	5
2.2.2. Type 2 hypervisor	5
2.2.3. VMware ESXi hypervisor	6
2.3. vCenter Server	6
2.4. VDI Technology	6
2.5. VMware Horizon View	7
2.5.1. Horizon View Connection Server	8
2.5.2. Horizon View Security Server	8
2.5.3. Horizon View Agent	8
2.5.4. Horizon View Client	9
2.5.5. Clones	9
2.5.5.1. Full Clone	9
2.5.5.2. Linked Clone	10
2.5.6. Template	10
2.5.7. Snapshot	11
2.5.8. Desktop Pool	11
2.5.9. Horizon View Composer	12
2.6. VMware Mirage	13
2.6.1. System Components	13
2.6.2. Centralized Virtual Desktop (CVD)	14
2.6.3. Mirage Server	14

2.6.4. Mirage Client	15
2.6.5. Mirage Branch Reflector.....	15
2.6.6. Multi-Layer Desktop Design	15
2.6.6.1. Base Layer.....	16
2.6.6.2. Application Layer.....	16
2.6.7. Reference Machine.....	17
Chapter 3: System Design and Architecture	18
3.1. VDI as Valid Alternative for Traditional Desktop	18
3.1.1. VDI Drivers.....	18
3.1.2. Customer Expectations.....	18
3.1.3. Market Trends	19
3.1.4. Customer Skepticism	20
3.2. Design and Technological Challenges.....	21
3.2.1. Storage Constrains	22
3.2.1.1. Storage Design Potential Issues	22
3.2.1.2. Storage Cost save – Use of VSAN.....	23
3.2.2. Antivirus Implementation Challenges.....	23
3.2.3. Advanced Graphics Support Constrains	24
3.2.4. Multiple Site Support – Cloud Pod Architecture	26
3.2.4.1. General Overview	26
3.2.4.2. Current Limitations	27
3.3. Administration Challenges.....	29
3.3.1. Multiple Management Interfaces	29
3.3.2. Lack of Functionality	31
3.3.2.1. Lack of Sufficient Virtual Machine Information	31
3.3.2.2. Lack of Power Management Options.....	31
3.3.2.3. Missing Remote Console	32
3.3.2.4. Poor Pool Entitlement Scheme	32
3.3.3. Poor Administrator Permissions granularity	32
3.4. Template Management Challenges.....	33
3.4.1. How Template is Used	33
3.4.2. Template Design – OS Optimization	34

3.4.3. Reasons Behind Maintaining Multiple Templates	35
3.4.4. Initial Investigation and Setting up Baseline for Improvements	36
3.4.5. Template Maintenance and Update Policies	37
3.4.6. Traditional Template Update Process and Its Disadvantages	39
Chapter 4: Environment Development – Overall Recommendations	40
4.1. Multiple Site Support – Cloud Pod Architecture	40
4.1.1. Single Graphical Interface.....	40
4.1.2. Unified Configuration Items Management	41
4.1.3. Improvement of Template Management	42
4.1.4. Overcoming Other Restrictions	43
4.2. Management Console Unification	43
4.2.1. Virtual Machine Configuration Information Availability.....	44
4.2.2. Remote Console Integration.....	45
4.2.3. Administration Challenges - Conclusions.....	45
4.3. Template Management – POC Improvement Proposal	46
4.3.1. Design Choices Explained	46
4.3.2. Improvement Proposal	48
4.3.3. Design Challenges.....	48
4.3.4. Overcoming Design Challenges.....	50
4.3.5. Test Environment Details.....	51
4.3.6. Measuring and Execution Time	52
4.3.7. Implementation Details	52
4.3.8. Experiment Results	55
4.3.9. Discussion Surrounding Implementation Results	56
Chapter 5: Conclusions	58
5.1. Final Conclusions.....	58
5.2. Future Work	60
References	61
Appendix	65

LIST OF FIGURES

Figure 1. Type 1 vs Type 2 Hypervisor design	5
Figure 2. VMware Horizon View architecture.....	7
Figure 3. Full Clone design	9
Figure 4. Linked Clone design	10
Figure 5. Virtual machine snapshot tree structure.....	11
Figure 6. VMware Mirage architecture.....	13
Figure 7. Logical layers of a Horizon Mirage desktop image.	16
Figure 8. VMware Horizon View Single Pod design - 5 Blocks up to 10k VMs	26
Figure 9. Pod local and global permissions span	27
Figure 10. Linked Clone image update process	34
Figure 11. Template update process using VMware Mirage	48
Figure 12. List of testing templates in vCenter Server inventory	51
Figure 13. CVD policy details in Mirage console.....	52
Figure 14. VMware Mirage - Base Layers view.....	53
Figure 15. Base Layer deployment scheme	53
Figure 16. Base Layer deployment process	54

LIST OF TABLES

Table 1. IOPS per VM type	22
Table 2. Update times using traditional approach	37
Table 3. Template update times using Mirage	55

LIST OF CHARTS

Chart 1. Customers VDI expectations	19
Chart 2. Why some customers are not adopting VDI technology	21
Chart 3. Traditional approach vs. Mirage update time	56

Chapter 1: Introduction

1.1 An Introduction

Data centers today wouldn't be the same without well-established virtualization solutions allowing high ratios of consolidation and optimizing usage of server resources. The continuous need to economise and maximize productivity encourages customers to look for mature products that can address their challenges while effectively running their business. Some of these problems can be addressed by virtual desktop infrastructure (VDI). VDI, a growing trend, provides new delivery methods for desktops and applications to end users. Thanks to its design, VDI shifts end user data and workloads back to the datacenter. VDI simply redefines how companies can run their business in the era of mobility, where the need of continuous access to corporate data and computing resources is required. Benefits include centralized management, better security, energy consumption reductions, device independence, ease of patching and updating, which in turn means more cost saving.

However VDI has its limitations. Missing functionality is forcing administrator to switch between multiple consoles to perform their daily tasks. Poor virtual machine template management strategies that usually rely on manual labor multiply repetitive IT administrator tasks. Complicated management processes that if used incorrectly can cause configuration inconsistencies. Additionally mistakes at implementation phase may impede maximizing benefits drawn from the technology.

This thesis analyzes the current state of the VMware VDI solution. Identification of its current problems and already addressed issues will be presented. Recommendations for creating template management process with increased automation will be provided. Recommendations for developing overall simplified management system with improved automation are also going to be presented.

1.2 A Thesis Contribution

The undertaken research addresses three main objectives:

- Analysis of current state of VMware desktop virtualization solution to identify the gaps, bottlenecks, areas for improvement and benefits drawn from already fixed issues.
- Provide an overall set of recommendations to addressing current issues that can help building system with improved automation.
- Implementation and evaluation of a “proof of concept” virtual machine template management automation solution based on suggested recommendations

1.3 A Thesis Outline

The thesis is divided into five chapters:

Chapter 2 includes literature review from the area of virtualization that introduces aspects of desktop virtualization and its place in the virtual infrastructure design. It outlines information relevant to undertaken research and some technical aspect referred to in further parts of the paper.

Chapter 3 analyzes current state of VDI technology. It shows issues and bottlenecks related to current design including management interfaces and gold image management processes. Some of the problems that have been already addressed, and improvements they introduced are also analyzed.

Chapter 4 concentrates on providing detailed analysis of viable solutions to address issues from chapter 3. Improvement suggestions are all presented in this chapter including implementation of template management automation solution.

Chapter 5 consists of final conclusions and results analysis.

Chapter 2: Literature Review

2.1 Virtualization

Virtualization in computing often refers to the abstraction of physical component into a logical object. Use of logical objects can help with improved resource utilization, and ease of its management. Virtualization is present across all branches of IT, from networks where virtual local area networks (VLANs) are used, through storage virtualization (virtual SAN), to server and PC. Virtualization makes decoupling software from hardware possible. This means that the software can run independently to underlying hardware. By removing this dependency greater flexibility, improved availability, and more efficient use of resources is achieved.

The first mainstream virtualization was done on IBM mainframes in the 60s. Later Gerald J. Popek and Robert P. Goldberg codified the framework that describes the requirements for a computer system to support virtualization. They were the first defining the roles and properties of virtual machines (VMs) and virtual machine monitors (VMMs) still used today. It was a few decades later when the first commercially available virtualization solution for x86 platform was released by VMware in 2001. Since then virtualization market has grown exponentially leading to new era of IT. [1] [2] [3] [4] [5]

2.1.1 Benefits of Virtualization

There are many benefits of using virtualization:

- Reduction of operational costs
- Reduced hardware and infrastructure cost by improved resource utilization
- Improved efficiency and speed of system deployment
- Increased availability and security

Virtualization helps reducing number of physical servers. This process is called server consolidation, and its ratio provides information on how many virtual servers are hosted per physical server. Consolidation ratio of 5:1 indicates five physical servers that were virtualized and replaced with only one physical server. That's a big saving of rack space, energy, and cooling. Thanks to virtualization data centers today can largely reduce their footprint leading to overall cost save.

2.1.2 Virtual Machines

At the center of virtualization lies the virtual machine (VM). A VM is a software emulated computer. Like a physical computer it runs an operating system and other applications. It creates tightly isolated container with an operating system and applications running within its boundaries. Each virtual machine is fully independent from other VMs, and completely unaware of other VMs using the same physical hardware. This approach allows for many VMs to run simultaneously on a single physical computer.

A VM is nothing else as a bunch of files residing on local or shared storage. These files consist configuration and system data backed by server physical resources. Once loaded into server memory behave as physical counterpart. Every VM similarly to physical machine has virtual hardware available for operating system (OS) use, this includes virtual CPU (vCPU), virtual RAM (vRAM), and virtual storage, network, and I/O devices.

Thanks to the file packaged design virtual machines are much more flexible and manageable through the use of the traditional file properties that now can be used in a new way. VMs can be cloned, upgraded, moved from place to place, and all that without disrupting user applications. All this gives VMs big advantage over traditional physical systems. [6] [7] [8] [9] [10] [11]

2.2 Hypervisors

Hypervisor can be defined as a piece of software that allows multiple operating systems to exist simultaneously, and use the same hardware. It provides a hardware virtualization platform that allows multiple virtual machines (VMs) to run on a single physical system at the same time. Hypervisor is often referred to as a host because it is responsible for hosting virtual machines (VMs) also called guests. All guest operating systems are not aware of hardware abstraction and use virtual resources in the same way as if they were physical, assuming they are the only operating system owning the hardware. For the guest OS this appears to be its own processor, memory, network or hard disk but is actually only a small portion of the server's hardware resources. Thanks to advanced CPU scheduling and RAM allocation techniques physical server resources can be used to its maximum potential. [1]

Hypervisors can be divided into two categories, those running directly on the hardware – called Type 1, and those using other operating system as their platform – called Type2.

2.2.1 Type 1 hypervisor

It runs directly on the hardware with no operating system beneath it. The first thing to be installed on a server as the operating system is the hypervisor itself. Because of that Type 1 hypervisor is often referred to as bare-metal installation. Lack of any intermediary means that hypervisor can directly communicate with the underlying hardware resources, which minimizes processing overhead. Figure 1 illustrates simplified architectural differences between Type 1 and Type 2 hypervisors.

Thanks to lack of redundant software layers Type 1 hypervisors have better performance characteristics, and are also considered to be more secure due to small footprint that minimizes size of attack surface. All guest operations are handed so that guest OS can't affect underlying hypervisor. A virtual machine can damage only itself, causing a single guest crash, but that does not affect either host or other guest OS, as each VM is contained within own boundaries of virtual resources.

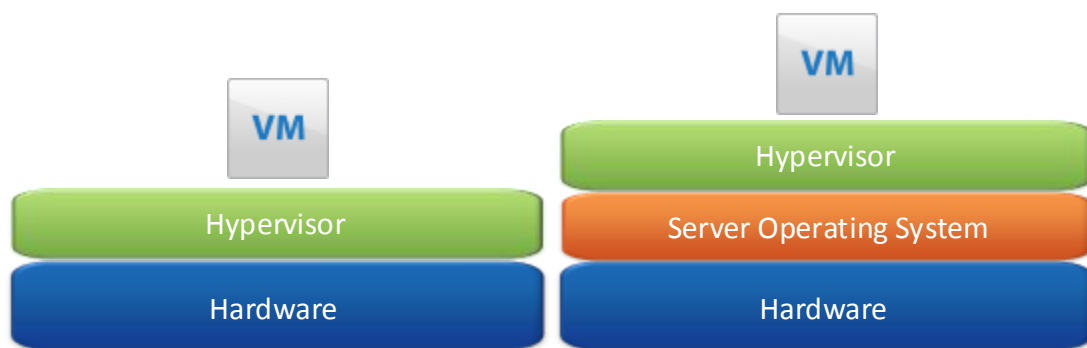


Figure 1. Type 1 vs Type 2 Hypervisor design

2.2.2 Type 2 hypervisor

This is also known as a hosted hypervisor. It is an application that runs on top of traditional operating system like Windows or Linux. Type 2 hypervisor is never installed on the bare-metal as it was with Type 1. The first x86 hypervisors were actually Type 2 because they're much easier to implement due to the fact that the server operating system already handled all of the hardware resources and the hypervisor would only leverage that capability. All requests passed from VM have to go through operating system that handles the hardware. Because of an extra layer between hardware and hypervisor software Type 2 hypervisor has higher latency and resources overhead making it less efficient than Type 1.

Type 2 hypervisors are also less reliable due to more points of failure. Anything that affects the availability of the underlying operating system can potentially impact the hypervisor and the guests it supports. [12] [13] [14] [15] [16]

2.2.3 VMware ESXi hypervisor

VMware ESXi is VMware's enterprise-class virtualization platform bare-metal hypervisor running directly on top of physical server. ESXi abstracts the physical layer and presents it to virtualized servers or virtual machines to consume. Lack of underlying operating system reduces the install footprint to less than 150MB which also minimizes attack surface.

ESXi does more than just provide a platform for running VMs. It enables enhanced availability features, allows better resource usage, and creates new, better ways for provisioning and management of virtual resources.

The most critical component of ESXi is the VMkernel. It is a purpose-built operating system developed by VMware that runs all processes including management applications and agents as well as virtual machines. VMkernel provides functionality for process creation and control, management of file system, I/O stack and device drivers. It runs directly in the memory, and controls all hardware devices on the server, also manages resources for the applications. [17] [18]

2.3 vCenter Server

The vCenter Server provides essential datacenter services such as access control, performance monitoring, storage management, network virtualization, and configuration. It pools multiple server and storage resources allowing sharing these resources among virtual machines in the entire datacenter. VCenter Server is a core component of virtual infrastructure that centralizes management functionality for all parts of the environment. It provides the tools to centrally manage multiple ESXi hosts. It also provides advanced virtualization features, VM and ESXi host failover features, and security mechanisms providing high level of resiliency and virtual infrastructure protection.

2.4 VDI Technology

Virtual desktop infrastructure (VDI) is a variation of the client-server computing model where desktop operating system or an application within a virtual machine is running in the

data center rather than on physical local devices. Users can connect remotely to their virtual desktop or application over the LAN/WAN/Internet from any device like laptop or mobile phone in any location. [19]

The key point to remember is that with VDI solution data never leaves the data center. Only image from the virtual desktop's display gets sent over the network, and the keyboard strokes, and mouse movements from the endpoint device are sent back to the virtual desktop machine. This approach guarantees high data security, as all operations are executed within data center boundaries and minimizes network bandwidth requirements that can now be minimized to as little as 50Kbps per virtual desktop when over WAN.

2.5 VMware Horizon View

Horizon View is VMware's leading VDI technology implementation for corporate environments. Backed by vCenter Server resource management tools and powered by ESXi hypervisor. It is a robust solution enabling automated deployment and management of thousands of end user virtual desktops.

There are several components building Horizon View infrastructure, some of them might be omitted depending on the size of the planned deployment. Figure 2 illustrates typical components.

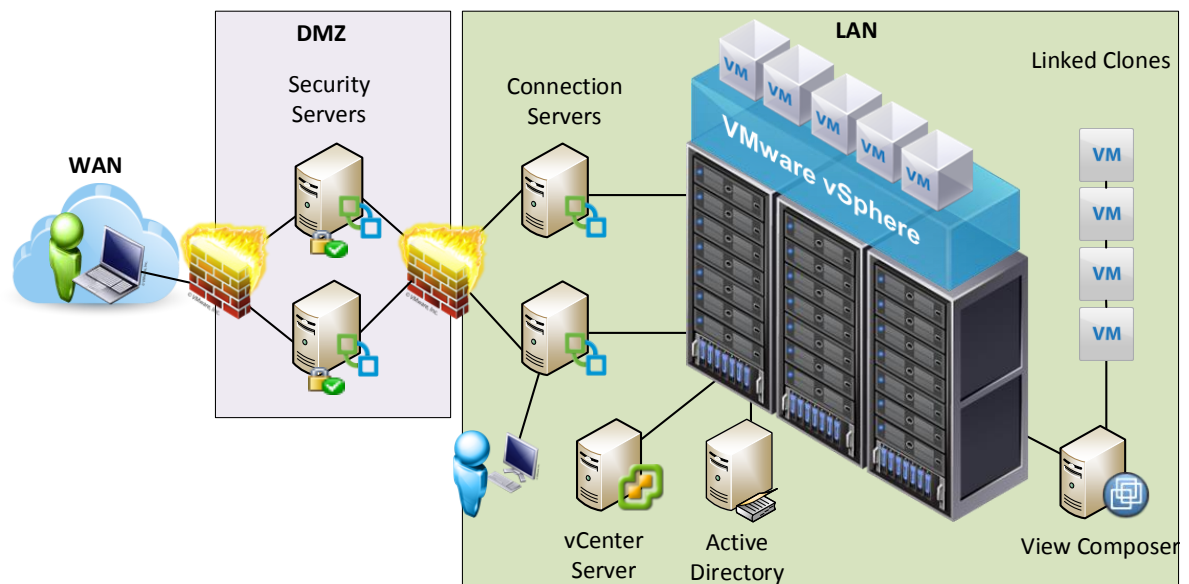


Figure 2. VMware Horizon View architecture

2.5.1 Horizon View Connection Server

Connection Server (CS) is the central point of the VDI solution. It is used to manage most infrastructure components. System administrators connect to CS using internet browser interface to monitor, manage, and change Horizon View deployment configuration. End users use CS as a gateway to connect to their virtual desktop stored in the data center.

Normally multiple connection servers are deployed and masked behind load balancer. This helps with better handling of active connections, and provides redundancy in case of failure.

2.5.2 Horizon View Security Server

Security Server is a type of CS that adds extra layer of security between the Internet and internal network. A security server resides within a DMZ and acts as a proxy host for connections inside corporate network. The reason for this is to allow end users to securely connect to their virtual desktop from an external network without the need to connect to VPN first. Each security server is paired with an instance of Connection Server and forwards all traffic to that instance. This design provides extra security by shielding the View Connection Server instance from the public-facing. There is no need to implement a security server in purely LAN-based deployment, as users can connect directly using any View Connection Server instance residing in the internal network.

2.5.3 Horizon View Agent

Horizon View Agent is a component that installs on the target virtual desktop. It allows the Connection Server to establish remote connection with user VM. It is also responsible for user session maintaining and ending. Additional features of View Agent include:

- USB redirection – allows locally connected USB devices to appear as connected to the VM
- Virtual printing – is the ability to streamline printer driver management and present printers to virtual desktops to enable print services
- Persona management – feature enabling user profiles management across single or multiple floating pools.

2.5.4 Horizon View Client

A component that installs on the end device like laptop, tablet or Thin Client (TC), and allows end device to connect to a Connection Server. CS directs the device to assigned virtual desktop and initiates remote session. View Client communicates with View Agent and maintains remote session, it also allows additional functionality like USB redirection and Single Sign-On.

2.5.5 Clones

A clone of a virtual machine is an exact copy of an existing parent virtual machine. Parent also referred to as master image is the original machine that was used for copying. Clone VM is completely new system that has its own unique identity different from its parent. Cloning process itself is not a Horizon View feature but vCenter that works in the same way as cloning server virtual machines. The View Composer is taking advantage of this feature and applies the same process to end user virtual machines. There are two different types of clones full and linked, both described in the section below.

2.5.5.1 Full Clone

Full clone is an exact, full-sized copy of the parent. It is completely independent copy of a virtual machine that shares nothing with the master image once the cloning operation is complete. It also operates as a fully independent virtual desktop, entirely separate from the parent. Because full clone is exact copy of a parent VM, it consumes the same amount of storage as the parent. This may introduce cost implications for deploying VDI due to higher storage costs. This type of desktop is used in environments where users need dedicated desktops with ability to highly customize their system e.g. developers.

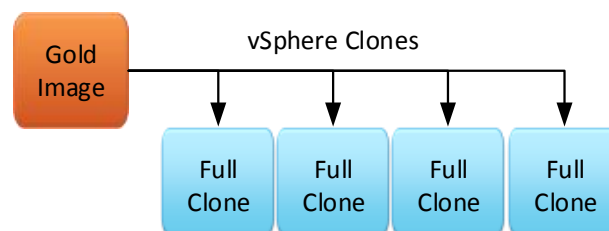


Figure 3. Full Clone design

2.5.5.2 Linked clone

Linked clone is a copy of a virtual machine that continues to indirectly share virtual disks with its parent. Parent is cloned to an object called replica, which is a full clone VM. Replica is further used to back all linked clones. All linked clones are tied to replica that's feeding them as illustrated in Figure 4.

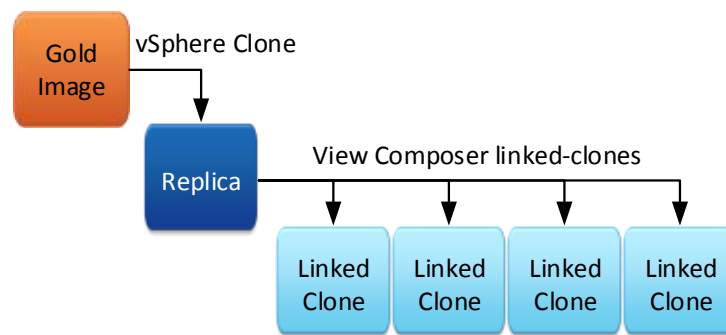


Figure 4. Linked Clone design

Linked clones maintain unique identity, including hostname and IP configuration even though they are only partial copies of the parent consisting only differentiating bits discerning them from the replica. The concept of sharing the same bits of system disk allows linked clones to occupy much less disk space than the parent, and still allow access to software installed on the master image. The use of replica protects master image in case of corruption, giving more flexibility to the parent that isn't tied to linked clones.

Linked clone technology has many benefits like faster and more efficient provisioning times, simplifying patch management and base image updates and enhanced mechanisms for floating desktop pools

2.5.6 Template

Virtual Machine template also called master image, golden image, or parent is a full size virtual machine used for VM clones deployment. Use of single template for multiple clones' deployment dramatically accelerates new virtual machines preparation. IT administrator doesn't have to customize every single VM he intends to deployed, instead cloning can be used to copy already prepared image. All the changes including OS customizations and required end user software is installed on the template. [20] [21] [22] [23] [15]

2.5.7 Snapshot

A snapshot is point in time capture of virtual machine state and data. This includes all the files making up the VM, like hard disks, memory, and other I/O device configuration including NICs and network setup.

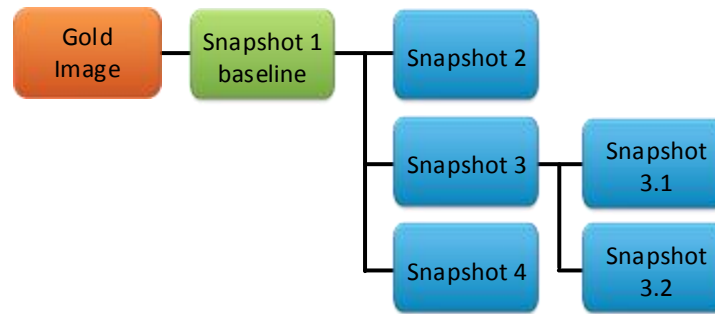


Figure 5. Virtual machine snapshot tree structure

Multiple nested snapshots can exist for a single VM as illustrated in Figure 5. In the chain of snapshots only the differences between previous and current snapshot are captured. By doing so the storage usage is minimized. VM can be reverted to any of its previous states captured in snapshots. Snapshots are inseparable part of composer linked clone deployment process explained in further section. Composer uses gold image snapshots to create replicas that further are used for linked clone deployment. As mentioned before multiple template snapshots can exist and be used at the same time. For each used snapshot separate instance of replica is created to reflect the changes captured in the snapshot. If three different template snapshots are used in a single View Pool, three different replicas will exist to back link clones deployed. [1]

2.5.8 Desktop Pool

A group of similarly configured virtual machines deployed from master image is called a desktop pool. The grouping may be based on department affiliation or a specific use case where the pool consists of virtual desktops configured in a particular way, or consisting particular piece of software. Two types of view pools can be distinguished.

Dedicated View Pool – also called persistent. Users entitled to this type of pool will always log in to the same virtual machine, dedicated for the user. All changes made by user will persist on the VM until IT administrator initiates upgrade or revert back to base image by the means of refresh or recompose discussed in further section. [24]

Floating View Pool – Users entitled to this type of pool log in to first available virtual machine from the pool. The assignment is random and users don't have single system dedicated to their use. By logging in to a VM from floating pool user takes the ownership of random available VM for the time of their session. After logoff user's lock on the VM is removed and VM is released to the pool of available machines, ready to log in by different user. All the changes to OS or user installed applications are removed after user logoff and machine is set back to base image. [25]

2.5.9 Horizon View Composer

Composer is a piece of software that manages the deployment of linked clones from base template snapshots. It also stores all the information regarding replica and snapshot mappings, deployment, or any other action concerning linked-clone desktops. View Composer can create a pool of linked clones from a specified parent virtual machine called template. This strategy reduces storage costs dramatically. Each linked clone virtual machine acts like an independent desktop, with a unique host name and IP address, yet linked clones require significantly less storage. Additional features of composer:

Recompose – process of overwriting linked clone current state with selected template snapshot. This feature is normally used to update linked clones with newer image consisting updates, patches, and software changes.

Refresh – process of overwriting linked clone current state with its own snapshot taken just after linked clone deployment. Linked clone snapshot consists working default linked clone configuration. This composer feature is normally used to fix issues with lined clones caused by end user changing VM configuration, or removing software components. There is no change to base image snapshot when refreshing the VM.

Rebalance – it's a recompose followed by redistributing linked clones evenly across used storage. This feature is used for performance optimization and to avoid issues with lack of storage space hosting linked clones. [26] [27] [19]

2.6 VMware Mirage

VMware Mirage is an alternative to desktop virtualization technologies hosted on datacenter. It can help IT to centrally manage, and customize desktop images of physical computers, and allow end users to personalize their desktops. Mirage can back up entire PC contents for management and protection purposes, keeping endpoint changes synchronized with saved desktop images in the datacenter. Changes and execution of operations are local to the endpoint, and the PC can be taken off line any time. The term endpoint is a generalization for any Windows end device like PC, laptop or virtual machine that is going to be managed by Horizon Mirage. Horizon Mirage is ideal for handling persistent, personalized physical desktops. Linked-Clone technology is not supported. [19]

2.6.1 System Components

There are several components building Mirage distributed infrastructure.

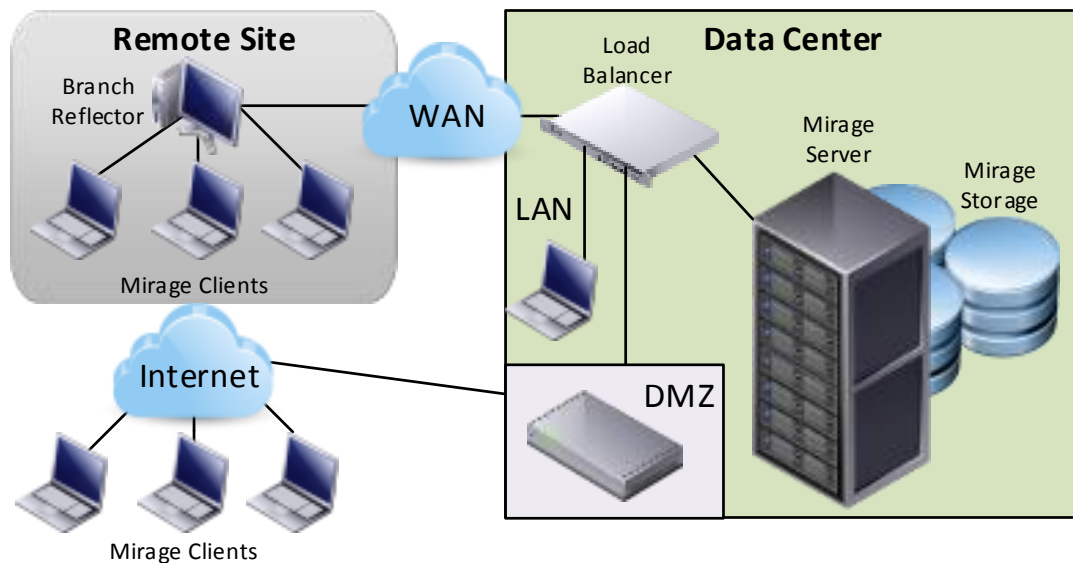


Figure 6. VMware Mirage architecture

The following relationships between the system components exist:

- Clients connect to a Mirage server, either directly or through a load balancer.
- The administrator connects to the system through the Mirage Management server.
- Mirage servers and the Mirage management server share access to the back end Mirage database and storage volumes. Any server can access any volume.

2.6.2 Centralized Virtual Desktop (CVD)

Is an object that contains backup of all PC content, it can also be called desktop image. All PC data building the CVD is synchronized and stored on the Mirage server providing maximum protection and IT standards compliance. Only one desktop image for each endpoint is stored in the datacenter and changes are saved as incremental snapshots that build on this base desktop image. CVD snapshots are different from VM snapshot by design. The use of term snapshot in Mirage terminology means point in time incremental backup of CVD content that enables restoration of endpoint or its files to this point. CVD snapshots are always automatic and taken at configurable interval.

Other important role of CVD is to help centrally manage, update, patch, back up, restore, troubleshoot, and audit the desktop in the data center, no matter if the endpoint is connected to the network or not.

The term CVD shouldn't be confused with VDI (Virtual Desktop Infrastructure) or VM (Virtual Machine) as there are several basic differences between how these are handled on the data center side. The most distinct difference is that Mirage endpoint is not a display of the remote desktop image centralized in the datacenter. Users always use local resources to perform daily tasks and can work off line at any time if required. There is no need to maintain constant connectivity with data center. The endpoint is not virtual, as opposed to VDI.

2.6.3 Mirage Server

Mirage server, is located in the data center, and is responsible for data synchronization between the Mirage client and the datacenter. Additional tasks performed by Mirage servers include:

- Storage management where the CVDs are kept
- Layer delivery including: base layers, application layers
- CVD deployment to clients
- Monitoring and communications management

Servers can be either virtual or physical machines, as long as the machine is dedicated for the Mirage server software only. The main tool used to administer Mirage components is

the management console. It provides graphical user interface for maintenance, management and monitoring of deployed systems including layers creation, update and deployment.

2.6.4 Mirage Client

Mirage Client is installed on all endpoints that are managed with Horizon Mirage. The Client supports Windows only, and is taking care of the endpoint images synchronization. It manages uploads and downloads between the datacenter desktop image and the endpoint and provides means to push IT layer updates to the end device. The client does not create or emulate a virtual machine. No virtual machines or hypervisors are required.

2.6.5 Mirage Branch Reflector

Branch Reflector (BR) is an optional component that can help with accelerating image download in remote and branch offices. BR provides peering service to endpoint devices on remote sites by acting like caching device for adjacent clients. Thanks to that download and upload of base layer or application layers between BR and data center happens only once, and then can be used multiple times on adjacent endpoints saving time and WAN network bandwidth usage. There is no need for each endpoint to download directly from Mirage server. The biggest benefits of using Branch Reflectors can be noticed during mass base layer deployments or updates.

2.6.6 Multi-Layer Desktop Design

Horizon Mirage divides the desktop image into logical layers. By doing so IT gains more control over creation and managing standardized sets of applications, drivers and updates stored in the datacenter, and further applied to user endpoints. Endpoints are governed by IT policies without compromising user experience, allowing users to install their own applications, and change data on their PCs.

Desktop images contain both the IT-managed layers and the user elements with installed software. IT can push layer updates to the endpoint keeping it compliant with all IT policies. On the endpoint side changes made by the user are communicated to Mirage server which updates desktop images stored in data center. This keeps data synchronized on both ends providing full backup of the endpoint device including user data.

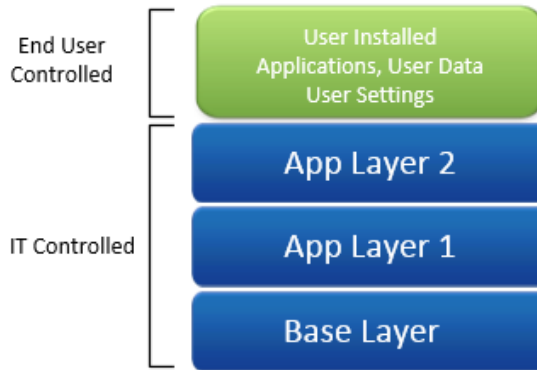


Figure 7. Logical layers of a Horizon Mirage desktop image.

2.6.6.1 Base Layer

A base layer is hardware independent desktop image used for mass deployment. Normally includes the operating system, core applications such as antivirus software, VPN software, firewalls, service packs, and patches, corporate applications common for all target end users, their settings, and customizations.

Because base layer is used as a template for desktop content it is cleared of specific identity information, this makes it suitable for central deployment to multiple endpoints without same identity issues like computer name or unique system ID. A base layer can be captured from a pre-configured reference machine in the datacenter. There is no hard limit on the maximum number of the base layers but as best practices they should be kept to minimum to avoid confusion and multiplying performing the same tasks.

2.6.6.2 Application Layer

Application layers include sets of one or more departmental or line-of-business (LOB) applications, and any updates or patches for already installed applications. They are suitable for deployment to a large number of endpoints, and can be captured from the same reference machine as base layer or from completely different system. Thanks to the use of application layers it is possible to capture only applications themselves, with no underlying operating system. These layers can be managed, and delivered independently of base layer provisioned on the endpoint device. Endpoints can receive multiple application layers depending on the IT policies related to the user owning the system.

The unique nature of application layer reduces the application installation time to minimum, or rather redefines how the software is deployed to the endpoint. None of the

standard software installation steps like installers are present. Application layers are pushed to the endpoint in the background at the file-system and registry level, and users can continue to work with existing applications while a new application layer is being deployed.

IT has total control over the content, assignment, and deployment of base and application layers. Depending upon IT security policies, users may be able to control their own data and settings, as well as manage their own applications but IT always stays in control of all base and application layers. [28] [29]

2.6.7 Reference Machine

Reference machine is a term used to describe Mirage endpoint that can serve to create a standard desktop base layer for a set of CVDs, or to capture application layers, containing business unit specific tools and applications. All updates and changes are made to reference machine and then captured as new or updated layers. Depending on the number of different base and application layers it might be easier to create multiple reference machines as there is no hard limit restricting maximum number of these. It is important to know that reference machine is used solely for creating, updating, and testing layers and not directly to update endpoints. For this purpose CVD of reference machine called reference CVD is used. Reference CVD is stored in the data center and can always be used as a source for base and application layer capture. Multiple base layers, application layers, and their updates can be stored on the same reference machine. Tracking changes is easy and intuitional thanks to the use of names and versions for each layer. All updates can be performed both over the LAN or WAN, using a Mirage reference CVD in the data center. [30]

Chapter 3: System Design and Architecture

Every VDI technology provider like VMware has to continuously adapt their software offering to suit customer needs better. As the trends change, and new functionality is being developed, many technological and design issues have to be overcome. This chapter is an attempt to identify some of the gaps in current VDI functionality and design, and will analyze and explore current state of VMware VDI technology. Overall market trends will also be analyzed to help understand which direction VDI technology is going, and what issues should be addressed first.

3.1 VDI as Valid Alternative for Traditional Desktop

3.1.1 VDI Drivers

Continuous drive towards cost save, better compliance, improved security, and fast system deployment are only a few of the key factors driving VDI technology forward. The increasing need for great flexibility and fast adaptation to new design conditions lead to increasing shift of end user workloads to more efficient, more reliable, centrally hosted environments. This approach makes more sense than ever before. Thanks to possibility of increased automation, the ability to perform many management tasks faster and simpler, patching or updating end user systems can be done almost on the fly. These are only some benefits that drive end users computing solutions back to the datacenter. [31]

3.1.2 Customer Expectations

Customer expectations are the strongest factors that drive VDI technology forward today. The direction of product development is dictated by most common issues customers want to address by implementing VDI. A lot of research was done in recent years by VMware, IBM and other companies. These companies defined seven main customer expectations when addressing end user systems. All of these are present in Chart 1. Coefficients presented are the percentage of surveyed customers who mentioned item as important. [32] [33] [34]

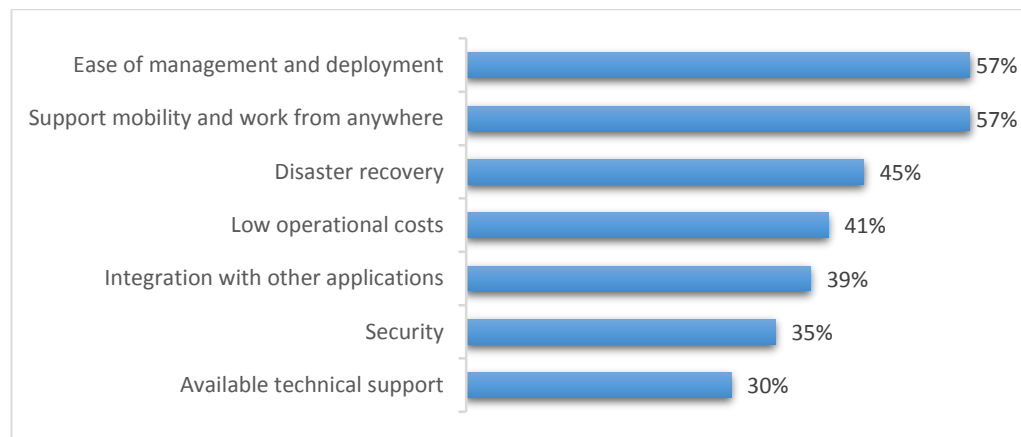


Chart 1. Customers VDI expectations [32] [34] [35]

The increasing need of making IT “greener” and shifting end user workloads to more efficient, more reliable, centrally hosted environments that allow ease of management, and support high level of user mobility are a must. A lot of pressure is also put on system security and disaster recovery. Lastly multiplatform support and interoperability with various systems allowing partial or full integration with current infrastructure is expected.

3.1.3 Market Trends

VDI technologies today can help companies save cost of not only on hardware but also on IT support which currently becomes growing need. This fact was highlighted in recent Gartner’s report regarding IT budget domains indicating that due to the cost of end user computing devices decreasing by 25% in recent years (from \$1,267 in 2009 to \$944 in 2013), personnel cost is the most expensive component of end user computing pillar. [36]

Gartner isn’t the only one that sees the VDI as viable solution. Tech Navio’s recent forecast on global VDI market indicates growing market need for remote access interfaces that can be provided by VDI. [37]

Over the years VDI technologies came long way and evolved to its mature form today. Thanks to many improvements the number of big VDI deployments (10000 VMs or more) in 2014 was almost similar to total number of the same in last 8 years before that. [38] [39]

The VDI growth is expected to break through 1 billion US dollars in EMEA by 2016 and experience strong annual compound growth rate of up to 7.5% for next three years.

Majority of industry experts are convincing that the time of the VDI has finally arrived. Many technology constraints were overcome, and VDI is now able to deliver and adapt to customers' expectations. Customers more often start considering VDI deployment as a viable option in their strategies for different parts of the business. IDC – which is one of the biggest research companies – claims that as a result of growing experiences with virtualization technologies in the data center more customers will move towards adopting VDI solutions. [40]

The Internet is full of case studies and white papers mentioning hundreds of companies that streamed VDI in their environments to deliver value to end users.

“I was very pleased as we did our first pilot of VDI for PARCC testing. The state took notice of Baltimore City, and it was a complete success.”

–Dr. Kenneth Thompson, Chief Information Officer, Baltimore City Public Schools [41]

This is all perfectly true. But VDI adopters should not treat the technology as a silver bullet that will fix all problems and adjust to everyone's needs. VDI has its own use cases along with laptops and mobile devices and can help improve both user experience and increased productivity. This is all due to the fact that we still live in the era of physical devices' dominance for end users. PCs, same as Windows, are not going anywhere any time soon. PCs will still play a major role in end user productivity, as a lot of corporate applications still require traditional desktops. Some of the workloads will shift to a growing number of smartphones and tablets, but that still will not stop VDI from being adopted and take over about 25% of overall corporate Windows desktop market share. [42] [43]

3.1.4 Customer Skepticism

On the other side of the spectrum many skeptics say that VDI adoption is up for debate, and depends on who is asked for their opinion. This is entirely true as not all companies are mature enough to adopt virtualization solutions, or are in very early stages of moving workloads from physical to virtual environment. Chart 2 illustrates a collective research results from recent years regarding slow VDI adoption and its reasons run by Gartner and other market researchers. Coefficients presented are the percentage of surveyed customers who mentioned item as important, and their most common responses for not adopting VDI.

Lack of familiarity with the technology, no funding and legacy application support are some of the main reasons stopping companies from adopting VDI. High cost and complexity provided by other sources are also reflected in Chart 2 as a results of lack of funding, poor familiarity of the VDI technology and no qualified support staff. [34]

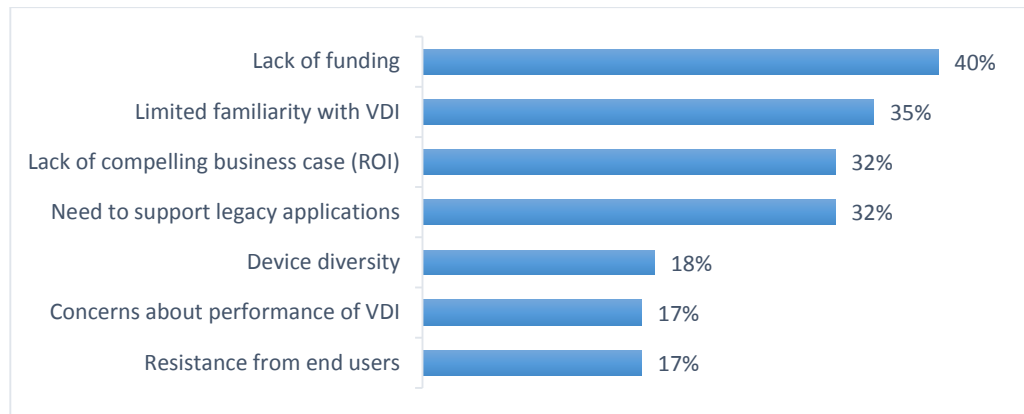


Chart 2. Why some customers are not adopting VDI technology

Regardless of sceptics opinions, during the coming years VDI will be adopted by many other industries. Maturation of this offering will make it more applicable to a wider range of customers. And the benefits of lowering costs, improved security and simplified management will drive highly regulated sectors to adapt VDI on a big scale. [44]

3.2 Design and Technological Challenges

VDI technologies are great for delivering Windows based applications to any device, from any location, 24hours a day. The technology allows ease of patch management of 1000s of VMs, improved security, ease of access, and implementation of BYOD (Bring Your Own Device) initiatives. Desktop management tasks are passed from end users to professionals who have better control over user systems. As great as the technology is already, it still has a lot room for improvements with its downsides and challenges. In this chapter a closer look at technology side of VMware Horizon View is taken. Critical analysis of current solution is going to be presented. Recent improvements and suggestions for future improvements are going to be identified and discussed. [36]

3.2.1 Storage Constrains

For many years storage was claimed to be a big bottleneck when it comes to all VDI deployments. There were two major issues. First was the cost of the storage, second the technology itself in the form of lack of sufficient capacity and speed backing VDI deployments. All this was limiting mainstreaming VDI to big population of corporate users. Over the last three to five years the capacity of disks grown making the cost of GB of storage more affordable to medium size companies. At the same time improvements and new technologies were implemented, making storage more intelligent, and able to achieve horrendous IOPS (Input / Output operations Per Second) speeds. Thanks to the use of block level deduplication, and use of enterprise grade solid state drives also called EFDs (Enterprise Flash Disks), both capacity and speed isn't an issue any more.

Even though it may sound like problems related to storage are no more, it isn't entirely true. The new bottleneck and potential issues shifted to problems with bad storage administration and incorrect design. How Horizon View environment is going to perform highly depends on these factors.

3.2.1.1 Storage Design Potential Issues

Horizon View deployment can't be approached as large scale PC or laptop deployment. Small random-write workloads, boot storms and antivirus scans can cripple storage system in environment with hundreds of VMs deployed. Environment storage has to be scaled correctly to be able to take potential IOPS bursts and increased needs for capacity. When considering right storage preparation IT administrator has to remember that, like with most technologies one size doesn't fit all. To achieve expected performance administrators have to know their user base to assess correctly the amount of required resources. Depending on the type of user, and the tasks they perform some guidelines can be defined.

Item	Task Workers	Power Users	Mobile Users
Average IOPS per VM - Steady State	20	50	15
Average IOPS per VM - Peak/Boot Storm	30	100	25
Reads vs. Writes (R% / W%) Steady State	40%/60%	40%/60%	40%/60%

Table 1. IOPS per VM type

As a baseline average between Steady State and Peak IOPS is good starting point. Table 1 presents typical storage IOPS requirements for a VM depending on the user type. Below data was taken from implementation guide of EMC VNX storage arrays.

When scaling storage important to remember is that the IOPS of each disk building storage array are not equal to IOPS the array can actually support. This is from simple reason, data protection and security mechanisms embedded in the array functionality come with the price of percentage of overall speed and capacity used for the functionality. [45]

3.2.1.2 Storage Cost save – Use of VSAN

In recent year VMware started implementing more solutions that can help with cost saving on expensive storage solutions. The VMware Virtual SAN (VSAN) is hypervisor-converged storage platform. As an optional component of VMware vSphere it allows to use local ESXi host disks to create fully replicated, high-performing, software-defined storage platform that can be used for virtual machines provisioning. Local disks can be mixture of flash disks and spinning disks.

VMware Virtual SAN can provide alternative for customers who don't wish to invest in a traditional shared storage array for Horizon View deployments. VSAN can deliver the capacity and performance needs that the Horizon View deployment require. VSAN allows automated data-tiering capabilities, and reduces the overall number of needed infrastructure components. Thanks to its integration with vCenter Server it simplifies the management of storage that can now be administered using the VMware vSphere Web Client. [26] [46]

3.2.2 Antivirus Implementation Challenges

One of the problems VDI experienced at the beginning was poor integration with antivirus software. The only way to implement antivirus on the endpoint system was to install standard version of selected product just like for typical PC. This brought several issues related to high density of virtual machines sharing the same physical resources.

To manage virtual environment more efficiently some assumptions have to be made. One of them is that in normal operational conditions only certain percentage of virtual machines

actively use physical hypervisor resources, this way high consolidation ratios are possible. All operations are scheduled on shared physical hardware. The situation changes when suddenly all VMs request access to hardware. This can happen when scheduled antivirus scan is initiated on all VMs in the resource pool. If the pool is big enough, say 300 – 500 VMs it can easily use all available CPU, RAM resources and saturate storage making whole virtual environment unusable for the time of antivirus scan. Initially solution for this issue was scheduling antivirus scans at night time or in small batches. This was troublesome, involved more IT resources and was more prone to error if e.g. schedules were setup incorrectly.

As the virtual environments grown bigger the night time scan maintenance windows were not long enough to complete the task. It became clear that it was not feasible to use traditional antivirus approach anymore. That was the time when VMware vShield Endpoint was developed to address the issue. VShield endpoint allows to offload VM scans from endpoint systems to server side. No Client software is required on guest OS. On ESXi side dedicated antivirus appliance is responsible for performing the activity. Each ESXi host is equipped with vShield appliance residing on local storage. All appliances are reporting back, and are managed by antivirus appliance manager. Management console can be accessed by browser interface. Many vendors like Trend Micro and McAfee offer their own versions of vShield appliances.

Offloading scanning operations helped addressing all the issues related to antivirus scan storms that could compromise virtual environment performance. It also improved overall desktop performance as none of desktop resources is used for scans anymore. Additional benefits are improved VM consolidation ratios, centralization of virtual environment antivirus protection and thanks to this improved automation in security policies enforcement by using advanced monitoring functions meeting compliance and audit requirements and detailed logging.

3.2.3 Advanced Graphics Support Constrains

Up to very recent one of the VMware View limitations was lack of support for high quality 2D and 3D graphics. The problem was in the way virtual environment was designed in the first place. As virtual machines resided on ESXi host, and used share resources, there was

not solution allowing GPU virtualization or sharing it across virtual environment. A GPU (Graphics Processing Unit) is highly specialized processor type used for image rendering. It uses parallel processing to calculate many elements of an image at once, comparing to CPUs that only calculate one command at a time.

At some point it became obvious that by not supporting the functionality VMware is limiting its customer base to a group of users in no need for high graphics capabilities like general task workers. And by incorporating GPU support Horizon View becomes attractive to new group of highly specialized personnel like CAD or graphic designers, architects and anyone else in need of desktop with powerful graphics capabilities.

These drivers led to release of first NVidia GPU providing full support for VMware Horizon View back in 2013. NVidia Grid – as this is the name of the product, evolved more in the last 18 months, and now VMware Horizon View GPU support options include:

- Soft 3-D graphics – the GPU is emulated in software, no need for video card
- GPU sharing – server-based GPUs are sliced into virtual GPUs and assigned to VMs
- Pass-through GPU – also called dedicated, it allows a VM to take exclusive ownership of whole GPU for its use.

GPU hardware virtualization on its own wouldn't be successful, if it wasn't for the protocols used for VM remote connection. PCoIP, which is a display protocol from Terra DiCi, started leveraging the capabilities of GPUs, this allowed to send higher quality graphics over lower bandwidth connections without putting much strain on the ESXi host CPUs.

Thanks to GPU virtualization many applications, before off-limits, can now be successfully run on virtual machines. Users can now perform most of the tasks in virtual environment, previously reserved for laptops and PCs. This opens new opportunities for thousands of customers looking for flexible systems at low costs that can deliver secure and controlled environment in conjunction with high performance graphics, and user mobility. [47] [48]

3.2.4 Multiple Sites Support – Cloud Pod Architecture

3.2.4.1 General Overview

Cloud Pod Architecture is a new component that provides tools for joining multiple instances of Horizon View to allow easier management. It also allows users to access multiple desktops from many locations based on their global entitlements.

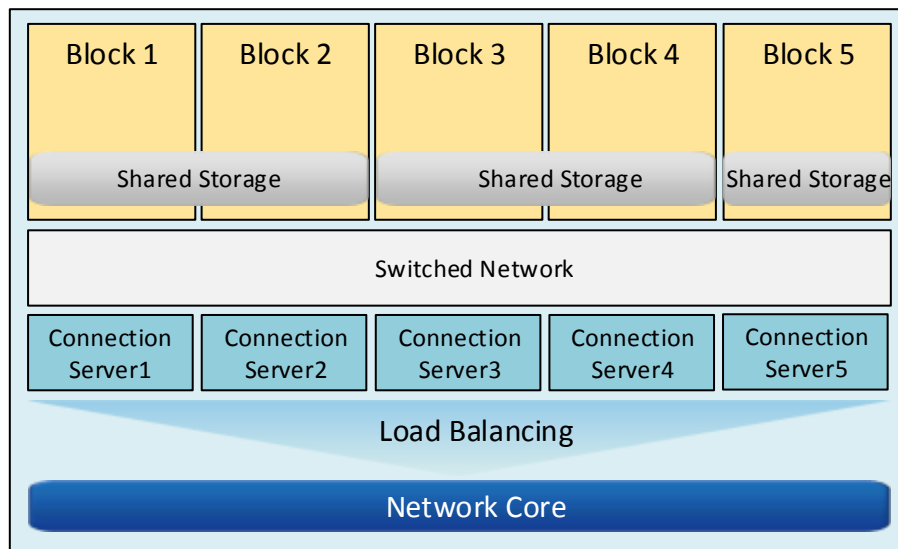


Figure 8. VMware Horizon View Single Pod design - 5 Blocks up to 10k VMs

There is a few components building Cloud Pod Architecture:

- Block – can be defined as fully independent View instance consisting of physical servers, vCenter infrastructure, network, all required Horizon View servers, shared storage, and end user VMs. Single block can support up to 2000 Desktops.
- Pod – is an organizational unit that can integrate between one to five building blocks under single View Manager Installation, and can be managed as single entity. This potentially allows IT administrators to manage up to 10000 Desktops as a single environment.
- Site – is a collection of LAN-connected Pods in the same physical location like datacenter. With multiple physical locations, Pods need to be configured as a separate Site per location.
- Global entitlements – are used to grant access to multiple view pools deployed in Cloud Pod, they span between all pods belonging to Cloud Pod Federation.

- Cloud Pod Federation – is a grouping of multiple Pods that share the same global entitlements.

In a traditional Horizon View implementation, each Pod has to be managed independently. Figure 8 present details of single Pod design. After implementing Cloud Pod Architecture, multiple Pods can be joined together to form a single Horizon View implementation called a Pod federation. The benefit of this is ability to span multiple sites and datacenters. It also allows sharing key data, like Pod topology, configuration items, user group entitlements, and policies, by all View Connection Servers in the federation. This in turn can greatly simplify the administration of a large-scale deployment.

3.2.4.2 Current Limitations

The most important part to know about Pod federation is, that only single Pod can be implemented within the same datacenter, and the same Pod cannot span multiple datacenters. This automatically limits the maximum number of desktops that can be deployed in the datacenter to 10000 VMs. If there is a need for multiple Pods in either the same or different location they can be managed separately or joined into a federation. Cloud Pod Architecture allows to aggregate up to four Pods and two Sites that can be managed using global entitlements. Maximum of 20000 VMs is supported.

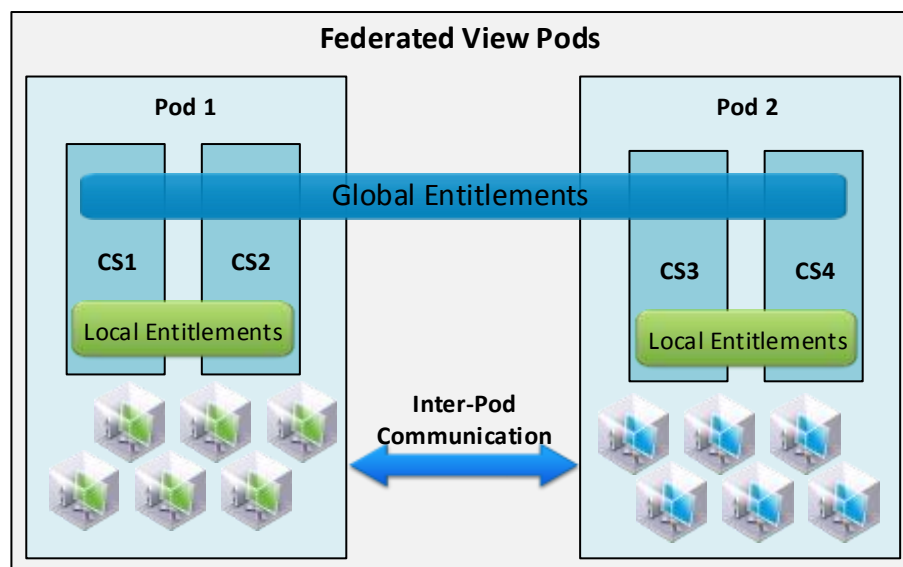


Figure 9. Pod local and global permissions span

In a single Pod configuration View Administrator console is used to assign local entitlements to users and groups for a specific desktop pool. Cloud Pod architecture environment is different. Global entitlements allow users or groups to multiple desktops across all configured Pods in the federation. Global entitlements can exist in parallel with local entitlements or on its own as illustrated in Figure 9.

A few issues surrounding global entitlements can be identified.

First there is no graphical interface and all Cloud Pod configuration has to be done through command line utility called “lvmutil.cmd”. Due to this fact managing multiple global groups is lacking clarity and can be troublesome. Adding new groups is prone to errors caused by typos or wrong command syntax.

Second issue is lack of visibility into global entitlements through the View Administrator console. Global entitlement information is only available through command line utility from View Connection Server instances in the Pod federation. This can cause multiple issues for View administrators not aware of the federation configuration. It can easily happen that there are no entitlements assigned to one of the Horizon View Pools in the local Administrator console which normally is used for single Block management, meaning no users are entitled. However, global entitlements might exist allowing access to the pool. IT Administrator wouldn’t be able to determine that, as in the Horizon View management console message: “No users or groups are entitled to this pool.” is displayed every time local entitlements are missing, even if a global entitlements exists for this pool. This can be very confusing and lead to long issue resolution times as IT support cannot perform right troubleshooting to address the problem.

Third issue is the limitation of connecting maximum four Pods in no more than two different Sites. Because of this some of the big deployments will have to be divided into a few separate federations managed by different set of global entitlements. This potentially can cause difficulties with administration, and multiplies management interfaces.

Another issue is the fact that most of the features of multiple View Pods still have to be maintained separately. This includes both configuration and golden images used for linked-clones deployment. This means that more time has to be spent on maintaining consistent configuration across the Pods, and managing master images across multiple environments.

Lastly even though multiple Pods can be joined into the federation there is no common, fully functional single management interface. Each Pod has its own View Manager Installation and separate management console. The only common information available across all View Managers is the status of other Pods in the federation and their connection servers, indicating if they are on or off line. No information regarding status of desktops or ability to perform any type of administration of other Pods is possible. In the event of user moving between physical locations, VM from different Pod can be assigned to user according to defined Global Entitlements. IT administrator will have to log in to multiple management consoles to find right desktop currently used by the user in case support is required. This multiplies support task that need to be performed by IT administrator, and can lead to longer resolution times.

The issues mentioned above are the main problems that Pod federation design is facing at the moment. Another facts worth mentioning is lack of support for Windows-based applications hosted on a Microsoft RDS host and no HTML Access. [26] [49]

3.3 Administration Challenges

Current design of Horizon View and underlying virtualization solutions influence how virtual environment is managed today. As mentioned before the technology brings many challenges that affect management and administration tasks and procedures. This chapter is an attempt to identify current state of the technology from IT administration point of view and present challenges of effective Horizon View deployment administration.

3.3.1 Multiple Management Interfaces

When VMware Horizon View was first released it was an extension of vCenter capabilities to manage pools of full clone virtual machines deployed from the same template. For that purpose separate management console was developed to allow easier administration, performing power options, and to provide advanced logging functionality. As the time passed new functionality on both vCenter and Horizon View side was implemented. Horizon View product matured a lot, but IT administrator is still forced to use several consoles for virtual desktop management. Today to successfully manage Horizon View deployment IT administrator has to actively work with vCenter, View Administrator Management console, Active Directory, and some additional third party tools. This

potentially causes a few issues. The need of maintaining configuration in a few different systems is more prone to error, and the need of using multiple management interfaces causes that more time is required for troubleshooting.

The main problem with current design is that IT administrator is forced to use multiple management consoles, as required information or functionality cannot be found in a single interface. Good example of the problem is getting and IP address of a VM that was reported as in “Error” state in View Administration Console. IP address information isn’t accessible in View Administrator Console, only virtual machine DNS name is. If IT administrator is working from different network or domain where there are restriction on how Active Directory and DNS information is propagated using command line, and using “ping” to reach VM DNS record will not resolve to IP. The only solution is loading vSphere Client, connect to right vCenter server hosting problematic VM and obtain IP address from there. If the problem has escalated and IT administrator needs a list of multiple IP addresses he will have to switch back and forward between two interfaces to obtain required information.

Use of multiple consoles can affect troubleshooting accuracy and resolution time. They can also cause potential issues with access level inconsistencies. Different level of privileges between Horizon View, vCenter and AD may exist due to IT administrator role in the organization. Normally helpdesk is the first level of support, they will be the once contacted by the user who can’t log in to their VM. If helpdesk engineer is missing access to some components of either Horizon View Administration console, vCenter Server or Active Directory that is reserved to level 2 and up engineers, he will not be able to address the issue correctly. Problem will have to be escalated to level 2 support. Realistically it can take up to one or two days before issue gets resolved. This potentially means user cannot perform their daily tasks for that period, which will affect their overall productivity.

Normally different teams manage access to different interfaces, this can lead to lack of sufficient access across all consoles. Multiple management interfaces can also lead to configuration inconsistencies, and complicated management processes explained in further paragraph.

3.3.2 Lack of Functionality

The need for multiple interfaces mentioned above can often be explained by lack of functionality provided by single interface. Some of the major functionality currently missing in Horizon View Administration console is analyzed below.

3.3.2.1 Lack of Sufficient Virtual Machine Information

Lack of information regarding virtual machine hardware configuration like amount of CPU, RAM, amount of disk space or the datastore on which VM is residing can only be obtained through vSphere client. Same for IP configuration and View VMs resource pool configuration. All this forces IT administrator to switch between multiple consoles as View Management console is missing required information. This is normally time consuming especially with big number of machines that need checking.

3.3.2.2 Lack of Power Management Options

Lack of Power On or Shut Down options in Management console. Even small it is very often used feature. Typically users after day of work shut down their VMs. After VM shutdown is initiated from OS level its status in View management console changes to “Provisioned” to indicate current “power off” state. Unfortunately there is no feature to power the VM back online or shut it down if needed from the View management console. Only user itself can initiate VM boot up on attempt of next log on. Lack of this feature forces IT administrator to log in to vSphere just to perform power on. As a workaround all VMs in the selected resource View pool can be set for “Stay Always on” which means that after OS level initiated shutdown VM will be powered back up by View Composer. This is not a perfect solution as a lot of electricity is wasted for hundreds or even thousands of idling VMs that as well could be shut down. It also creates another problem if View Pool is setup to keep all VMs always on, and IT administrator needs to perform some changes on a single machine while it’s shut down. View composer will attempt to power the VM every minute, this isn’t sufficient to perform any task.

3.3.2.3 Missing Remote Console

Another desired functionality that is currently missing, is remote console that could be used directly from View management interface level. Currently if IT administrator needs to perform any changes to the VM OS he needs to use vSphere to console in, alternatively he can use Remote Display Protocol (RDP) from Microsoft. This solution isn't good as multiplies the number of management consoles that need to be used. Additionally both RDP and vSphere consoles don't allow to watch or take over user's current session. This in turn slows troubleshooting even more and forces IT administrator to use third party tools like VNC remote client or Team Viewer to be able to see what happens on user desktop.

3.3.2.4 Poor Pool Entitlement Scheme

Currently Horizon View pool entitlement for end user can be achieved in two ways. First is by entitling Active Directory security group to Horizon View pool. Further domain users are assigned to the same AD security group to gain access to VM hosted in specified View pool. This scenario is preferred option, as permissions are easily accessible, and reflected in both AD and Horizon View management console. In second scenario end user can get access to specific Horizon View pool by entitling their domain account directly on the pool level, bypassing AD security group requirement. This type of access assignment can cause configuration inconsistencies leading to situation where user has access to a specific VM but IT administrator cannot find the entitlement in AD. The only way to find user entitlement to a specific pool in this case is drilling down through properties of each View Pool. In big environments with dozens of resource pools and hundreds of users assigned directly to Horizon View pool entitlement it can be quite troublesome and time consuming.

3.3.3 Poor Administrator Permissions Granularity

One of the flaws of current Horizon View management console is the poor granularity of permissions in Horizon View management console. Existing entitlements are grouped in larger permission containers, and in some cases it means all or nothing. Because of this reason many simpler tasks that could easily be addressed by first line of support cannot be performed are passed to second or third line of support as only they have sufficient access. This means more experienced IT personnel has to spend more time dealing with simple cases that could potentially be fixed lower tier team. This causes unnecessary overload of

highly trained staff and use of their work cycles, in turn leading to more money spend on IT support.

Good example of the problem is use of privilege defined in “Global Administrators View” called “Manage Composer Desktop Pool Image”. It entitles selected user group to perform all View Composer tasks like Refresh, Recompose, Rebalance, and changing default image for desktop pool. Changing default image, initiating any composer tasks on big number of VMs or whole resource pool can cause service interruption and performance degradation of remaining systems. Because of potential damage that can be done by unexperienced administrator this privilege is limited to Inventory Administrators and above. Increasing granularity of this permission by dividing into four separate permissions would mean that some of the simpler tasks like, refreshing single virtual machine could be performed by first line of IT support, none of the other actions included in the original permission could be performed. Thanks of high granularity and permissions segregation none of the performed tasks could influence other VMs or whole resource pool. The task can be safely performed, and fix applied minimizing resolution time. [25]

3.4 Template Management Challenges

One of many benefits of using technology like VMware Horizon View is the ability to promptly deploy new systems and update hundreds of current VMs with latest patches or new software. This is all possible thanks to the use of golden images also called templates.

There are many factors that can influence how the template is created, and managed. Depending on the end user type, intended functionality, and type of installed software, IT administrator has to overcome many challenges. There are many use cases for maintaining multiple templates. In this paper only some of the use cases concerning VMware Horizon View linked clone technology are going to be explored.

3.4.1 How Template is Used

VMware vSphere was developed with server virtualization in mind. When the need for developing end user solution arose additional components were built around it allowing the functionality. VMware Horizon View uses vSphere infrastructure as its foundation and adds new concept of View composer and linked clones to deliver end user systems.

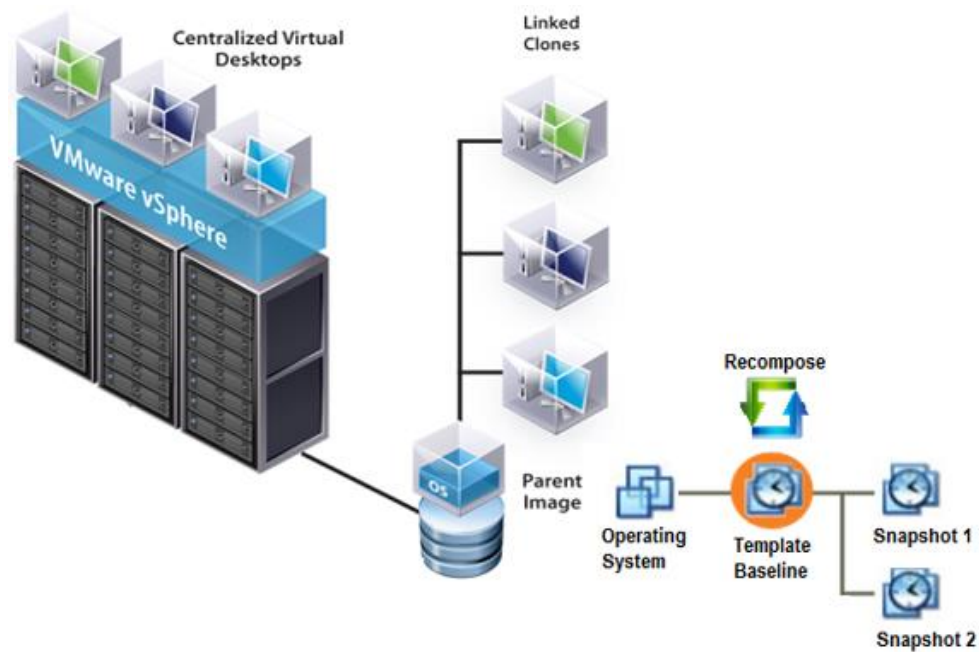


Figure 10. Linked Clone image update process

Parent images also called golden images or templates are the core component of VMware Horizon View design. They are used by View composer for any action that involves linked clones deployment, change, and maintenance by the means of refresh, recompose or rebalance. All the new changes incorporated into the template are captured in snapshots that are further used for linked clone updates. Figure 10 depicts typical components and processes involved in linked clone deployment or upgrade.

As mentioned before in chapter 2 linked clone is a copy of parent virtual machine that continuously shares its virtual OS disk with the parent. Linked clone technology allows manipulating end user systems configuration, by changing the settings of the parent. Thanks to this approach, applying changes to thousands of systems can be done automatically by making necessary adjustments only to golden image that feeds the linked clones. By applying this technique time spent by administrator on making changes to individual systems is reduced to zero. The only systems that needs to be maintained are golden images used for view pool deployment. [25] [26]

3.4.2 Template Design – OS Optimization

Even though end user system maintenance has been reduced, there is still a lot of work involved in the creation and maintenance of templates themselves. Support and management of desktop images has always been time consuming and labor intensive

process. Due to the nature of virtual environment and Horizon View design itself many OS customizations have to be implemented.

The only fully supported OS used for linked clones deployment today is Microsoft Windows. With thousands of built in features, registries and services it is quite complex to manage on enterprise scale. Many of the user experience improvements and features same as the Windows itself were designed with physical resource usage model in mind. A PC or laptop can guarantee dedicated amount of RAM and CPU. In virtual environment all resources are shared, and it is crucial that each running process provides value to the user experience. Because of this many Windows features designed to enhance the user experience may have the opposite effect on the system deployed in virtual environment, by increasing the usage of the shared physical resources like RAM and CPU from the ESXi host, causing poor application performance.

To provide a better user experience, improve the overall scalability, and performance of VMware Horizon View deployment all golden images have to be optimized.

Normally templates have to include below:

- OS changes and customizations including registry and services
- Custom logon scripts
- Security patches and OS updates
- Custom end user software.

Incorrect template preparation and insufficient testing before putting in production may lead to performance degradation and user dissatisfaction.

3.4.3 Reasons behind Maintaining Multiple Templates

One of the challenges related to template management is that one size doesn't fit all. Normally there are many departments in the company, and each department may have different type of users, that again may require different set of application. Different applications may in turn need different amount of resources to work smoothly. Looking at this from Horizon View perspective this means the need for multiple templates supporting different user configurations, both software and hardware.

There are many other reasons for maintaining multiple templates. One of them is lack of technology allowing virtualization of all application packages used by end users, such as those interacting with system services or drivers. Good example of these are all sorts of scanner and print drivers or antivirus software. IT has to create multiple gold images to deal with all these scenarios. Sometimes even if the application can be virtualized, IT doesn't have resources to do that, and continues defaulting to installing the applications on the base image as this is very often much easier and faster.

Another reason is licensing. Most of corporate software is billed per desktop, or OS it runs at. Building all software packages into one gold image would mean that every desktop has licensed software installed. If the licensed software is required only to small group of users this would be a big waste of money spend on never used licenses for other desktops.

Additionally, bundling all applications into single template means every time a departmental application needs updating, the gold image must be updated too. If all desktops were using the same template this potentially involves running full environment wide update for all virtual desktops, even though the number of users running the application may be a fracture of the whole population. Upgrades of bigger number of VMs mean longer maintenance windows, and more IT cycles to complete the activity. All this influences the decision of maintain multiple templates.

Looking at scenario where company XYZ has three big sites with Horizon View deployment. Each site has four types of workers, task workers, power users, mobile users, external users. For each type of worker separate template has to be created. As templates are bound to local pool and cannot be shared across different data centers or physical locations three copies of four template types mentioned above have to be created. It is easy to notice how fast the number of templates can grow especially in big multinational.

3.4.4 Initial Investigation and Setting up Baseline for Improvements

Updating multiple templates is time consuming process and very fast it can turn into full time job for IT administrator. Microsoft patches, software updates, new software requests and customizations can take several hours to complete on a single template.

As an example average times of updating ten Windows 7 templates were measured. All templates were deployed in one of VMware's test environment prepared for the purpose of

this research paper. Updates included running 4 months' worth of all Windows Updates (1.3GB), MS Office 2013 updates, Java, Flash Player. Internet browsers. Times presented below are the actual times from start to finish it took IT administrator to apply all mentioned updates, test newly installed and updates software, and prepare template snapshot for deployment.

Template Name	Time in hours Traditional approach
Template1	7h
Template2	6h
Template3	5.5h
Template4	6h
Template5	6h
Template6	5h
Template7	6h
Template8	6.5h
Template9	6h
Template10	5.5h
Total Time	59.5h

Table 2. Update times using traditional approach

The overall time spent by IT administrator to manually update each of the ten templates took 59.5 hours. That's almost two weeks of one person work to complete a single activity.

In real life scenario time spent on testing new template before releasing into Production has to be considered too. This can be from a few hours to several days, depending if changes can be tested by IT or by users due to the nature of software. E.g. If the software was deployed for finance users, and involves running reports on financial data that has to be verified by end users, and cannot be viewed by other departments like IT due to confidentiality concerns. [50]

3.4.5 Template Maintenance and Update Policies

Depending on the company's IT strategies and security policies end user systems have to be patched and updated with regular intervals. Only this can guarantee maximum level of security and compliance. With VMware Horizon View all configuration settings, patches and updates are pushed to linked clones by the means of recompose or rebalance. This type of activities are normally carefully planned in advance, and a lot of testing is done before releasing new updates into production environment. As keeping both the templates and

linked clones updated is continuous activity normally templates have to be upgraded monthly or quarterly depending on the size of deployed environment, the number, and type of virtual machines. There are some benefits of both.

Monthly updates can follow Microsoft patch Tuesday, and incorporate all MS patches and other software updates released since last upgrade. This guarantees that systems are always patched and up to date. Disadvantage of running updates with such short interval is need for more IT resource. Since all updates and changes first have to be applied to the template and then tested before release to production this process can become a full time job. Especially with big number of different templates.

Quarterly updates are more steady approach. Normally scheduled in the middle of the quarter are compromise between security and IT time constrains especially in big environments with many different golden images in use. This approach allows IT for more time to prepare and test bigger number of templates.

The time IT administrator has to spend annually on maintaining templates can be calculated from formula:

Number of upgrades per Year X Time to Patch single Template X Number of Templates

Example results for quarterly and monthly updates can be achieved by applying data from the tests mentioned in Table 2, where ten templates were upgraded with an average time to update single template of 6 hours.

- Monthly updates require 720 hours per year
- Quarterly updates require 180 hours per year

Last thing that should be considered when scheduling all upgrades and changes is the time it takes to update all VMs. Recompose of single VM takes 10 minutes on average.

Normally 12 VMs can be upgraded at the same time with default settings applied. This theoretically means that with the average of 72VMs recomposed per hour it would take around 14hours to upgrade medium size Horizon View environment consisting 1000 VMs. This can also influence how often upgrades are run.

3.4.6 Traditional Template Update Process and Its Disadvantages

The traditional approach to template update can be described as follows. IT administrator has to log in to each template individually, run all required OS and software updates, patches, drivers, and install new software if required. After installation is complete, initial testing with dummy end user account reflecting user level of privileges is run. After initial testing snapshot of current VM state is taken. At this stage the template is ready for deployment in test environment. After successful testing, the template is ready to be deployed in production with next scheduled upgrade window. All this activities take substantial amount of time. As previously mentioned due to many factors in most View deployments IT administrator has to deal with several golden images that have to be updated regularly.

Traditional approach is not effective. The big disadvantage of this process is lack of automation in maintaining multiple templates, leading to waste of IT time and money by multiplying repetitive tasks, as each golden image has to be worked on individually. This approach is also more prone to error from simple reason. If ten templates have to be updated manually it is ten times more likely something will be missed or configured incorrectly.

Chapter 4: Environment Development – Overall Recommendations

4.1 Multiple Site Support – Cloud Pod architecture

In chapter 3 five issues with current Cloud Pod architecture were identified and analyzed. The main challenge from an administration point of view is the lack of a graphical interface, and the need to use a command line utility to configure all Cloud Pod options. Additionally lack of visibility into global pool entitlements from Pod local management console makes administration harder, leading to problems with fast diagnose of issues with virtual machines. Another big problem is lack of single common interface to manage multiple Pods that belong to the federation, and the need to maintain multiple configurations and golden image sets in each Pod. Lastly a technology problem limiting maximum number of Pods to two different geographical locations exist. To address some of the problems a few improvements were proposed below.

4.1.1 Single Graphical Interface

One of the most beneficial improvement for current design would be development of common graphical interface that will allow to manage all Pods belonging to the federation through single console. Suggested graphical interface should also include both global and local entitlements to provide clarity in user View pool assignment. Developed graphical interface should also provide tools to easily edit any configuration item of the federation, or to add new Pod members.

Currently Pod federation architecture relies on two items:

- Global Data Layer (GDL) – is used by all Horizon View Connection Servers to share key data regarding Pod federation configuration, topology, policies, and user entitlements. LDAP (Lightweight Directory Access Protocol) is used for communication and storing shared data between Connection Servers (CS) in the global data layer.
- View Inter-pod API (VIPA) – is a type of communication CS instances use to launch new desktops, find existing desktops, and share the health status data and other information.

Since GDL relies on LDAP, all policies, and entitlements can theoretically be kept in Active Directory (AD), other configuration items can be implemented through the means of GPOs (Group Policy Objects) also kept in AD. This is already first step in building a common interface for multi-pod design. Additional conditions that would have to be met is that all Pods in the federation would have to belong to the same domain or have two way trust relationship established with one another. This will allow exchange of information between Pods on AD level.

VIPA is responsible for remaining items like desktop management, including session management, new desktops preparation, and desktop health status. Normally inter-pod communication is maintained by single connection server per Pod. This is because the communication between the connection servers and VMs is very sensitive to network latency. Small latency or connection loss between connection servers can cause corruption in the database. What VIPA allows is exchanging of information between pods over the WAN, and can help with WAN latency reaching up to 500ms. In case of time out of inter-pod requests, VIPA will automatically try to connect to different CS in the remote Pod.

In single Pod design all desktop and session information is kept in ADAM (Active Directory Application Mode) database unique to each Pod. ADAM database is very sensitive to any network latency. To implement single management console it may be required to create common inter-pod ADAM database holding all information for all Pods in the federation. Looking at the problem from different perspective, there might be no need for single inter-pod database. Instead if functionality allowing reading multiple ADAM databases through single interface was implemented, there would be no need for single database. Additionally if further improvements of latency tolerance for VIPA will be implemented, plus right Quality of Service (QoS) of WAN link can be guaranteed, it can theoretically be possible to overcome all the issues with developing single fully functional management interface.

4.1.2 Unified Configuration Items Management

Another beneficial improvement that can be suggested is implementing a solution that will allow maintaining centralized set of configurations and golden images across all Pods from the same federation without maintaining several versions of these items.

For centralizing the configuration items use of LDAP and Active Directory (AD) services can be beneficial. Tying Pod configuration items into Active Directory, and use of Microsoft Administrative Templates, and Group Policy Objects commonly called GPOs can allow mass edition of various components of developed objects. If such configuration templates existed it would be just a matter to provide Pod - specific information, and expected behavior with options to choose from would be already defined in the template itself. Similarly unified set of AD security groups and permissions could be set across the board to provide access to both configuration items and virtual machines. Active Directory can theoretically help centralizing some of above mentioned tasks and items and lessen burden with configuration inconsistency. There are more things to consider here, like level of trust relationship between Pods, if different domains exist, but they are outside the scope of this paper.

4.1.3 Improvement of Template Management

Maintaining multiple golden image copies across Pods is another issue. The problem with golden images comparing to configuration items or permissions is their size. Typical golden image can be between 15GB – 30GB in size which limits the number of available options. Due to latency issues golden image used for linked-clones deployment has to reside in the same physical location as the clones. A few solutions can be suggested, some of them are very simple, and improve level of consistency only. Some are more advanced, and can address the problem by automation and image centralization. Proposed solutions are presented below.

- Use of centralized change log or tracker for each golden image in the form of documentation – even though doesn't accelerate golden image preparation it can help with keeping consistency between multiple versions of the same image in different locations when used as reference guide.
- Use of central repository for all golden images – all images are customized and uploaded to repository by IT administrator, from there they can be downloaded to each Pod and deployed accordingly. This can guarantee use of exactly the same image across all Pods.
- Use of other product that can help with golden image configuration consistency in multiple locations – for this purpose solution presented in “**Template management** –

improvement proposal” chapter can be used. As proven in this paper VMware Mirage can be used for multiple template management. It can significantly simplify management tasks and deployment times thanks to improved automation.

4.1.4 Overcoming Other Restrictions

The limitation to two sites or four pods are mainly limited by currently available technologies. With adding every Pod to federation the size of used databases is growing. Bigger database means more time might be required to read items from the DB, longer seek times mean more database processing latency, and as mentioned before both global data layer and ADAM database are very sensitive to any latency.

Network latency is also the culprit for the number of sites that can join the federation. Each Pod in the federation continuously communicates with other Pods. Connection servers in each Pod stay in synch at all times. With adding next Pod the number of inter-pod communication grows exponentially as each Pod has to stay in synch with other federation members. Very often WAN link speed cannot be guaranteed and implementing dedicated WAN link with required level of quality of service (QoS) might be too expensive.

Since above problems are purely technological limitations the chances are that with continuously increasing speed of Internet links higher QoS can be guaranteed and implantation of more flash storage will reduce the database access latencies allowing faster access and more network traffic throughput required to join more Pods together.

4.2 Management Console Unification

In chapter 3 detailed analysis of administration challenges was provided. A few issues with missing functionality and implementation of existing features were identified. Lack of sufficient information regarding IP and hardware configuration of the virtual machine, and lack of integrated remote console force IT administrator to switch between several different interfaces to obtain required information. IT administrator is forced to use third party tools to be able to monitor or interact with current user sessions. Additionally potential issues caused by virtual desktop entitlements inconsistencies, and poor granularity of permissions in management console add to the whole picture. All these issues make VDI infrastructure more difficult and time consuming to manage.

To address some of the current limitations certain improvements and changes can be suggested. Improvements should bring more clarity, and simplicity to virtual desktop management process, reducing time required for completing maintenance or troubleshooting tasks.

4.2.1 Virtual Machine Configuration Information Availability

As mentioned before currently Horizon View Administrator console doesn't provide any information regarding VM hardware or IP configuration. To address the issue a few items can be suggested.

IP address itself isn't a feature or characteristic of a virtual machine as many can mistakenly think. It is additional component dictated by DHCP server allowing Layer 3 network communication. IP address assignment to a VM is completely independent of vCenter or Horizon View components and can change between restarts depending on the status of IP address lease. Because none of the VMware components have any influence on IP address selection, the only way to read the current IP is from VM operating system itself. For this purpose additional APIs are installed on Windows OS as part of VMware Tools. VMware Tools is a software package that provides better integration of Windows VMs with virtual environment and improves user experience. Since the API is already installed and used for the purpose of vCenter Server it could potentially be shared with Horizon View Agent responsible for communication with View Administration console. Feeding IP configuration to both VMware Tools and Horizon View Agent would fix the need of IT administrator switching between vCenter and Horizon View management console to obtain IP information.

Virtual machine hardware configuration information, like amount of CPU, RAM or MAC address, are currently only available through vCenter Server. All this information is contained in single file with extension .VMX residing with the rest of virtual machine files located on shared storage. To address the problem, some changes could be made to allow reading the .VMX file by both vCenter and one of Horizon View components. This should allow to stream the information back to management console. Theoretically that should be possible to implement as Horizon View Composer oversees all linked-clones deployment process on shared storage and works closely with vCenter on every linked-clone operation.

Obtaining this functionality would be really helpful in performing IT administrator daily tasks.

4.2.2 Remote Console Integration

Currently Horizon View has no MKS (mouse, keyboard, and screen) console available for virtual machine access. Every time interaction with user OS configuration is required, IT administrator needs to switch to vSphere client or some other third party software like VNC viewer. Integrating remote console as part of Horizon View management console would help with minimizing number of interfaces required for virtual environment support.

One of possible solutions to address the issue would be redesigning Horizon View manager, and implement already available functionality from vSphere web client. This may be very challenging as vSphere web client relies on HTML5. VCenter server uses open source Tomcat web server to allow web client HTML5 remote console access and passing MKS functionality. HTML5 is the latest web design technology that uses the code directly to generate interactive content. This means that the elements are not pre-made and stored in an exact form, but have just characteristics defined in the code, and web browser is responsible for rendering of the actual content when the page is loaded.

Current design of View management console relies on Flash technology only. Flash is binary format for multimedia content that uses object oriented scripting to compile applications into SWF files that can be run across many web browsers and platforms. What that means is that the displayed web page content is premade and designed fully beforehand, which is fundamentally different from HTML5 approach. Since both technologies are so different the easiest solution could be redesigning Horizon View console in different language like HTML5 and use available vSphere functionality to display the VM console or at least add some components allowing HTML5 remote console on current platform. Use of HTML5 could potentially address other issues caused by Flash technology itself, like slow content load and refresh times in low bandwidth networks.

4.2.3 Administration Challenges – Conclusions

Improvements suggested above address some of the biggest problems stopping unification of multiple management consoles. They would potentially reduce troubleshooting times, and add simplicity to current management processes.

By addressing remaining items like missing power options, improvement of virtual machine assignment scheme, or better administrator console granularity simpler issues could be addressed by first line of support without escalation needs, reducing total time or resolution to minimum. All this is theoretically possible with better integration with VMware Tools APIs and Active Directory.

Implementing all improvements should bring more clarity to virtual desktop management process. Possibly the consoles could be used interchangeably by choice, as all tools and functionality would be available in single console, the need of using multiple interfaces would be no more required. All VM related tasks could be performed from single console.

4.3 Template Management – POC Improvement Proposal

In previous chapter a few issues with traditional template management approach were identified. Lack of automation in keeping multiple View templates updated, and significant amount of time required to customize and update single golden image are the main problems. This state of things can very quickly cause need for significant IT resources just to keep environment updated and compliant with security and software refresh policies. To address some of the issues new approaches to template management should be explored. This chapter concentrates on presenting valid improvement solution and analyzing potential challenges with its implementation. Additionally design and implementation decisions are going to be explained.

4.3.1 Design Choices Explained

At the beginning of the design stage decision has to be made on the best approach allowing maximizing the benefits. There are many ways that could potentially improve the current design. Simplest and fastest solution would be attempt to decrease the number of golden images to minimum. This can be achieved in two ways.

First method is incorporating all software packages into single template. When this scenario is considered balance between time saved to maintain multiple templates and extra money spent on licenses for all systems have to be evaluated. It may be that the cost of extra licenses will be much greater than the money saved on IT cycles. Some of license costs can potentially be reduced by using open source and free software. But there are

some other constraints like longer maintenance windows, caused by the fact that all systems are upgraded when any change to template components have been made. This approach can also have influence on increased storage usage, as additional applications installed on all VMs require more space.

Second method to minimize total number of templates is matching the hardware specification of highest performance VM in the environment, and applying those settings to all VMs. This way all templates share the same hardware configuration, meaning have the same amount of CPU and RAM assigned. This approach may be feasible in small deployment, with minimal resource assignment differences between templates. In big dynamic environment where many templates with different virtual hardware specification exist this could potentially lead to big waste of resources by assigning them to systems that don't need the capacity in the first place.

Both proposals mentioned above are valid and can bring a lot of benefit to small virtual environment that configuration changes very little over time. But when considering bigger deployments they are not feasible options and the benefit from their implementation might be minimal.

To overcome the issue for any size deployment, different approach should be considered. Instead of concentrating on decreasing number of templates, that will eventually grow anyway, template update process has to be automated, and amount of repetitive upgrade tasks run manually has to be minimized.

After analyzing all pros and cons, decision has been made, that VMware Mirage can potentially be used to address the issue, if implemented correctly.

Mirage is a piece of software that allows OS and applications layering. It was designed to mainstream images to physical PCs. Thanks to its design it allows keeping single copy of base layer used for provisioning on multiple physical system with different hardware and software configuration. Even though it doesn't support linked clone virtual machine deployments, it can potentially be beneficial to some extent. The analysis of implementation guidelines, proposed design and achieved benefits are going to be presented in further sections of this chapter.

4.3.2 Improvement Proposal

Mirage capabilities to upgrade or reimage physical systems can potentially be applied in similar manner to virtual environment to help with automating template maintenance activities. Looking at the problem, and what Mirage can do, high level improvement process was illustrated in Figure 11.

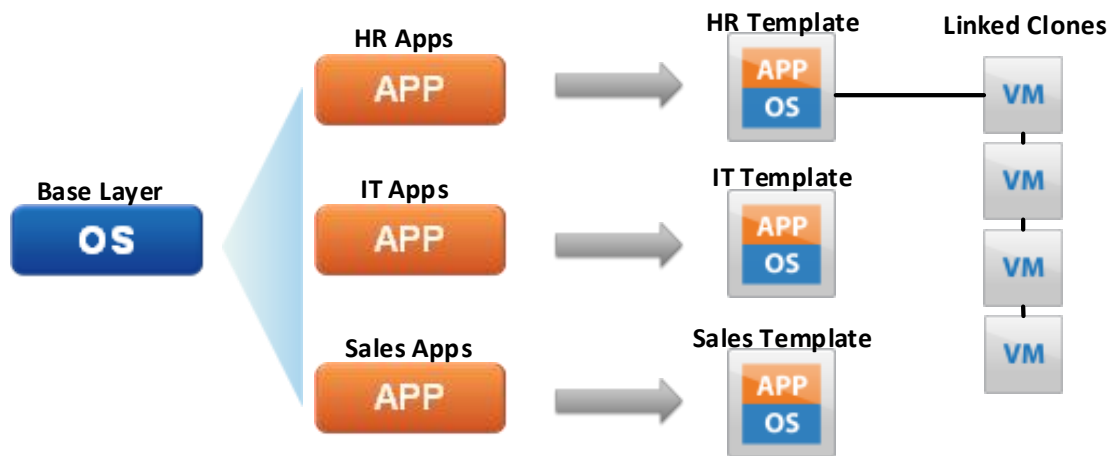


Figure 11. Template update process using VMware Mirage

Administrator updates single base layer, and using Mirage uploads its CVD to data center. After that, the base layer CVD can be streamed to endless number of Horizon View golden images with mirage client installed. If a need for department of specific applications exist, they can be captured separately in application layer and deployed with base layer to specific View templates. After all templates are updated snapshots are taken. These snapshots are further used for linked clone pools deployment.

Using above approach means, that the OS with all its patches and updates, same as core applications, and additional departmental applications have to be updated only one time and captured in layer container. After that, selected layer CVD can be distributed endless amount of times to all View templates accordingly. This is potentially big save on time and IT resources.

4.3.3 Design Challenges

Proposed design as beneficial as it sounds brings a few challenges. First of all Mirage isn't supported on View deployments with linked clone technology. Second, lack of support for linked clones also means that Mirage functionality was never tested in scenario proposed.

Because of issues mentioned above, some problems with increased network use, and storage overload, can potentially happen. This part of research project will concentrate on analyzing and overcoming all known and potential issues to allow use of Mirage for template management automation. The proof of concept (POC) test environment is going to be built and used to run all template automation scenarios, and to provide real life data also analyzed in further part of the project.

In physical world Mirage is end to end solution that maintains end user PCs. This cannot be done the same in virtual environment. For successful implementation in virtual environment there has to be separation between Mirage being used for View templates and templates being used for linked clone deployment. The reason for that is as follows. When Mirage manages endpoint system all its data is backed up in the data center. Continuous data uploads between endpoint and server exist. Additionally base layer uploads or backup restores are pushed from data center directly to endpoint. This creates a few issues that are related to the fact of dealing with virtual not physical environment. The architecture of linked clone deployment simply will not allow for efficient endpoint management using Mirage. Main reasons for that are:

- Increased network bandwidth usage caused by continuous bidirectional communication and data transfer between endpoint and Mirage server in the data center
- Doubling of VM backups – normally backup solution is part of Horizon View deployment plan, using Mirage would create unnecessary overhead and could potentially create double copies of VM data backups
- Increased storage requirements on Mirage servers – sufficient storage required to backup thousands for Linked Clone VMs
- Increased storage usage on the datastores hosting virtual machines files – Initial size of Linked Clone OS disk is only 4KB in size, and grows only with user changes that differentiate it from replica it is linked to. If new Base Layer is applied to linked clone by the means of Mirage, View Composer may detect newly applied Base Layer as all new blocks of data different to replica feeding the clone. Those changed blocks will be treated as user initiated and added to Delta disk causing OS disk to inflate quickly. As a comparison if operation to update linked clone OS disk was a View composer

initiated task, the OS disk size would be equal to minimum disk size as both VM disk and replica feeding the VM are identical.

To avoid all of those issues Mirage can only be used up to some extent. The details of how to take advantage of Mirage capabilities and avoid potential issues are going to be explained in detail in further section. [30]

4.3.4 Overcoming Design Challenges

Because Mirage wasn't designed to work with linked clones virtual machines some adjustments have to be made. Not implementing suggested changes can result in storage speed degradation, increased VM datastore use, and increased network bandwidth usage. Explanation of reasons for all above can be found in design challenges section.

To avoid all mentioned problems Mirage interaction with virtual environment has to be restricted to template level only, none of the linked clones can be managed by Mirage.

Mirage relies solely on agent service running on endpoint. The easiest way to avoid the problem of Mirage interaction with linked clones is either removing or disabling Mirage agent from all clones. This can be done by disabling "Wanova Mirage Desktop Service" responsible for running Mirage agent, on all View templates after Base Layer has been successfully deployed, and just before taking the snapshot used for linked-clones deployment. This way the required CVD changes are applied to templates and the snapshot used for linked-clones deployment has Mirage agent disabled.

There are a few ways to disable Mirage agent:

- From services.msc Stop and Disable "Wanova Mirage Desktop Service"
- From command line locally by running below commands

```
sc config "Wanova Mirage Desktop Service" start= disabled  
sc stop "Wanova Mirage Desktop Service"
```

- From command line remotely by running below command:

```
sc \\<Template_IP_Address> stop "Wanova Mirage Desktop Service"
```

4.3.5 Test Environment Details

To make the POC as realistic as possible one of VMware internal test environments was used to run whole experiment and testing. Details of existing environment prerequisites to run the POC:

- VMware vCenter Server 5.5
- VMware ESXi 5.5 Update2
- VMware Horizon View 6.0
- VMware Mirage 5.2

All components mentioned above were already deployed in test environment, only changes in configuration and resource pools layout had to be done to suit the POC needs, and help with presentation.

The specification of underlying ESXi hosts hardware and some of the naming patterns used in Virtual environment are omitted due to confidentiality and lack of relevance to presented research.

Permission from VMware management was given to deploy number of test golden images in the test environment. Resource pool named: “ViewTemplate-POC” was created especially for the purpose of the experiment. Figure 12 presents, vCenter view of all templates used in the POC

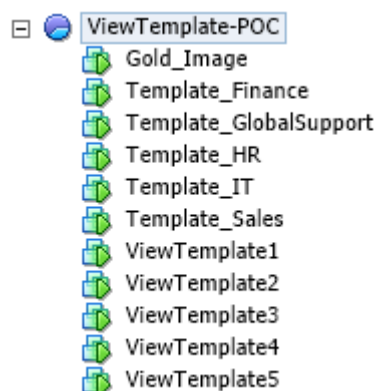


Figure 12. List of testing templates in vCenter Server inventory

Eleven templates were prepared to run multiple tests and scenarios. Primary template used for Mirage reference CVD creation is presented at the top of the resource pool, and labeled as “Gold_Image”. Next five department specific templates for Finance, Global Support,

HR, IT, and Sales were prepared. Lastly five general View templates named as “ViewTemplate1” through “ViewTemplate5” were deployed. General templates have various amount of assigned resources like CPU and RAM to make testing more realistic.

4.3.6 Measuring and Execution Time

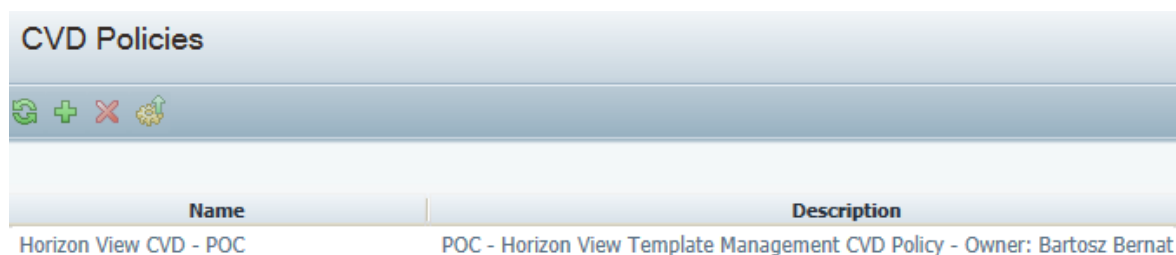
To assess the experiment correctly some baselines had to be setup in advance. As a base line for experiment IT administrator time involved in updating 10 View templates using traditional template management approach is used. All results were measured and recorded. The outcomes of the POC experiment are going to be check against mentioned benchmarks. Traditional approach process detail and achieved results can be found in chapter “**Initial investigation and setting up baseline for improvements**”. Both baseline tests and POC tests were run in the same test environment with the same underlying hardware, storage and network to avoid potential distortions in measurements.

Successful experiment should prove significant time save of IT professional cycles and present increased level of automation comparing to current process.

4.3.7 Implementation Details

With all eleven templates prepared, implementation phase can start. There are a few steps involved.

All Mirage management tasks are policy driven. That is why first step is to create CVD policy defining which Virtual Machine files should be backed up in the data center, and which files should be excluded. CVD policy needs to be setup only once at the beginning and can be continuously used for future backups and updates. For the purpose of this project CVD policy called “Horizon View CVD - POC” policy was created.



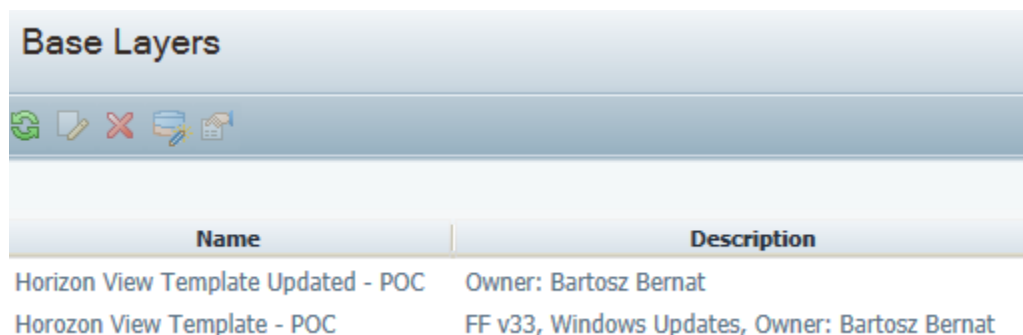
CVD Policies	
Name	Description
Horizon View CVD - POC	POC - Horizon View Template Management CVD Policy - Owner: Bartosz Bernat

Figure 13. CVD policy details in Mirage console

According to the CVD policy, Mirage server will back up all system files residing on C: drive including drivers and installed standard set of corporate applications. All temporary files, swap files, and page files are excluded from backup.

Mirage communication between datacenter and endpoints relies on the agent running on the endpoint system. This is why second step is to install Mirage agent on all eleven templates and make sure they all communicate with the server.

After communication with Mirage server is established creation of reference CVD can start. “Gold_Image” VM is going to be used to generate the reference CVD according to rules defined in “Horizon View CVD - POC” policy. Last step to complete this part of experiment is converting reference CVD into Base Layer. This part is initiated by administrator and done automatically by Mirage server.



Name	Description
Horizon View Template Updated - POC	Owner: Bartosz Bernat
Horizon View Template - POC	FF v33, Windows Updates, Owner: Bartosz Bernat

Figure 14. VMware Mirage - Base Layers view

Once Base Layers are ready they can be provisioned to other systems with Mirage agent installed. In this implementation Base Layer created from “Gold_Image” VM is going to be deployed to ten View templates as presented in Figure 15.

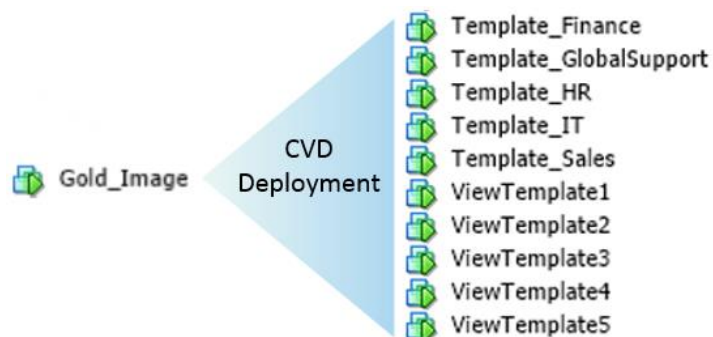


Figure 15. Base Layer deployment scheme

What this effectively means is that only single golden image has to be manually updated. After that, any other template can instead be updated automatically using Base Layer provisioning, previously captured by Mirage. The whole process can be described by Figure 16

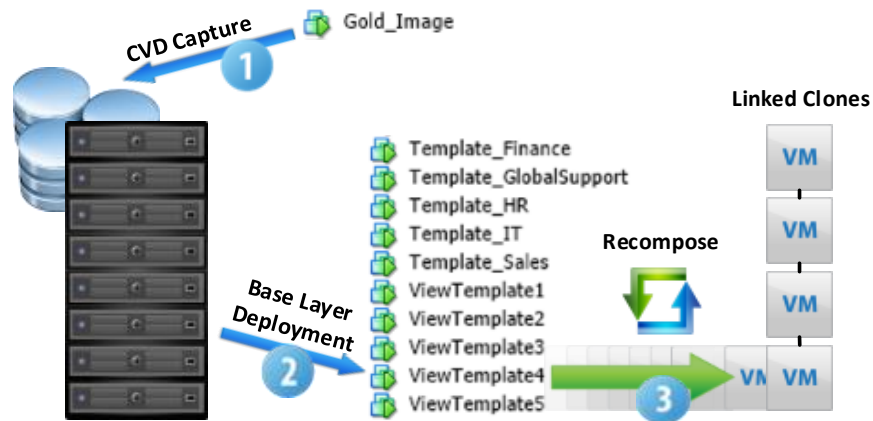


Figure 16. Base Layer deployment process

Once all templates have required layers deployed, Mirage agent can be disabled using either command line or “services.msc” Windows Snap-in as previously mentioned in “Overcoming design challenge” section. Last step is taking View template snapshot, and running recompose task on some of the linked clones deployed in selected View pool. Recomposing linked clones will apply all changes incorporated in the template snapshot. After completed recompose all affected linked-clones run exactly the same image as the one captured from “Gold_Image” in Mirage CVD.

If further updates or changes are required, IT administrator has to follow the same steps explained above. In short steps involved are:

- Update “Gold_Image” used for CVD creation
- Capture updated CVD from “Gold_Image”
- Upload Base Layer to all required View Templates
- Use View templates to recompose linked-clones to apply updates

4.3.8 Experiment Results

To provide reliable data, and be able to present trends and level of consistency between update times, eleven templates were used. Following all guidelines, and steps outlined in design section allowed to successfully run the template updates and gather required data. For each template time spent by IT administrator working on one View template at the time was measured.

Assuming that the template is included in quarterly update schedule, and minimum four months' worth of updates need to be applied. This includes OS, software, registry, drivers, and services updates, customizations, enabling and disabling Mirage services and taking the snapshot for linked-clones deployment.

Template Name	Time Mirage
Gold_Image	7h
Template_Finance	1h
Template_GlobalSupport	0.5h
Template_HR	0.5h
Template_IT	0.5h
Template_Sales	0.5h
ViewTemplate1	0.5h
ViewTemplate2	0.25h
ViewTemplate3	0.5h
ViewTemplate4	0.25h
ViewTemplate5	0.25h

Table 3. Template update times using Mirage

All results are presented in the Table 3. It can be noted that most of the measured times are fairly similar, with one exception. “Gold_Image” took substantially more time to update. This is because it is the only template that has to be updated manually from start to finish by IT administrator.

4.3.9 Discussion Surrounding Implementation Results

Now that the implementation phase is complete it's time to assess the results. As mentioned in chapter 3 the main benchmark for the experiment is the total time IT administrator spent on maintaining ten Horizon View templates. These results are going to be compared to results from the POC implementation, outlined in chapter 4, detailing times required to upgrade eleven Horizon View templates using proposed improved process.

The reason for eleven templates used in POC is as follows. The first out of eleven templates named "Gold_Image" is only used to update remaining ten View templates and not for linked-clones deployment. It's the additional overhead that has to be included to assure clarity of the whole design. Since this is one of the elements that IT administrator needs to update it has to be included in the overall results.

Time it took to update both baseline templates and POC implementation templates are presented again side by side in Chart 4.

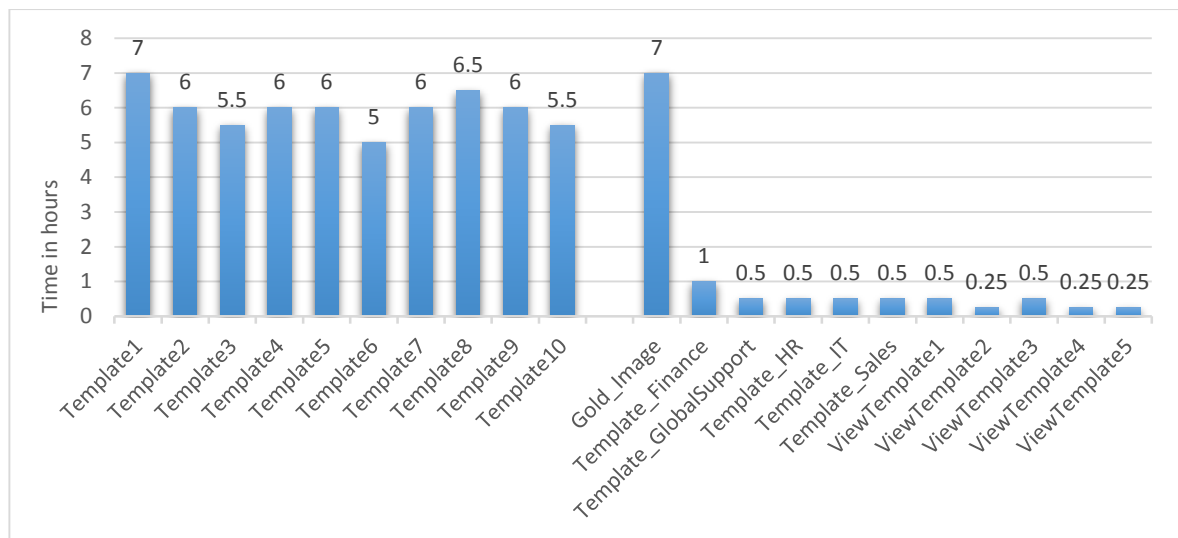


Chart 3. Traditional approach vs. Mirage update time

It is easy to notice a big difference between both solutions. Using traditional approach to upgrade all templates took 59.5hours, comparing to suggested improved process that took only 12.5hours of IT administrator time to complete.

Currently used design – which for the purpose of this paper was also called traditional approach, times are much higher, but fairly consistent. Results are depend purely on the total number of updates and changes that need to be applied per template. The average

update time is 6h per template. From the gathered data the trend can also be assumed. The manual process of updating each template will take not less than 6 hours of IT administrator time to complete.

Things look a bit different for POC implementation. There is only single spike in the time it took to update “Gold_Image” template. Remaining templates have considerably shorter times recorded. The average time to update both “Gold_Image” used for CVD deployment and ten templates used for linked-clones deployment is 1.2h. The trend for POC implementation is dictated by Horizon Mirage characteristic, that single CVD can be applied to endless number of systems. After manual labor on “Gold_Image” is finished, rest of the View templates are updated automatically. The more View templates are being updated by applying Mirage CVD the less time in total IT administrator has to spend on this activity comparing to traditional approach. It can be said that after the gold image is captured in CVD, IT administrator will not have to spend more than 1 hour of his time to complete all tasks per template. That is a big save, that even on as small population as ten View templates can be easily noticeable.

All above proves the advantage of using automation of template update activities over manual process. Thanks to the use of suggested solution, only minimal IT administrator interaction is required. The full benefit of using suggested solution can be even more visible when looking at the annual IT costs. The more updates involving long and time consuming customizations have to be implemented, the more potential savings comparing to traditional template update approach can be achieved.

Chapter 5: Conclusions

Desktop virtualization solutions gained much interest in recent years. Today VDI technologies become inseparable part of IT strategies of most medium and large enterprises. Thanks to many technology benefits, including infrastructure and energy cost saving, high availability, improved security and disaster recovery, access flexibility and central management more user workloads are shifted to virtual environment.

5.1 Final Conclusions

The first objective of the research was an analysis of current state of desktop virtualization solution provided by VMware, identify the gaps, bottlenecks, and areas for improvement. This has been realized by first introducing the technology in literature review chapter. After that, bottlenecks identification, and detailed problem analysis was done in chapter 3. Analysis in chapter 3 showed that VDI technology has matured significantly in recent years, and many enterprises are confident in its implementation. It also showed that the main market drivers towards implementation are increased reliability, ease of management, security, disaster recovery, and mobility support from any device. These are also the items that set the direction for the technology development. The main challenges stopping some companies from implementing VDI are lack of funding, limited familiarity with VDI, lack of trained personnel to support it, and in some cases resistance from users.

Further in chapter 3 both current, and already addressed technology challenges were identified. Big improvement in the storage speed, development of GPU virtualization solutions and integrating antivirus solution as part of virtual environment design dramatically improved performance of VMware Horizon View deployments today.

The outstanding problem that still needs addressing is multiple site support and its limitations. Lack of graphical interface allowing Pod configuration, global permissions not reflected in administrator console and only available through command line utility, and limited information from management console level regarding other Pods belonging to the federation are only some problems that were identified. Addressing current design problems and limitations would greatly simplify multiple sites support and administration by adding clarity and shortening time required for configuration and troubleshooting. Even

with all current problems and limitations Cloud Pod architecture is worth exploring and has big potential to bring value to the enterprises.

Administration challenges were also addressed in the analysis. Multiple management interfaces, lack of functionality and sufficient VM information, same as poor management console permissions granularity, and Horizon View pool entitlement scheme can cause many administration challenges. They force IT administrator to use many tools and implement workaround-like processes to perform some of the management tasks.

Lastly the challenges related with Horizon View template management were analyzed. The research shows that current process is not effective. IT administrator has to perform each template update manually. By doing so the same long update tasks have to be repeated multiple times. This means big waste of IT time, and is more prone to error causing configuration inconsistencies between templates. To address the issue a few solutions were proposed. Some improvements can be made by introducing update tracker, or decreasing number of templates to minimum. They all address only some issues. The best results can be achieved by introducing automated template update process. For the purpose of this research VMware Mirage was suggested as solution that can help minimize IT administrator time and efforts to minimum by fully automating template update.

Second objective of the research is to provide overall set of recommendations that can help building system with improved automation. Chapter 4 outlines list of recommended improvements that can potentially address current Horizon View issues defined in chapter 3. Research shows that automation level improvement can be achieved on a few different levels. First by minimizing the need for multiple management consoles use, by implementing missing functionality like integrated remote console, virtual machine IP and hardware configuration information, and power options. If all tools and functionality were available in single console, the need of using multiple interfaces would be no more required. All VM related tasks could be performed from single console. This design would minimize IT support times to minimum and add clarity to whole management process.

Additional automation level improvements could be achieved by taking advantage of latest technology improvements like VSAN or vShield. This could simplify the infrastructure

management process as less time would be spend on complicated storage management or antivirus scans on OS level.

Lastly implementing improvements to current Cloud Pod architecture should improve automation of management and administration tasks, that could be performed faster and in easier manner. Developing graphical interface fully supporting multi-Pod design management, global permissions visibility in administrator console, and increased number of Pods per federation could potentially reduce troubleshooting times, and add simplicity to current administration processes.

Third objective of the research is the implementation of “proof of concept” virtual machine template management automation solution based on recommendations made in chapter 4. The research proves, that by overcoming some of the technology and design restrictions of VMware Mirage, which currently doesn’t support linked clone technology, improved automation level for multiple images maintenance and update process can be achieved. The implementation results show real life benefits of proposed recommendations, allowing four times faster multiple templates update. The implementation also shows that the manual update process has to be performed only once minimizing chances of configuration inconsistencies between multiple templates to zero. Because all Horizon View templates used for linked clones deployment are based on the same CVD the need of testing templates before implementing to Production environment is reduced to minimum allowing additional save of IT cycles. Lastly successful implementation of POC helped testing new use cases for VMware Mirage that until now hasn’t been proven to work in linked clone deployments.

5.2 Future Work

For the future a more in depth analysis of protocols used for remote access and inter-Pod communication could be performed. Additionally potential benefits of using floating desktops, challenges related to the use of roaming profiles, and user Persona migration bottlenecks could be analyzed. Related work could analyze both PCoIP and Persona logs to identify potential issues, same as different file hosting solutions required for building user profile repository. Lastly interoperability of file sharing solution with Active Directory and challenges of use profile migrations between file systems could be explored.

References

- [1] M. Portnoy, *Virtualization Essentials*, Sybex, 2012.
- [2] M. Rosenblum and T. Garfinkel, "Virtual machine monitors: current technology and future trends," *Computer*, vol. 38, no. 5, pp. 39 - 47, 16 May 2005.
- [3] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," in *Proceedings of the Fourth Symposium on Operating System*, New York, 1973.
- [4] K. Adams and O. Agesen, "A comparison of software and hardware techniques for x86 virtualization," in *ASPLOS XII Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, New York, 2006.
- [5] E. Mollick , "Establishing Moore's Law," *Annals of the History of Computing*, vol. 28, no. 3, pp. 62-75, 21 August 2006.
- [6] R. Fitzhugh, *vSphere Virtual Machine Management*, Packt Publishing, 2014.
- [7] A. Suzuki, M. S. Johnson, J. Gillette and D. N. Jeppesen, "System and method for deploying a virtual machine". USA Patent US 20070234302 A1, 4 October 2007.
- [8] K. Jin and E. L. Miller, "The effectiveness of deduplication on virtual machine disk images," in *Proceeding of SYSTOR '09 Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, New York, 2009.
- [9] S. B. Vaghani, *Virtual machine file system*, vol. 44, ACM, Ed., New York: Newsletter, 2010, pp. 57-70.
- [10] C.-M. Tu , S.-H. Ku, J.-C. Tseng, H.-T. Kao, F.-S. Lu and F. Lai, "CHT cloud orchestration: An integrated cloud system of virtualization platform," in *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, Hsinchu, 2014.
- [11] J. Smith and R. Nair, *Virtual Machines: Versatile Platforms for Systems and Processes*, San Francisco: Morgan Kaufmann Publishers Inc., 2005.
- [12] S. Lee, "Movable virtual machine image". USA Patent US20100107163 A1, 29 April 2010.
- [13] E. N. Bayeh, "Virtual machine pooling". USA Patent US6223202 B1, 24 April 2001.
- [14] J. Che , Q. Gao and D. Huang, "Performance Measuring and Comparing of Virtual Machine Monitors," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008. EUC '08.*, Shanghai, 2008.

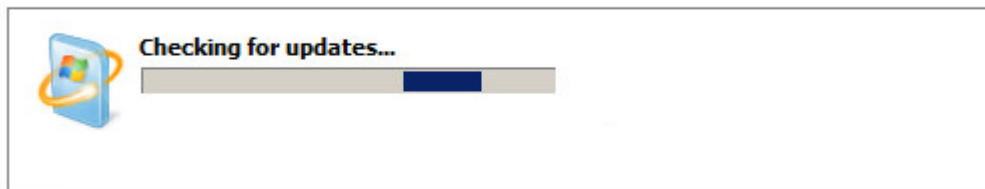
- [15] W. Huang, J. Liu, B. Abali and D. K. Panda, "A case for high performance computing with virtual machines," in *Proceedings of the 20th annual international conference on Supercomputing*, New York, 2006.
- [16] G. Tong , H. Jin, X. Xie, W. Cao and P. Yuan, "Measuring and Analyzing CPU Overhead of Virtualization System," in *IEEE Asia-Pacific Services Computing Conference (APSCC)*, Jeju Island, 2011.
- [17] B. Kleyman , "Hypervisor 101: Understanding the Virtualization Market," 2012. [Online]. Available: <http://www.datacenterknowledge.com>.
- [18] A. Elsayed and N. Abdelbaki, "Performance evaluation and comparision of the top market virtualization hypervisors," in *2013 8th International Conference on Computer Engineering & Systems (ICCES)*, Cairo, 2013.
- [19] P. v. Oven, VMware Horizon View Essentials, Packt Publishing, 2014.
- [20] K. Wang, J. Rao and C.-Z. Xu, "Rethink the virtual machine template," in *Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, New York, 2011.
- [21] K. R. Jayaram, C. Peng, Z. Zhang, M. Kim, C. Han and H. Lei, "An empirical analysis of similarity in virtual machine images," in *Proceedings of the Middleware 2011 Industry Track Workshop*, New York, 2011.
- [22] J. Kirschnick, J. Alcaraz Calero, L. Wilcock and N. Edwards, "Toward an architecture for the automated provisioning of cloud services," *Communications Magazine*, vol. 48, no. 12, pp. 124-131, 2010.
- [23] I. Melinte, A. Bujor, R. Dobre and A. Herisanu, "Running KVM Virutal Machines in Batch Systems," in *2013 19th International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, 2013.
- [24] S. Wu, J. Yuan, W. Gao, X. Shi and H. Jin, "Effectively Deploying Virtual Machines on Cluster," in *Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE*, Yilan, 2008.
- [25] VMware Inc., "Documentation for VMware Horizon with View," 2015. [Online]. Available: https://www.vmware.com/support/pubs/view_pubs.html.
- [26] J. Ventresco, VMware Horizon View 6 Desktop Virtualization Cookbook, Packt Publishing, 2014.
- [27] J. Langone and A. Leibovici, VMware View 5 Desktop Virtualization Solutions, Packt Publishing, 2012.
- [28] P. von Oven, VMware Horizon Mirage Essentials, Packt Publishing, 2013.
- [29] J. Domenichini and J. Wu, "VMware Mirage 5.0 Reviewer's Guide," 2014.

- [30] VMware Inc, "VMware Mirage 5.2 Documentation Center," 9 December 2014. [Online]. Available: https://www.vmware.com/support/pubs/mirage_pubs.html.
- [31] Salek, Bryan ; McLeod-Warrick , James ;, "VMware Emerging Strategies for Managing Mobility," 2013.
- [32] V. Viarengo, "Mobility Journey," 17 March 2011. [Online]. Available: <http://mobilityjourney.com>.
- [33] C. Burry, "VMware SMB Blog," VMware, 18 October 2012. [Online]. Available: <http://blogs.vmware.com/smb>.
- [34] J. Olesen and F. Kristensen, "VDI on IBM Flex Systems," 23 January 2013. [Online]. Available: <http://www.slideshare.net/>.
- [35] C. Wolf, "Desktop Virtualization Trends at Gartner Data Center," Gartner, 10 December 2012. [Online]. Available: <http://blogs.gartner.com/chris-wolf/2012/12/10/desktop-virtualization-trends-at-gartner-data-center/>.
- [36] Federica , Troni ; Gartner, "Best Practices to Drive Cost and Value Optimization in End-User Computing," Gartner, 2014.
- [37] PR Newswire, "Global Virtual Desktop Infrastructure (VDI) Market 2014-2018," PR Newswire, New York , 2013.
- [38] B. Madden, "The state of the VDI industry in 2014," 2014. [Online]. Available: <http://www.computerweekly.com>.
- [39] Infiniti Research Ltd. , "Global VDI Market 2015-2019," Infiniti Research Ltd. , 2014.
- [40] Tech Target, "Money making opportunities in virtualization," Tech Target, 2013.
- [41] VMware Inc., "VMware Success Stories - Baltimore City Public Schools," [Online]. Available: <http://www.vmware.com/a/customers/product/3/VMware+View>.
- [42] G. Berger, "VMware has some big news for EUC," 9 April 2014. [Online]. Available: <http://blogs.gartner.com>.
- [43] G. Deka, "Cost-Benefit Analysis of Datacenter Consolidation Using Virtualization," *IT Professional* , vol. 16, no. 6, pp. 54-62, 24 November 2014.
- [44] M. A. Margevicius and N. Hill , "Is the Hosted Virtual Desktop Market Struggling to Grow?," Gartner, 2014.
- [45] C. LeThanhMan and Kayashima, M., "Desktop workload characteristics and their utility in optimizing virtual machine placement in cloud," Hangzhou, 2012.

- [46] L. Cheng, "VMware Horizon 6 – Driving Costs Down with VMware Virtual SAN Integration," 9 April 2014. [Online]. Available: <http://blogs.vmware.com>.
- [47] KangWoo Hong, Il-Koo Jung, Won Ryu and Jun Kyun Choi, "A study on GPU virtualization in a virtualized server environment," Busan, 2014.
- [48] R. Spruijt , "3D graphics for Virtual Desktops Smackdown: AMD, Amazon, Citrix, Intel, HP, Microsoft, Mainframe2, NICE, NVIDIA, OTOY, Teradici, VMware++," 8 April 2014. [Online]. Available: <http://www.brianmadden.com>.
- [49] S. Huisman, "Horizon View 6 – Cloud Pod Architecture," 19 June 2014. [Online]. Available: <http://virtualfuture.info/2014/06/horizon-view-6-cloud-pod-architecture/>.
- [50] F. Wang , Y. Liu, B. Lei and J. Li, "Benchmark Driven Virtual Desktop Planning: A Case Study from Telecom Operator," Shanghai, 2012.
- [51] B. Madden, G. Knuth and J. Madden, The New VDI Reality, 2nd ed., San Francisco, California: Burning Troll Productions, LLC, 2013.
- [52] B. Madden, K. Bage and J. Madden, The VDI Delusion: Why Desktop Virtualization Failed to Live Up to the Hype, and What the Future Enterprise Desktop Will Really Look Like, San Francisco, California: Burning Troll Productions, 2012.
- [53] M. Carmody, "Announcing VMware Mirage 5.1," 16 September 2014. [Online]. Available: <http://blogs.vmware.com/euc/2014/09/vmware-mirage-5-1-release.html>.
- [54] S. Clarke, End-user Computing: Concepts, Methodologies, Tools and Applications, Information Science Reference, 2008.
- [55] "Wikipedia," 2015. [Online]. Available: <http://en.wikipedia.org>.

Appendix 1 – Amount of Windows Updates included in each Template update.

Windows Update



Most recent check for updates: 07/10/2014 at 14:31
Updates were installed: 07/10/2014 at 13:16. [View update history](#)
You receive updates: For Windows and other products from Microsoft Update



Most recent check for updates: Today at 11:01
Updates were installed: 07/10/2014 at 13:16. [View update history](#)
You receive updates: For Windows and other products from Microsoft Update

