EMC Performance Optimization for Microsoft Windows XP for the Virtual Desktop Infrastructure

Applied Best Practices

EMC Performance Optimization for Microsoft Windows XP for the Virtual Desktop Infrastructure Applied Best Practices

P/N **H7168**

Contents

This applied best practices guide discusses how to configure Microsoft Windows XP, for optimal performance, in a Virtual Desktop Infrastructure (VDI) implementation. This document provides an overview on how to configure various subsystems of Windows XP to minimize the performance requirements on the shared storage and ESX environment.

Audience

This applied best practices document is intended for all parties responsible for planning, architecting, configuring, deploying, and maintaining the VDI.

Related documents

The following documents provide additional, relevant information.

♦ *Windows XP Deployment Guide* at *http://www.vmware.com*

♦ *VMware VDI Storage Considerations* at *http://www.vmware.com*

♦ *Windows XP: Kernel Improvements Create a More Robust, Powerful, and Scalable OS* at *http://msdn.microsoft.com*

♦ *Recommendations for Aligning VMFS Partitions* at *http://www.vmware.com*

♦ *Pushing the Limits of Windows: Virtual Memory* at *http://blogs.technet.com*

♦ *Understanding Memory Management* at *http://vpivot.com*

Chapter 1 Optimizing Windows XP for VDI - An Overview

This chapter presents this topic:

VDI storage performance challenges

The current VDI platform of choice — Windows XP — was never meant to be deployed in the manner required to support a VDI use case. In its default state, the operating system (OS) performs various activities that cause unwanted overhead in a virtualized instance of the OS and it is critical to control the OS such that it is efficient and predictable.

Planning and designing the storage infrastructure is a critical step because the shared storage must be able to absorb the bursts of input/output (I/O) that occur during the course of a day. A flaw in the design will lead to periods of erratic and unpredictable virtual desktop performance. Users can adapt to slow performance, but unpredictable performance is sure to quickly frustrate them.

To provide a predictable performance to a VDI infrastructure, the storage should be able to handle the I/O load from the clients without causing excessive increases in the response time as measured by the user. This is difficult in VDI implementations due to the volatile nature of the workload.

Figure 1 shows resource utilization of a typical server and a VDI workload.
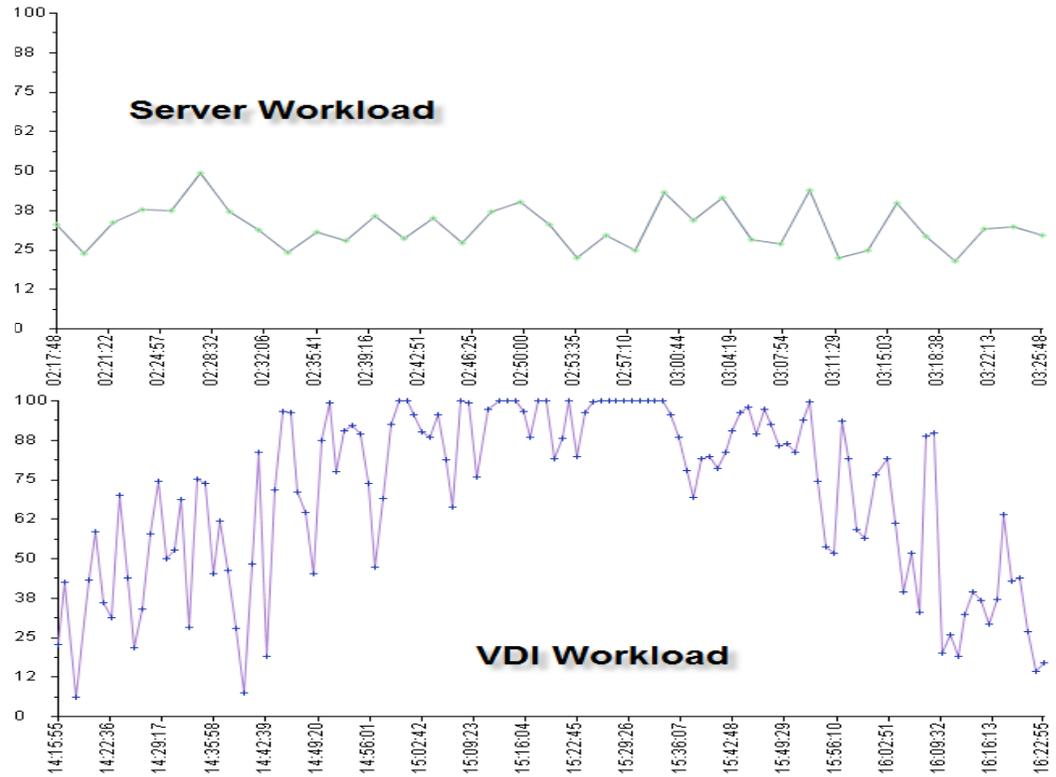
Figure 1   **VDI workload – Example**

In smaller to medium size VDI deployments, the most common issues center around the client experience and total cost of ownership (TCO) of the solution. However, in large-scale deployments, storage scaling becomes the primary issue that prevents production implementations. When thousands of VDI images are deployed on a single array synchronous I/O, spikes from all the virtual desktops can potentially cause performance issues.

The purpose of this applied best practices guide is to provide a set of optimizations that can be implemented to make the virtual machine, Windows, and user profiles to perform more efficiently. In doing so, it is possible to reduce the I/O burden of the OS significantly. This will reduce the storage requirements required to support the VDI infrastructure, which in turn reduces

the initial capital expenditure and operational expenditure outlays that can significantly increase the return on investment (ROI) of the solution.

This guide assumes that a Microsoft Windows domain running on Windows Server 2008 R2 is configured to house the VMware View users and desktops. Many of the optimizations mentioned in this guide can be accomplished by using Microsoft Active Directory (AD) Group Policy Objects (GPO) or by editing the registry directly.

Many of the tweaks and configuration options described in this guide can substantially improve the performance of a virtual machine. However, performance comes at the cost of features. It is the responsibility of the desktop engineering team to find a suitable balance between the features and the performance for their environment.

Chapter 2   Customizing the Virtual Machine

This chapter covers the basic configuration steps to be performed on the virtual machine before creating a base image for use in a VDI deployment. To begin, a Windows XP Professional virtual machine should be created using the procedures described in this chapter.

This chapter presents these topics:

Configure the virtual machine settings

vCPU count

Use as few virtual CPUs (vCPUs) as possible while providing the required amount of CPU resources to the guest OS while creating the virtual machine. Increasing the number of vCPUs increases the work required by the hypervisor to coschedule CPU resources on the physical CPU cores. To ensure that adequate CPU resources are allocated to the virtual machine, ensure that the average CPU utilization reported by the virtual machine is less than 70 percent and the %RDY counter for the virtual machine is less than 10 percent.

If the virtual machine reports low CPU utilization and the %RDY value is higher than 10 percent, the virtual machine vCPU is waiting for free physical CPU resources to schedule its workload. This is an indication of over-provisioned physical CPUs on the ESX host.

Change default SCSI controller

The default small computer system interface (SCSI) controller for new virtual machines is the BusLogic Parallel controller. Performance of the guest OS can be improved by using the LSI Logic Parallel controller instead. Figure 1 on page 8 shows the SCSI Controller dialog box that is displayed during the creation of a new virtual machine.
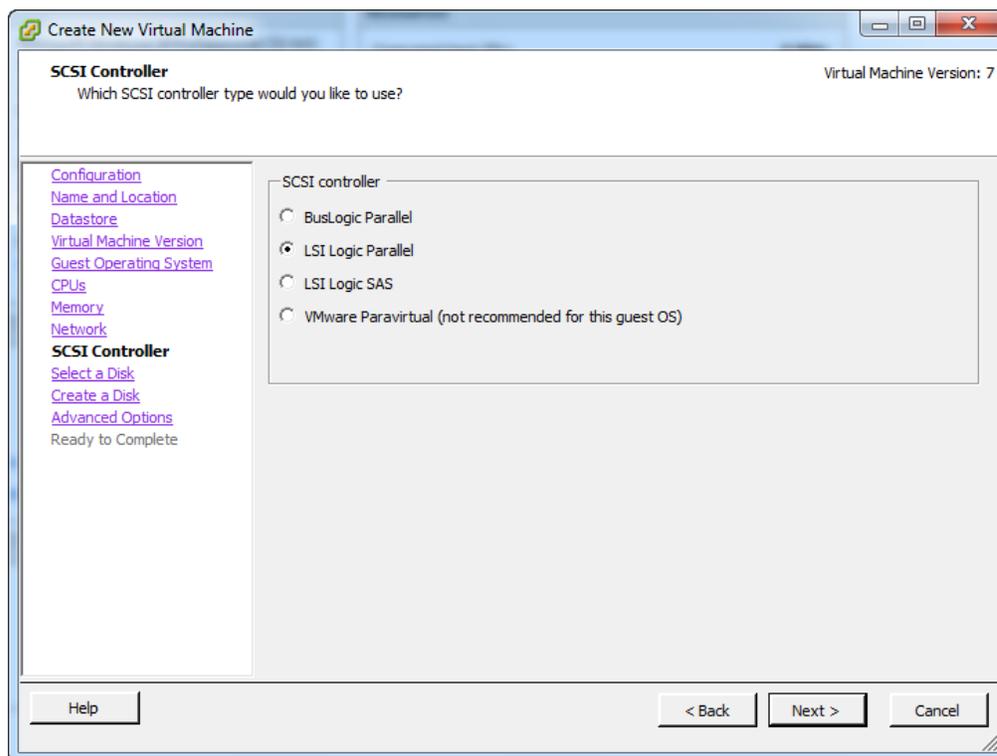
Figure 2   **SCSI Controller dialog box**

Note: Choosing the LSI Logic Parallel controller requires loading the LSI Logic driver during installation such that the boot volume is visible to Windows XP.

Set the Virtual Device Node, on the Advanced Options dialog box, to SCSI (0:0) while creating the virtual machine.
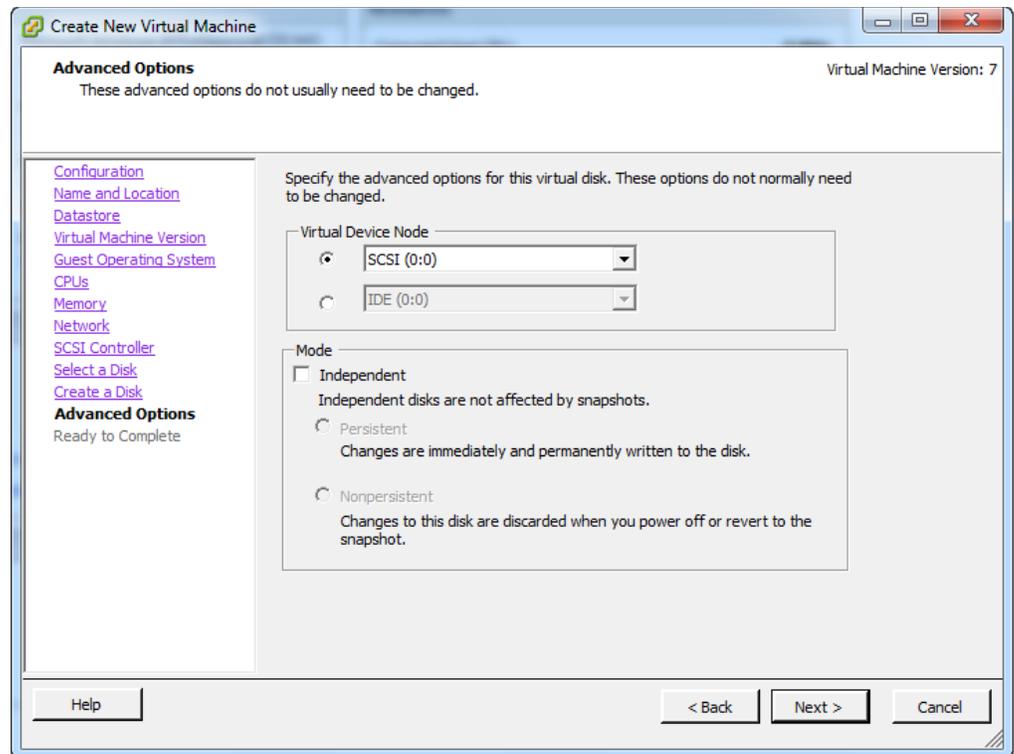
Figure 3   Advanced Options dialog box

Disable virtual machine logging

Every time a virtual machine is powered on, it logs diagnostic information to the data store hosting its VMDK file. For large concentrations of virtual machines, this overhead can be significant.

To disable logging, clear **Enable logging** from the Settings pane in the Options tab as shown in Figure 4. This option sets the **logging = "FALSE"** option in the VMX file for the virtual machine.
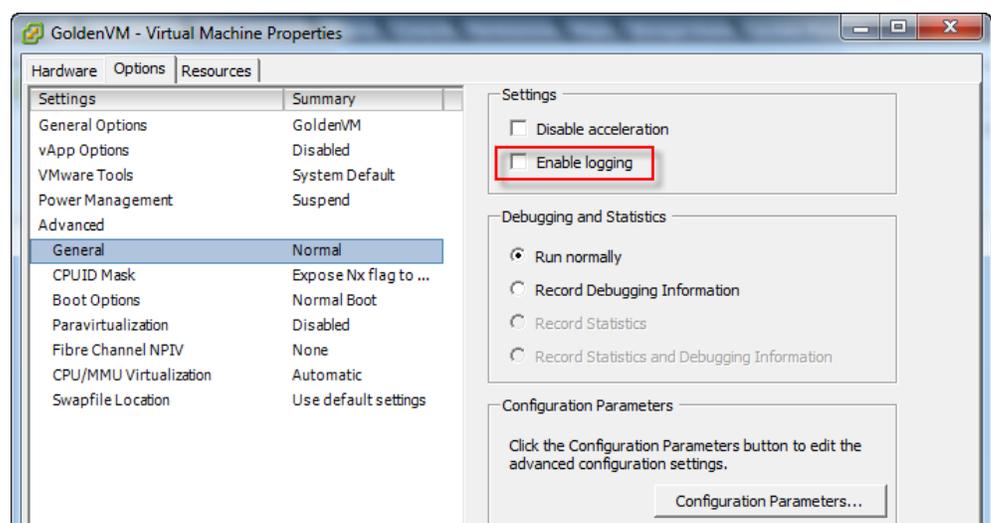
Figure 4   Virtual Machine Properties dialog box

Edit BIOS settings

Any device attached to the virtual machine requires resources to load during startup. Disabling devices that the virtual machine will not use offer more resources for use by the ESX server.

To disable unneeded BIOS devices for a virtual machine:

1. Select **Force BIOS Setup** in the properties of the virtual machine as shown in Figure 5.
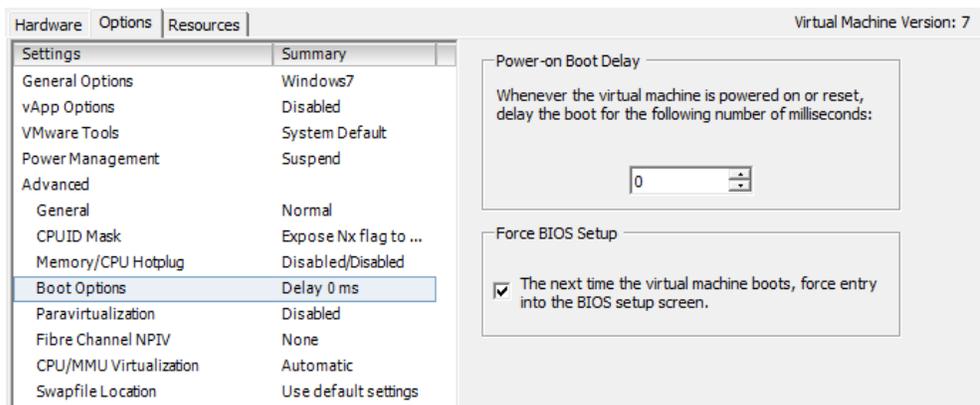
Figure 5   Forcing the virtual machine into BIOS

2. Click **OK** to close the Virtual Machine Properties dialog box and start the virtual machine. The virtual machine will now boot to the BIOS menu.

3. In the BIOS menu select the **Advanced** tab.

4. Select **I/O Device Configuration**.

   Disable the serial and parallel ports for the virtual machine. Do not disable the floppy controller. This is required while installing the OS.

5. Press **F10** to save the configuration and exit the BIOS.

Align the NTFS file system

Aligning the file system of a virtual machine increases the I/O efficiency from the NTFS file system to the underlying storage. Creating a virtual machine within vCenter automatically aligns the VMFS but the guest file system must also be aligned. *Recommendations for Aligning VMFS Partitions* located at VMware.com provides more information on file system alignment.

Windows XP does not provide a method for creating aligned file systems during installation. To work around this, the virtual machine file system can be accessed using fdisk from the ESX service console. The goal is to set the beginning offset for the file system to sector 128. This has the effect of starting the file system at the first 64k boundary
(128 sectors x 512 bytes/sector = 64k). If not done:

1. Create a new virtual machine with the LSI Logic controller as described in Change default SCSI controller on page 12.

2. Log in to the console of the ESX server hosting the new virtual machine.

3. View the VMDK file parameters by typing `cat`.

   Note the ddb.geometry.cylinders value. In this example it is 1305.

```
[root@rtpsol463 GoldenVM]# cat GoldenVM.vmdk
# Disk DescriptorFile
version=1
encoding="UTF-8"
CID=fd3571f8
parentCID=ffffffff
createType="vmfs"

# Extent description
RW 20971520 VMFS "GoldenVM-flat.vmdk"

# The Disk Data Base
#DDB

ddb.thinProvisioned = "1"
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "1305"
ddb.uuid = "60 00 C2 98 fb 32 c2 93-b6 a8 e9 ee 94 54 70 2e"
ddb.longContentID = "6434c8b68356368780b4e971fd3571f8"
ddb.virtualHWVersion = "7"
[root@rtpsol463 GoldenVM]# fdisk ./GoldenVM-flat.vmdk
last_lba(): I don't know how to handle files with mode 8180
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

You must set cylinders.
You can do this from the extra functions menu.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help):
```

Figure 6   VDMK file parameters

4. Edit the -flat VMDK virtual disk file of the virtual machine using fdisk by typing `fdisk ./<VMDK-flat file>` as shown in Figure 7.

```
[root@rtpsol464 GoldenVM]# fdisk ./GoldenVM-flat.vmdk
last_lba(): I don't know how to handle files with mode 8180
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklab
el
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

You must set cylinders.
You can do this from the extra functions menu.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help):
```

Figure 7   Edit VMDK file with fdisk

5. Enter expert mode by typing x and then c to enter the number of cylinders equal to the value of ddb.geometry.cylinders, which was 1305 in this example.

```
Command (m for help): x

Expert command (m for help): c
Number of cylinders (1-1048576): 1305

The number of cylinders for this disk is set to 1305.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

Figure 8   Set the cylinder count

6. Type r to return to normal mode.

7. Type n for new partition. Type 1 to select the first partition and 1 to set the first cylinder and press **Enter**.

   The default value for the last cylinder, which should be equal to the ddb.geometry.cylinders value, appears.

```
Expert command (m for help): r

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1305, default 1305):
Using default value 1305
```

Figure 9   Create the primary partition

8. While still in the expert mode, type b to specify the beginning offset for the disk and then type 1 to specify the first partition and set the new beginning of data to 128.

9. Type p to view the partition table and ensure that **Start** has the correct value of 128.

```
Command (m for help): x

Expert command (m for help): b
Partition number (1-4): 1
New beginning of data (63-20964824, default 63): 128

Expert command (m for help): p

Disk ./GoldenVM-flat.vmdk: 255 heads, 63 sectors, 1305 cylinders

Nr AF  Hd Sec  Cyl  Hd Sec  Cyl     Start      Size ID
 1 00   1   1    0 254  63 1023        128  20964697 83
 2 00   0   0    0   0   0    0          0         0 00
 3 00   0   0    0   0   0    0          0         0 00
 4 00   0   0    0   0   0    0          0         0 00
```

Figure 10   Verify the partition table

10. Type r to return to normal mode.

11. Type t to specify the partition type.

12. Type 7 to set the partition type to NTFS and then type w to write the partition table to disk.

```
Expert command (m for help): r

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 7
Changed system type of partition 1 to 7 (HPFS/NTFS)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 25: Inappropriate ioctl for device.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```

Figure 11   Write the partition table to disk

Now the virtual machine can be booted to install Windows XP.

Note: The error regarding the partition table is normal and can be ignored.

Install Windows XP with LSI driver

To start the install,

1. Attach the ISO image for Windows XP to the CDROM of the virtual machine. Additionally, to get the SCSI driver for the LSI Logic SCSI controller download the driver package LSI_U320_W2003_IT_MID1011438.zip from LSI referenced in VMware KB 1006224.

2. Use MagicISO, Virtual Floppy Drive 2.1 or WinImage to create a floppy image from the files included in symmpi_w2003_1201800.ZIP, which is included in the download.

3. Attach the floppy drive of the virtual machine to the floppy image just created.

4. Start the virtual machine.

The virtual machine boots from the Windows XP ISO to begin the installation.

5. Press **F6** when prompted and load the drivers for the LSI Logic SCSI controller.
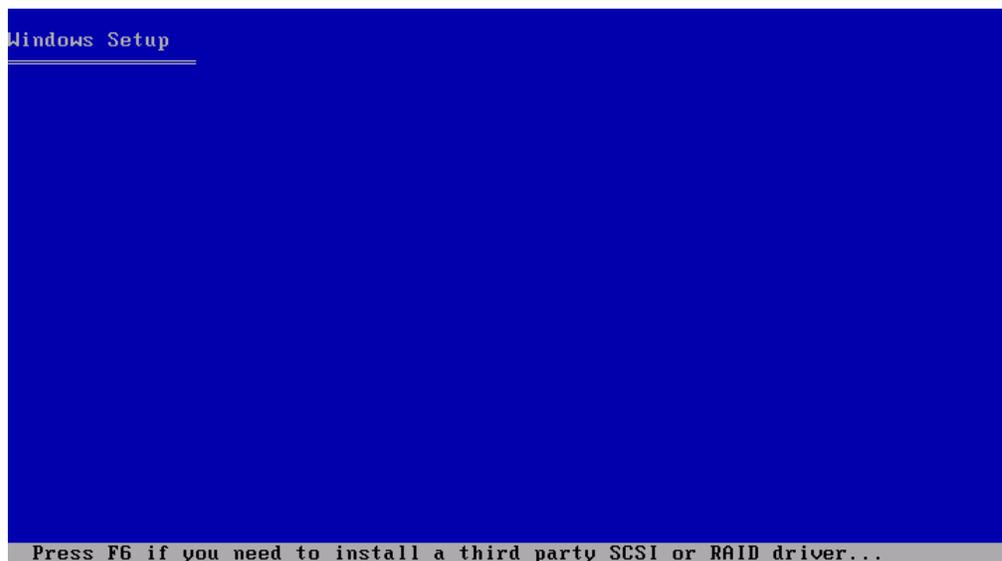
Figure 12   F6 prompt to load SCSI drivers

After loading the LSI Logic SCSI controller drivers, the Windows XP Licensing Agreement page appears.

6. Press **F8** to accept the license.

The **Windows XP Professional Setup** screen appears as shown in Figure 13.
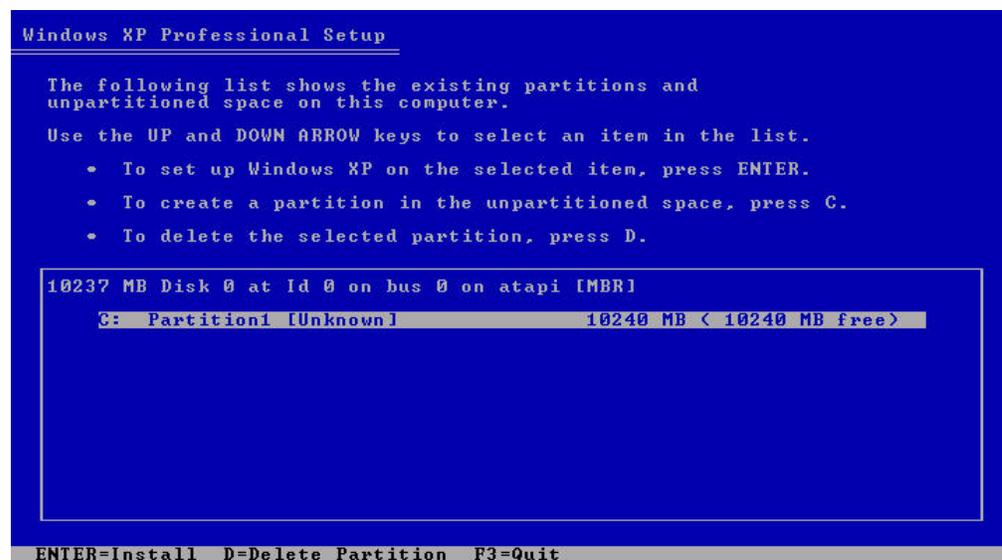
Figure 13   Choose the installation partition

7. Press **Enter** to install Windows XP on the existing entry — Partition1. The existing entry is the aligned disk created earlier to ensure that the entry is not deleted.

When the installation is complete, ensure that a clone of the image is taken for backup.

Chapter 3 Windows XP Operating System Optimization

The Windows boot process creates a brief period of extremely heavy read activity as the OS is loaded into memory. During the boot process, Windows loads drivers, fonts, applications, processes, and other runtime components. Each component, driver, service, and application loaded represents I/Os that must be serviced from the back-end storage.

Managing large VDI deployments at scale requires using the tools, available to the domain administrator, effectively. Many of these settings can be applied by binding GPOs to the organizational units (OUs) containing the virtual desktops. This can be done using the Group Policy Management Console. For the examples provided in this chapter, it is assumed that all desktops are contained in an OU named VMware_View.

For more information on using the group policy effectively, see the Windows Server Group Policy homepage at http://technet.microsoft.com.

This chapter presents these topics:

VMware optimizations

Install VMware tools

Install the latest version of VMware tools to ensure that the virtual desktops are running the latest drivers. These drivers should be a part of the base image. VMware Update Manager can also be used to update VMware tools after deployment.

Disable debug logging for View 3.1.2

The VMware View agent enables debug logging for the client version 3.1.2. This can cause an increase in the IOPS load. This does not affect the View 4 client versions.

To disable debug logging in the 3.1.2 clients, open regedit on the master virtual machine and edit the following keys:

HKLM/SOFTWARE/VMware, Inc./VMware VDM/

Set **DebugEnabled=False** and **TraceEnabled=False**

HKLM/SYSTEM/CurrentControlSet/Services/vmware-viewcomposer-ga/

Set **LogLevel=0**

Windows optimizations

Disable error reporting

When an application or an OS crashes, Windows compiles error reports and attempts to contact Microsoft to debug the issue. Usually, this can be safely disabled. To disable error reporting by using a group policy object, edit the VMware_View GPO and navigate to **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Error Reporting** and set **Disable Windows Error Reporting=Enabled**.

Remove Windows update folders

This is potentially dangerous so consider making a snapshot of the virtual machine before proceeding.

During the patching process, Windows saves a copy of the changed files at the following location:

C:\Windows\$NtUninstall<update name>$

If a service pack has been installed, the files are located at the following location:

C:\Windows\$NtServicePackUninstall$

If these updates are not uninstalled, remove these folders from the system. Implement this tweak if space consumption is of utmost importance.

Additionally, consider deleting the contents of C:\Windows\temp and any profiles in C:\Documents and Settings except for All Users, Default User, Local Service, and Network Service.

Disable automatic updates

The preferred method to update virtual desktops is to update the master image and create new virtual desktops from the updated master. This provides a measure of change control and limits the extreme I/O overhead that can be caused by "patch Tuesday" or other events that cause the entire desktop population to simultaneously download and apply updates.

To configure this option, edit the VMware_View GPO and navigate to **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Update** and set the **Configure Automatic Updates** to **Disabled**.

Remove unnecessary applications

Many applications load boot time processes to aid with the application performance by caching and prefetching specific data to memory ahead of any actual user request. Examples of this are the boot time stubs loaded by QuickTime, Real, Adobe Acrobat, and others. This can be desirable for dedicated machines, but it causes unnecessary I/O in a VDI environment.

Remove Windows components

Remove any Windows components that are not required in the environment such as MSN Explorer or Outlook Express. Navigate to **Control Panel > Add/Remove Programs > Add/Remove Windows Components** and remove any unneeded components.

Some components are not shown by default, such as WordPad, Messenger, and accessories. KB article 223182 at http://support.microsoft.com provides more information to remove these hidden applications.

Change NTFS behavior

There are several attributes of NTFS that can be tuned using the fsutil command to minimize the file system overhead. These options include the ability to disable the creation of DOS style 8.3 file names and disabling the "last accessed" time stamp. The "last accessed" option can reduce the write workload for users who run applications that access many files.

To disable these features, open a command window and type the following commands:

fsutil behavior set disablelastaccess 1

fsutil behavior set disable8dot3 1

Purge Windows file protection DLL cache

Windows file protection caches system files in the %systemroot%\system32\dllcache folder to help protect against file corruption. This feature is explained in the Microsoft article KB222193. There is no supported method for disabling WFP for XP post SP2, but the DLL cache can be purged to save space by running the following commands from a command prompt:

sfc /purgecache

sfc /cachesize=0

WFP is still active, but does not cache DLL files in the %systemroot%\system32\dllcache folder.

Uninstall unneeded fonts

System fonts are loaded during the boot process from the C:\windows\fonts folder. Removing fonts can reduce the I/O load during the system boot and make the bootup process faster. The freeware utility FontXplorer from http://moonsoftware.com can be used to easily view and delete unneeded fonts from the system.

Disable the Windows prefetcher

The Windows prefetcher proactively places files required for the boot and application load in a contiguous area of the disk, in an effort to improve the application load performance. By default, XP prefetches data for both applications and the system To change this behavior, launch regedit.exe and edit **HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher**.

The valid settings for EnablePrefetcher are [0 = Disabled; 1 = Application Launch Prefetch; 2 = Boot Prefetch; 3 = Prefetch All]. The default value is set to **Prefetch All**, which prefetches both application and boot time files. Setting this value to 0 disables all Windows prefetching. If disabling all prefetching, delete all prefetch files under C:\Windows\Prefetch.

Windows XP: Kernel Improvements Create a More Robust, Powerful, and Scalable OS on http://msdn.micrsosoft.com provides more information about the Windows XP prefetcher.

Disable hibernation

Virtual desktops should not be allowed to go into a hibernation state due to the high I/O cost of writing the contents of the system RAM to the C:\hiberfil.sys file. Because the hiberfil.sys file is equal in size to the RAM allocated to the virtual machine, removing it can save quite a bit of space as well.

To disable hibernation, navigate to **Start > Settings > Control Panel > Power Options > Hibernate** and clear **Enable Hibernation**.

Disable the language bar

If there is no need for the language bar, it can be disabled to save some memory and I/O overhead by opening a command prompt and typing **Regsvr32.exe /u msutb.dll**. This unregisters the DLL for the language bar. There will not be an option to re-enable it from the GUI. The DLL can be re-registered if the functionality is required in the future.

Disable system restore

System restore is not required for a majority of VDI environments and can be safely disabled as users' data should be moved out of the virtual machine by using folder redirection. If the virtual machine becomes corrupt, a replacement can be quickly provisioned. This saves on the I/O overhead that is required for creating and maintaining snapshot data on the virtual machine as well as saving a significant amount of space.

To disable system restore, right-click **My Computer** and select **Properties**, select the **System Restore** tab, and then select **Turn off System Restore**.

Disable paging the executive

By default, Windows writes kernel-mode drivers and system code to the Windows page file when not in use to make more memory available to the system. If the virtual machine is assigned sufficient memory, this behavior can be disabled to save the additional I/O overhead of writing and then reading system code into the Windows page file.

To disable paging of the executive, open regedit.exe on the master image and navigate to **HKLM > SYSTEM > CurrentControlSet > Control > Session Manager > Memory Management**. Find the **DisablePagingExecutive** key and change the value from 0 (default) to 1.

Properly size virtual machine RAM

From a capacity point of view, setting RAM for the virtual machine has an immediate effect on the size of a virtual machine due to page files. For every virtual machine, there are two page files to be considered. Windows creates the C:\pagefile system file equal to the size of the RAM installed to begin with and expands it on demand to meet the virtual memory requirements of the Windows OS.

Additionally, VMware creates a vswap file for the virtual machine that is equal to the RAM size allocated to the virtual machine and can be stored with the virtual machine or in a separate data store. In effect, the total swap file space allocated to a virtual machine is approximately two times the RAM allocated to the virtual machine. Therefore, accurately sizing the memory is important.

Sizing Memory for Virtual Machines on page 29 provides details about sizing the virtual machine RAM. *Pushing the Limits of Windows: Virtual Memory* by Mark Russinovich provides more details (http://blogs.technet.com).

Set page file to fixed size

By default, Windows dynamically expands and shrinks the Windows page file as required. This can lead to fragmentation of the page file and unnecessary I/O overhead. It is recommended to set the page file to a fixed size. *Pushing the Limits of Windows: Virtual Memory* at http://blogs.technet.com provides guidance on how large the fixed page file size should be.

Disable indexing

Indexing can create a tremendous amount of read overhead for a virtual machine because it builds the index cache. To disable indexing of the local disk, from **My Computer,** highlight C:\ drive and select **Properties**. Select the **General** tab and clear **Allow Indexing Service to index this disk for fast file searching**.

Disable logon screen saver

The logon screen saver can consume CPU cycles when the virtual machine is powered on but when not in use. On the golden image, open regedit.exe and navigate to **HKEY_USERS\.DEFAULT\Control Panel\Desktop**. Edit **ScreenSaverActive** and change the value to 0 to disable.

Optimize the registry

The Windows registry is a database that holds much of the configuration information for how the OS and applications should operate. In the process of building a golden image for use in a VDI environment, it is possible to have a significant amount of space in the registry committed to programs that have been uninstalled in the past.

RegScrubXP.exe is a freeware utility that can remove deprecated keys from the registry, which makes the registry smaller and more lightweight. It can be downloaded from CNET at http://download.cnet.com.

The registry can be loaded more quickly and with less I/O overhead after it is compressed by eliminating file space gaps. The registry can also be compressed by using the NTRegOpt.exe utility from http://download.cnet.com.

Disable unnecessary services

Not all Windows XP default services are useful in a virtualized instance of the OS. Services such as the Wireless Zero Configuration serve no purpose in a VDI context and should be disabled. Disable services by launching the Services MMC plugin by running **START\RUN\services.msc**.

Table 1 lists several services that can be considered for disabling in a VDI environment.

Table 1   Services to consider disabling

Wireless Zero Configuration	Shell Hardware detection	Machine Debug Manager (MDM)
Telephony	Task Scheduler	Network Location Awareness
SSDP Discovery Service	Themes	IPSec Services
Remote Registry	Help and Support	Windows Audio
Security Center	Indexing Service	

The msconfig.exe tool included with Windows XP can also be used, but it is not recommended because it disables services that Windows needs to run correctly. For a comprehensive list of services and configuration options, review http://www.blackviper.com.

User profile hive cleanup service

Some applications do not clean up dependencies correctly and fail to close all open handles to other resources, which can cause profile corruption. UPHClean works by closing handles left open that prevent a clean logoff of the current user. It accomplishes this by logging all application and file handles opened during a user session and closing them during logoff. This should be considered mandatory when using roaming profiles because it prevents profile corruption.

Disable logging of informational printing events

By default, all printing events are written to the system log file. To reduce the amount of data that is logged, disable the logging of information events by opening **Printers and Faxes\File\Server Properties\Advanced** tab and clearing **Log spooler informational events**.

Application and service shutdown time

Slow logoff time can be caused by Windows waiting to kill hung applications or services on shutdown. By default, Windows waits 20,000 ms to kill hung services and applications. This can be shortened by changing the values in the following locations:

♦ Current user — Time to wait before killing an application for the currently logged-in user. To change this value, edit the key **HKCU\Control Panel\Desktop\WaitToKillAppTimeout**.

♦ All users — Time to wait before killing an application for any user logged in to the machine. To change this value, edit the key **HKU\.Default\Desktop\WaitToKillAppTimeout**.

♦ System — Time to wait before killing services. To change this value, edit the key **HKLM\System\CurrentControlSet\Control\WaitToKillServiceTimeout**.

Disable success logging

By default, Windows records both successful and failed login attempts. For some environments, the overhead of logging successful logins is not required and can be disabled. This could be considered a potential security issue.

To disable successful login events from being written to the security log, edit the VMware_View GPO and navigate to **Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Audit Policy** and configure the **Audit account logon events** setting to audit failure attempts only.

Group policy refresh interval

By default, all computers in the domain attempt to refresh their group policy settings every 90 minutes with a 30-minute offset. This can be extended to limit the amount of network bandwidth that is consumed when refreshing the group policy. By default, the group policy is also updated at every boot of the OS.

To change the group policy refresh interval, edit the VMware_View GPO and select **Computer Configuration\Policies\Administrative Templates\System\Group Policy** and configure the **Group Policy refresh interval for computers** and set it to the desired time. In most environments, this can safely be set to 24 hours.

Chapter 4   Default User Profile Customization

This chapter presents these topics:

Windows profile customizations

Creating a default profile

Creating a default user profile enables customizing the desktop to a greater extent when unique users connect to the virtual desktop. This section describes the procedures to be followed when setting up the default user profile for the golden image virtual machine.

It is recommended to create a local user for the sole purpose of creating the default user profile. Use this user account to log in to the virtual machine and customize the environment. When all the customizations are completed, this user profile is copied over the current default user profile located at C:\Documents and Settings\Default User. This process ensures that each unique user gets a customized desktop that adheres to the specifications of the desktop engineer.

Change the default theme

Windows Themes controls how the desktop is rendered. The default theme uses many optimizations to paint a more graphically complex user interface that creates increased processor and network overhead. The Classic theme is suggested for VDI environments due to its lower overhead. If the Themes service is disabled, the user interface (UI) defaults to the Windows classic theme.

Adjust for best performance

Some of the more advanced UI features such as menu fading and animations require extra CPU and memory resources to use. To turn off these effects, navigate to:

My Computer > Properties > Advanced > Performance and select **Adjust for best performance**.

Disable the screen saver

The screen saver is completely unneeded in a VDI environment because users are not connecting directly to the console of the virtual machine. To disable the screen saver for all VDI users, edit the VMware_View GPO and navigate to **User Configuration\Policies\Administrative Templates\Control Panel\Display** and set **Disable the Display Control Panel=Enabled**.

Disable desktop cleanup wizard

The desktop cleanup wizard runs every 60 days to remind users to delete rarely used desktop items. This can be disabled by right-clicking the desktop and selecting the **Properties/Desktop/Customize Desktop and clearing the Run Desktop Cleanup Wizard every 60 days** checkbox.

Turn off system sounds

Go to **Control Panel/Sounds and Audio Devices/Sounds** and set the "Sound scheme" to "No Sounds".

Completing the default profile

After all the required changes are made to the setup user account, log off the virtual desktop. At this point, Windows finishes creating the user profile for that specific user. To complete the process, log in as an administrator or an administrative user.

After logging in, complete the following steps:

1. Navigate to **Control Panel/System/Advanced/User Profiles/Settings**.

2. Highlight the profile of the user account that will be the template for the default user and click **Copy To**.

3. Click **Browse** under **Copy profile to** and navigate to C:\Documents and Settings\Default User, and then click **OK**.

4. Click **Change** under **Permitted to use** and click **Everyone**.

5. Click **OK**.

Any user logging in to the virtual machine inherits the changes made to the default user profile. The default user profile can also be copied out to a network share, if required.

Post-install

At this juncture, all user configuration is done. There are two more steps to perform before taking a snapshot of the virtual machine to deploy the linked clones.

Disk cleanup

Run the disk cleanup wizard to find and delete any files that have accumulated in the image that do not need to be present in the image. To get started, right-click the C:\ drive and select **Properties/General Tab** and click **Disk Cleanup**. Remove all files that do not need to be part of the image.

Defrag the hard drive

The golden image should have its disk defragmented before being used for image deployment. However, users should never defrag on the deployed virtual desktops. There are two reasons for this — defragging a drive creates a very high I/O overhead and can quickly cause sufficient I/O to the shared storage to cause performance issues.

Secondly, if the desktops are thinly provisioned using VMware linked clones, defragmenting the virtual machine hard drive causes the virtual machine hard disk to quickly expand in size. For example, if you run defragmentation on a virtual machine that was using only 600 MB of space, after deployment, it can increase to 4 GB or more of allocated space.

Appendix A Sizing Memory for Virtual Machines

This appendix presents this topic:

Introduction

The amount of RAM allocated to a virtual machine has a direct effect on a number of factors that impact both capacity and performance requirements for a virtual machine. This appendix briefly explains on how to allocate RAM to Windows XP virtual machines. For the sake of simplicity, memory overhead of the ESX host will not be included in any calculations.

This appendix also assumes a good understanding of ESX memory management technologies such as memory over-commitment, transparent page sharing (TPS) and vswap usage. The *vSphere Resource Management Guide* at www.vmware.com provides more information.

If not overcommitting memory then it is recommended to disable TPS, which will free up CPU cycles which can be used by the virtual machines. TPS can be disabled by setting Mem.ShareScanGhz to zero under the advanced settings for an ESX host. This can provide a 5 percent performance boost to the virtual machines on the host.

Virtual machine active memory (working set)

The amount of RAM that is actively used by a virtual machine is referred to as the active working set. This can be seen with the %ACTV counter in esxtop or the Active memory counter in vCenter under the Resource Allocation tab. On a virtual machine with 2 GB of RAM, if the max working set size is 1 GB (%ACTV == 50%) then the virtual machine is using 50 percent of its allocated RAM. Assuming other virtual machines exhibit similar behavior a 2:1 over-commit ratio could be used without causing excessive swapping by the ESX host.

On the ESX server, if the active working set for all virtual machines is less than the total available host memory, then all the virtual machines run at full speed because each virtual machine is able to address all the RAM it requires without the hypervisor ballooning or swapping virtual machine memory pages to disk.

Conversely, when the sum of all active working sets on the host exceeds the amount of available RAM on the ESX host, the hypervisor will be forced to swap pages from the virtual machine memory to vswap. The hypervisor has no knowledge of which pages are in the active working set of the virtual machine and will swap pages to meet the memory demands placed on the ESX host. This will lead to very poor performance of the virtual machine and should be avoided at all costs.

As an example, see the following table in which a host with 32 GB of RAM is hosting virtual machines each configured for 2 GB of RAM. Each virtual machine has on average 50 percent active memory.

Table 2   Memory over-commit

VM count	Active memory in host	Comments
16	16 * 2 GB * 50% = 16 **GB**	Without memory over-commit, only 50% of the host's memory is actively in use.
32	32 * 2 GB * 50% = **32 GB**	Memory is over-committed by 200% but active memory is equal to host memory. Virtual machines will run at full speed until usage exceeds 100% of host memory.
48	48 * 2 GB * 50% = **48 GB**, limited to **32 GB** by host	These virtual machines want 48 GB of RAM but are limited to the 32 GB that is installed on the host. ESX must swap to allow these machines to run and performance will be degraded.

When used appropriately the memory over-commit feature allows virtual infrastructure administrators to drive the ESX hosts to high memory utilization without degrading performance which is not possible with most physical systems.

Assigning RAM to virtual machines

The goal in choosing the appropriate RAM size for the golden image virtual machines is to give the virtual machines enough memory to hold the entirety of the working set in memory while keeping the memory over-commit ratio as low as possible. This avoids Windows having to page because there is not enough RAM available in the guest and also avoids ESX having to page because it cannot reclaim memory from active guests fast enough when under sufficient memory pressure.

To determine how much RAM to allocate to the golden image virtual machine, the guest should be placed under load. With the peak load running on the virtual machine the amount of active memory needs to be measured. This is called the active working set size.

There are several methods that can be used to measure the active working set size. In esxtop look at the %ACTV memory counter. This will display the active working set size as a percentage of allocated memory. If this value never approaches 100% then the virtual machine is not actively using all RAM allocated to it.

In vCenter it is reported as "Active" under the Guest Memory section on the Resource Allocation tab. Alternatively, a more accurate measurement can be found by launching VM Statistics Logging on the VM and observing the Memory Active counter under VM Memory.

For best balance between performance and memory utilization it is suggested that the virtual machine have at least 25% more RAM allocated than the max active load witnessed by the VM to avoid having Windows write data to its page file, which keeps the active working set for the virtual machine in RAM instead of virtual memory space.

As an example, if the golden image virtual machine has a peak active memory threshold of 600 MB then the virtual machine should have at least 750 MB of RAM allocated. The maximum number of virtual machines to load on the host could be calculated by dividing the maximum amount of ESX host memory by the maximum active memory of the golden virtual machine. In this case it would be 32,768 / 600 = 54 virtual machines. This gives us a safe over-commit ratio of approximately 1.25 to 1 if virtual machine and ESX memory overhead are not factored in.

Additionally, when Windows is booted it zeros out its memory space. This causes the %ACTV memory to briefly run up to 100 percent. This can lead to periods of vswap usage if multiple virtual machines are rebooted in concert as could happen in the event of an ESX host failure and VMware HA is enabled for the cluster.