



## VMware Workstation 5.5

# Performance Benchmarking Guidelines for VMware Workstation 5.5

## Introduction

This white paper provides guidance in implementing benchmark tests using VMware Workstation 5.5.

The paper discusses:

- Performance benchmarking methodology.
- Configuring the systems under test to produce the best performance possible. This includes the host system, the VMware software, and the guest system.
- Ensuring that the tests are making “apples-to-apples” comparisons.

The paper also includes examples of common pitfalls that can affect the accuracy or relevance of the results obtained by benchmarking tests.

The guidelines in this paper are intended to assist in the acquisition of meaningful, accurate, and repeatable benchmarking results. These guidelines are useful when you perform comparisons between VMware Workstation and either native systems or other virtualization products, as well as when you run benchmarks in virtual machines in general. The guidelines should not, however, be considered VMware best practices in all cases. Among other reasons, this is because benchmarking sometimes involves saturating one resource while overprovisioning others, something that would not be desirable in a production environment.

**Note:** Though this paper specifically addresses VMware Workstation 5.5, most of the guidelines recommended here also apply to VMware Player 1.0.

## Highest-Performance Configurations

There are some areas in which the best-performing configurations of VMware Workstation virtual machines vary slightly from the configurations of native machines. One of the goals of this paper is to provide guidance about these variations.

To this end, we discuss configuration of the hardware, the host operating system, the VMware Workstation software, and the operating systems and applications in the individual virtual machines.

## “Apples-to-Apples” Comparisons

It is important when doing performance comparisons to make sure that the configurations of the systems being compared are as similar as possible. Good performance comparisons only change one variable at a time. For example, if comparing the performance of VMware



Workstation with another virtualization product, the hardware and software configurations should be equivalent, with the only difference being which virtualization product is used.

If there is a performance or resource cost associated with an optional feature or capability of a VMware product, that cost should be measured and reported separately from other benchmarking results.

For example, the use of **growable disks** (as opposed to the more traditional **preallocated disks**) within VMware Workstation can significantly reduce disk-space requirements. Growable disks can also have an impact on performance, and that performance impact should not be charged against virtualization. Instead, the increased disk-usage efficiency of growable disks is a potential added benefit of virtualization, but an optional one that should be weighed separately against its (usually) slight decrease in performance.

**Note:** Please check the relevant VMware product end-user license agreement (EULA) before publishing any benchmarking data regarding VMware products.



## Overview

This section begins with information about terminology, then provides a checklist-style overview of the guidelines in this white paper. Most of the items listed in the overview are described in much greater detail later in this paper.

### Terminology

Throughout this paper we use the terms **native**, **host**, and **guest**. Brief definitions are included here. More information about these terms, as well as about the other terms in bold font throughout this white paper, may be found in the [Glossary on page 28](#).

A **native system** is a computer running a single operating system, and on which the applications run directly in that operating system.

A **host system** is a computer on which VMware Workstation software is running.

A **host operating system** is an operating system running directly on a host computer. VMware Workstation runs within the host operating system.

A **guest** is a virtual machine running within VMware Workstation.

A **guest operating system** is an operating system that runs inside a virtual machine.



## Benchmark Design

This section provides an overview of the topics in this paper applying to benchmark design.

More detail about each of these topics is provided in [Benchmarking Design on page 11](#).

- Clearly define the parameters being measured and a metric with which to measure them (for example, operations per second, jobs per hour, average response time).
- If running a publicly-available benchmark, make sure to follow all guidelines associated with it.
- If running a custom-written benchmark, make sure it incorporates common benchmarking principles (specific test purpose, meaningful metric, reproducibility, and so forth).
- Any report of the benchmark results should include enough details about the experimental setup for the reader to reproduce the results.
- When attempting to saturate and benchmark a specific system component (CPU, memory, network, and so forth) make sure that no other system component is constrained.
- Avoid using the timestamp counter (TSC) from within a **virtual machine**.
- If you do use the TSC from within a virtual machine, edit the virtual machine configuration file (the `.vmx` file) to pass the actual physical TSC through to the guest. Otherwise the timing results may be inaccurate.
- Consider using an external client (on a different physical system) for workloads that involve network traffic.
- One method of timing workloads is to ping an external machine and capture timestamps when that machine receives the pings.



## Hardware

This section provides an overview of the topics in this paper applying to hardware. More detail about each of these topics is provided in [Hardware on page 15](#).

### General

- For performance comparisons, it is best to run all tests on the same system. When running on the same system is not possible, at least use identical systems.
- Make sure to run all tests on hardware supported by the VMware software you are using.

### CPU

- When comparing a virtual system to a native system, make sure the native system is configured to have the same number of **physical CPUs** as the virtual machine has **virtual CPUs**.

### Memory

- When comparing a virtual system to a native system, make sure the native system is configured to have the same amount of physical memory as the virtual machine has virtual memory.
- Make sure that all systems have sufficient memory.

### Disks (General)

- Make sure the host hard drive is large enough that it has an ample amount of free space.

### Disks (SAN, NAS, and RAID)

- If using a network storage device (that is, **SAN** or **NAS**), make sure the system under test is the only system using that device.
- Use a dedicated switch for SAN or NAS.
- If using Fibre Channel storage, ensure that your connection bandwidth is as expected (that is, 1Gbps or 2Gbps).
- Make sure the read and write caches on SAN and NAS storage devices are enabled and configured to the appropriate sizes.
- Make sure the queue depth on host bus adapters is configured properly, as this configuration can have a large impact on performance.
- If using RAID, make sure you have chosen the correct type for your application (for example, RAID level 0, 1, 2, 3, 4, 5, 6, 0+1, etc.)

### Networking

- Make sure the test systems do not share networking components with other systems.
- Use network switches instead of hubs.
- Make sure that all networking components are of the appropriate speed (that is, use Gigabit switches and cables with Gigabit network interface cards).
- Use similar network interface cards on server and client.
- Make sure the systems have only as many network interface cards as necessary for the tests.



**Miscellaneous**

- Remove or disable all devices that are not part of your experiment. These might include audio devices, optical drives (i.e., CD or DVD), floppy drives, USB ports and devices, network cards, and so forth.



## Host Operating System

This section provides an overview of the topics in this paper applying to the host operating system. More detail about each of these topics is provided in [Host Operating System on page 17](#).

### Memory

- Make sure memory is sized so as to avoid excessive page faults whenever possible.

### Disks (General)

- In Windows host systems, make sure hard drive write caching is enabled.
- If your host system uses IDE hard drives, make sure DMA access for these drives is enabled.
- Make sure you have ample free space on the host hard drive.

### Disks (SAN and NAS)

- Make sure the read and write caches on SAN and NAS storage devices are enabled and configured to the appropriate sizes.
- Use write back (as opposed to write through) for SAN and NAS write caches unless write-through is required for reliability reasons.

### Networking

- To avoid using the wrong network interface card, disable any card you will not use.
- Make sure all network interface cards are configured to their highest-performance mode.
- Unless there is a good reason to do otherwise, use OEM-recommended settings for the network interface card driver.
- If your network performance is low, consider disabling the VMware host-only or **network address translation** (NAT) adapters from within the host operating system.

### Miscellaneous

- Make sure that any non-default system configuration settings are intentional.
- Disable screen savers, virus checkers, and any unnecessary services on the host system.



## VMware Workstation

This section provides an overview of the topics in this paper applying to the VMware Workstation software. More detail about each of these topics is provided in [VMware Workstation on page 19](#).

### CPU

- Don't overcommit CPU resources (for example, don't run multiple virtual machines on a single-processor host system).
- Allow for the CPU overhead required by virtualization.

### Memory

- Carefully select the amount of virtual memory you allocate to your virtual machines. Either too much or too little virtual memory can reduce performance.
- Disable memory trimming for the virtual machine.
- Disable page sharing for the virtual machine.

### Disks

- Store **virtual disks** on local drives (rather than on network drives).
- Configure the virtual machine to emulate SCSI instead of IDE disks.
- Configure the virtual machine to use preallocated disks (as opposed to growable disks).
- Configure the virtual machine to use **persistent disks** (as opposed to **nonpersistent disks**).

### Networking

- Remove or disable any virtual networking devices that are not required for your tests.

### Miscellaneous

- Make sure you are using a generally-available version of VMware Workstation (rather than a beta or debug version).
- Make sure you are not running VMware Workstation in debug mode.
- Leave VMware Workstation logging enabled, as it has an extremely small impact on performance.
- Make sure the log files are stored on a local disk (rather than on a network drive).
- Full screen mode** and **normal (windowed) mode** now have nearly the same performance.
- Make sure you have selected the correct guest operating system in the virtual machine settings editor.
- Make sure that any non-default VMware Workstation configuration settings are intentional.
- Configure the virtual machines to start with optical drives (i.e., CD or DVD) disconnected. If the optical drives won't be used, completely remove them from the virtual machine's configuration.
- Use purpose-built virtual machines.



## Guest Operating System and Applications

This section provides an overview of the topics in this paper applying to guest operating systems and application software. More detail about each of these topics is provided in [Guest Operating Systems on page 23](#).

### CPU

- A single-processor virtual machine should usually use a **uniprocessor (UP) hardware abstraction layer (HAL) or kernel** (as opposed to a **symmetric multiprocessor (SMP) HAL/kernel**). Remember that when changing a virtual machine from multiprocessor to single-processor, the HAL/kernel usually remains SMP.
- When conducting SMP scaling experiments, consider configuring both the single-processor and the multiprocessor virtual machines with SMP HALs/kernels.
- When deciding between single-processor and dual-processor virtual machines, consider configuring the single-processor virtual machine with a UP HAL/kernel and the dual-processor virtual machine with an SMP HAL/kernel.
- If your benchmarks don't involve processor scaling, it may be best to stick with single-processor configurations.
- If comparing a native system to a virtual machine, make sure the HAL/kernel types in both are kept the same (that is, either both UP or both SMP).
- When running a single-threaded benchmark, use a single-processor virtual machine.
- When running a multi-threaded benchmark with an SMP HAL/kernel, make sure there is enough parallelism to keep both CPUs busy.
- With both 32-bit and 64-bit versions of software becoming increasingly common, make sure that the native and virtual systems are both running comparable versions of operating systems and applications (that is, either both 32-bit or both 64-bit).
- 64-bit guests run only on 64-bit systems. If the 64-bit system has **hardware virtualization assist**, it should be enabled (typically done through the BIOS).
- Consider the possible impact of guest operating system idle loop behavior on host CPU usage.

### Memory

- Make sure that memory is sized so as to accommodate the working set size of the guest, thus avoiding excessive page faults.

### Disks

- In Windows guest systems, make sure hard drive write caching is enabled.
- If you are using virtual IDE hard drives in the virtual machine (as opposed to higher-performance virtual SCSI drives), you should ensure DMA access for these drives is enabled.
- Defragment disks before running tests. Note that this must be done from the inside out: first within the virtual machine, then with the VMware Workstation defragmentation tool, and lastly within the host operating system.

### Networking

- Use the high-performance vmxnet or e1000 virtual network interface card drivers (instead of the default vlmance driver).



**Miscellaneous**

- Make sure you are using generally-available versions of all operating systems, applications, and benchmarks (rather than beta or debug versions), and that all applicable patches and updates are installed.
- Make sure you are using a guest operating system version that is supported by the VMware software you are using.
- If the native system has been tuned (registry, swap space, and so forth), repeat the tuning procedure in the guest. Note that the best results in the guest may be achieved with different settings than those in the native system.
- Make sure that any non-default configuration settings are intentional. This applies to operating systems, applications, and benchmarks.
- Make sure the guest operating system is not detecting new hardware (the New Hardware wizard may impact performance).
- Disable screen savers, virus checkers, and any unnecessary services on the guest system.
- Direct benchmarking output to a log file instead of to the display.
- Install **VMware Tools** in the guest operating system.



## Benchmarking Design

This section provides detailed tips and guidance in the design of benchmarking experiments.

### General Methodology

- Before planning the testing, clearly define the parameters being measured and a metric with which to measure them (such as operations per second, jobs per hour, or average response time).
- If you are running a publicly-available benchmark (for example, SPEC\*), make sure you follow all the guidelines associated with configuring, running, and reporting for that benchmark, both on the native system and within the guest.
- If you are running a custom benchmark, make sure that it incorporates the well-understood principles of benchmarking: specific test purpose, a meaningful metric (such as time, operations per second, or bytes transferred), reproducibility, and so forth.
- Any report of the benchmark results should include enough details about the experimental setup for the reader to reproduce the results.
- When trying to saturate and benchmark any specific system component (such as CPU, memory, or network), ensure that no other resource on the system is constrained in either the native or the virtual machine. Be aware that virtualization has overhead in terms of CPU, memory, etc., and provision for this overhead for components not being tested. For example, before making a native to virtual machine network-bandwidth comparison, make sure the processor load is not limiting the bandwidth achieved in either case.
- An ideal setup for workloads that involve network traffic is to use an external client (on a different physical system) to send network traffic to and receive network traffic from a virtual machine.

### Timing Considerations

- Timing numbers reported within the virtual machine can be inaccurate, especially when the processor is overcommitted. If you report numbers from within the guest, this fact should be noted in your results.
- One method of timing workloads is to ping an external machine and capture timestamps when that machine receives the pings. (If the benchmark takes long enough, you can also use a stopwatch to measure time.)

To use `ping` to time workloads, run `tcpdump` (in Linux) or `WinDUMP` (in Windows) on an external system. (The examples below show `tcpdump` in a C shell. `WinDUMP` works similarly.)

On the external system, run:

```
tcpdump ip proto icmp and host <system-under-test>
```

In this example, `<system-under-test>` is the IP address of the system under test.

From within the virtual machine, run:

```
ping -c 1 <external system>
```

Then run the benchmark to be timed:

```
ping -c 1 <external system>
```

In this case, `<external system>` is the IP address of the external system running `tcpdump` or `WinDUMP`.



You can then determine the duration of the experiment by simply subtracting one time stamp from the other.

**Note:** Make sure the external system has sufficient resources (`tcpdump` is sensitive to low memory, for example). Also, make sure the external system is fairly close in the network topology to the system under test (that is, make sure there are relatively few “hops” between the systems).

For more information about `tcpdump` and `WinDUMP`, see:

<http://www.tcpdump.org/>

<http://www.winpcap.org/windump/>

- In some cases your workload can't use the above ping method (perhaps because the timing granularity is so small that the network latency of the pings is significant). If your workload uses the timestamp counter (TSC) to measure time from within the virtual machine, you can allow the virtual machine to see the real hardware TSC (rather than a virtualized TSC that represents virtual time) by adding the following line to the applicable `.vmx` file:

```
monitor_control.virtual_rdtsc = "FALSE"
```

Note that some guest operating systems, especially SMP guests, won't boot with this setting. Therefore, before modifying this variable, refer to the timekeeping white paper available at:

[http://www.vmware.com/pdf/vmware\\_timekeeping.pdf](http://www.vmware.com/pdf/vmware_timekeeping.pdf)

- More information about Linux guest timing issues may be found in VMware knowledge base article 1420 “Clock in a Linux Guest Runs More Slowly or Quickly Than Real Time,” at: [http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1420](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1420)



## Benchmarking Tools

This section lists benchmarking tools that may help you compare the performance of various systems.

### CPU-Related Benchmarks

- Passmark (available at <http://www.passmark.com/>)
- SPEC CPU2000 (available at <http://www.spec.org/cpu2000/>)

### Memory-Related Benchmarks

- Passmark (available at <http://www.passmark.com/>)

### Disk-Related Benchmarks

- Passmark (available at <http://www.passmark.com/>)
- IOMeter (available at <http://www.iometer.org/>)

### Networking-Related Benchmarks

- Netperf (available at <http://www.netperf.org/netperf/NetperfPage.html>)
- SPECweb2005 (available at <http://www.spec.org/web2005/>)

### Comprehensive Multi-Tier Benchmarks

- SPECjAppServer2004 (available at <http://www.spec.org/jAppServer2004/>)

### Discouraged Benchmarks

We don't recommend use of the following benchmarks, as our experience has shown that they can produce unexpected results in virtual machines:

- Sisoft Sandra
- LMBench

## Performance Monitoring Tools

A number of tools are available for profiling Windows and Linux systems. Although they are useful for profiling native and host systems, their use within guest virtual machines should be avoided, due to unreliable results (the white paper *Timekeeping in VMware Virtual Machines*, at [http://www.vmware.com/pdf/vmware\\_timekeeping.pdf](http://www.vmware.com/pdf/vmware_timekeeping.pdf), contains more information on this topic). These profiling tools include:

### Windows: Perfmon

**Perfmon** can be used to profile all aspects of the system, including CPU, memory, disk, network, and so forth. VMware Workstation exports **Perfmon** counters from a Windows guest operating system to the host operating system (when the host operating system is also Windows). To run **Perfmon**, choose **Start > Run**, then type **perfmon** in the **Open:** box.

In the **Perfmon** console, choose **Console Root**, then double-click **System Monitor**. Click the + sign to open the Add Counters dialog box. Under **Performance Object**, there are a number of objects of interest including **Memory**, **Network Interface**, **Physical Disk**, **Processor**, **System**, and **VMware**.

To see detailed information regarding what the counter actually profiles, select any of the counters in any of the performance objects, then click the **Explain** button.

Further information about **Perfmon** may be found in the **Help** menu within the application, or by visiting:



<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/8e9e5b72-13af-4525-9b60-52864b9d49b2.msp>

Further information about the VMware performance counters may be found in the *VMware Workstation 5 User's Manual*, or by using the help function within `Perfmon`.

**Linux: vmstat**

Invoke this tool by executing:

```
vmstat sample_interval num_times_to_sample
```

For example, `vmstat 1 10` samples every second and generates about ten samples.

`vmstat` provides information regarding the CPU, memory, swap, disk, and system (interrupts, context switches).

Refer to the `vmstat` man page for more information.

**Linux: sar**

Invoke this tool by executing:

```
sar sample_interval num_times_to_sample
```

For example, `sar 1 10` samples every second and generates about ten samples.

`sar` provides comprehensive information regarding the CPU, memory, swap, disk, system, processes, file system, and network. For the comprehensive list of counters monitored, run `sar -A`.

Refer to the `sar` man page for more information.



## Hardware

This section provides detailed guidance about hardware used in the benchmark testing.

### General

- In general for performance comparisons it is best to run both tests on the same system. When this is not possible, use identical systems.
- Make sure to run all tests on hardware supported by the VMware software you are using.

### CPU

- For an “apples-to-apples” performance comparison between native and virtual machines, configure the same number of physical CPUs in the native system (while measuring the native performance) as you configured virtual CPUs in the virtual machine. This may require reducing the number of processors in your system when doing the native tests.

To do this in Windows, use the `/numproc` switch in the `boot.ini` file.

To do this in Linux, use the `maxcpus` variable in either the `grub.conf` or `lilo.conf` file (depending on your Linux version).

### Memory

- For an “apples-to-apples” performance comparison between native and virtual machines, configure the same amount of physical memory in the native system (while measuring the native performance) as you configured virtual memory in the virtual machine. This may require reducing the amount of memory in your system when doing the native tests.

To do this in Windows, use the `/maxmem` switch in the `boot.ini` file.

To do this in Linux, use the `mem` variable in either the `grub.conf` or `lilo.conf` file (depending on your Linux version).

- Don't overcommit memory:
  - Make sure the host has more memory than the total amount of memory that will be used by the host operating system plus the sum of the memory that will be used by all the virtual machines.
  - Though it may not be true for production environments, over-provisioning memory in experimental configurations may be a good idea, as over-provisioning can ensure that excessive page faults are avoided. This can allow other resources to be benchmarked without the need to worry about page faults.

### Disks (General)

- For best performance, make sure the host hard drive is large enough that it will have an ample amount of free space. Performance can degrade considerably when VMware Workstation has to use a nearly-full host hard drive to write guest growable disks and **snapshot**, checkpoint, or redo files.

### Disks (SAN, NAS, and RAID)

- If the virtual disks are stored on a network storage device (such as SAN or NAS), make sure you are the only user connected to that device. If connecting through a switch (Fibre Channel or Ethernet), make sure your machine is the only one on the switch. Other traffic on a storage device or switch can affect your results.



- For Fibre Channel connections, is your connection bandwidth 1Gbps or 2Gbps? This bandwidth is dependent on the configuration of both the host bus adapter and the SAN switch port. To determine the storage connection bandwidth, refer to the management tool provided by your SAN vendor (EMC Navisphere, for example).
- When possible, make sure the read and write caches on SAN and NAS storage devices are enabled and configured to the appropriate sizes. Caches that are either disabled or too small can affect performance.
- Make sure that the queue depth on the host bus adapter is configured appropriately, as this configuration can significantly affect performance (refer to the documentation supplied by your storage vendor for details).
- RAID storage systems come in various configurations (for example, RAID level 0, 1, 2, 3, 4, 5, 6, 0+1, etc.), with different trade-offs for performance and reliability. If using RAID, make sure you have made the appropriate choice for your experimental setup.

### Networking

- Avoid cross-traffic noise over the network while conducting the experiments. Either use direct cables between the systems or use a private network switch. Use dedicated network interface cards on both machines for the connection.
- Use network switches instead of hubs.
- Make sure that all the networking infrastructure is appropriately rated. For example, when connecting systems containing Gigabit network interface cards, make sure to use Gigabit switches and Gigabit-rated cables.
- If possible, use similar network interface cards on systems under test so that they function well with each other. Using similar cards also helps to ensure that send and receive have similar performance. Ideally, you should use similar client and server machines as well, with similar PCI architecture and configuration. The difference between PCI and PCI-X, for example, can affect networking performance.
- Don't have more physical network interface cards than absolutely necessary. This avoids the unnecessary overhead associated with processing broadcast packets, protocol control packets, and so forth.

### Miscellaneous

- Remove or disable all devices that are not part of your experiment. These might include audio devices, optical drives (i.e., CD or DVD), floppy drives, USB ports and devices, network interface cards, and so on.



## Host Operating System

This section provides detailed guidance regarding the host operating system used in benchmark tests.

### Memory

- Make sure memory is sized so as to avoid excessive page faults in the host operating system. Information about measuring page faults can be found in [Memory on page 25](#).

### Disks (General)

- On Windows systems the Disk Properties tab associated with each hard drive provides a check box to enable write caching on the disk and, in some cases, to enable advanced performance on the disk. Selecting one or both of these boxes can improve the host's disk performance in general, and selecting them for the host disks containing VMware virtual disk files can improve VMware disk performance in particular, especially when VMware is making heavy use of the disk.
- Most versions of Windows can use direct memory access (DMA) for higher performance on IDE hard disks. However, this feature may not be enabled by default.  
You can enable DMA access in Windows NT family operating systems as follows.
  1. Right-click **My Computer** and choose **Manage** from the pop-up menu. The Computer Management console will appear.
  2. Select **Device Manager** in the left pane.
  3. Click the + sign beside **IDE ATA/ATAPI controllers**.
  4. For each IDE controller, right-click the entry and open its Properties dialog box.
  5. Under **Advanced Settings**, make sure the **Transfer Mode** for each device is set to **DMA if available**.
  6. Restart Windows for the new settings to take effect.
- For best performance, avoid low free space on the host drive. Performance can degrade considerably when VMware Workstation has to use a nearly-full host hard drive to write guest growable disks and snapshot, checkpoint, or redo files.

### Disks (SAN and NAS)

- When possible, make sure the read and write caches on SAN and NAS storage devices are enabled and configured to the appropriate sizes. Caches that are either disabled or too small could have an adverse impact on performance.
- SAN and NAS storage devices can often be configured with either write through or write back caching. Write-back caching provides better performance (with a slight impact on reliability, as the system could go down with data in cache). Write-through caching trades some performance for increased reliability (data is always synchronized between the cache and storage).

### Networking

- If your system has multiple physical network interface cards (built-in or in PCI slots) make sure you are using the intended network interface cards for the experiment. If you have more than one network interface card, it is easy to enable and use the wrong one. To avoid this confusion, disable the network interface cards you do not plan to use.



- Most modern network interface cards can operate in multiple modes (such as 10, 100, or 1000Mbps; half duplex or full duplex). Make sure the network interface cards are in full-duplex mode and are configured at their maximum possible bandwidth (that is, 1000Mbps for a Gigabit network interface card).
- Don't change any of the default network interface card driver settings unless there is a valid reason to do so (use the OEM listed recommendations).
- If your network performance is low and your virtual machines are not using **host-only** or NAT networking, you can try disabling the VMware host-only or NAT networking adapters from within the host operating system. For more information about this, see VMware VMware knowledge base article 684 "Slow Network Performance when Using Windows Browsing on the Host," at:  
[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=684](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=684)

### Miscellaneous

- Check that any non-default system configuration settings are intentional.

### Disable Unnecessary Services

- Disable screen savers, virus checkers, and any unnecessary services on the host system.

In Windows:

- To see all currently running processes, bring up Windows Task Manager and click the **Processes** tab.
- To enable or disable services, choose **Control Panel > Administrative Tools > Services**.
- To see programs that are launched on startup from the Start menu, choose **Start > Programs > Startup**.
- To add or remove programs that are launched on startup from the Start menu, do one of the following:

**In Windows 2000:** Right-click within the tool bar, select **Properties**, and click the **Advanced** tab in the Taskbar and Start Menu Properties window. Click the **Advanced** button, select **Start Menu > Programs > Startup**. Make sure to perform this procedure for the **Start Menu** under both **All Users** and your user name.

**In Windows XP or Windows 2003:** Right-click within the tool bar, select **Properties**, and click the **Start Menu** tab in the Taskbar and Start Menu Properties window. Click the **Customize** button, then the **Advanced** button. Select **Start Menu > Programs > Startup**. Do this under both **All Users** and your user name.

In Linux:

- To see all currently running processes, run `ps auxww` or `top`
- To see all system services, run `chkconfig --list`

The `chkconfig` command provides a command-line interface for administering the system services that are invoked from `/etc/rc[0-6].d`

- To add or remove system services, run:  
`chkconfig service-name [on|off|reset]`

For example, `chkconfig atd off` and `chkconfig crond off` turn off the `at` and `cron` daemons so that no scheduled jobs can run during the experiment.



## VMware Workstation

This section provides detailed guidance regarding the VMware Workstation software used in benchmark tests.

### CPU

- Don't overcommit CPU resources when running benchmarks:
  - Avoid running a dual-processor virtual machine on a single-processor host system, even if the single-processor host has **hyper-threading** (that is, two logical CPUs).
  - Avoid running two or more single-processor virtual machines on a single-processor host system, even if the single-processor host has hyper-threading.
  - Since virtualization has its own CPU overhead, make sure that CPU resources are not overcommitted on the host system.

### Memory

#### Memory Allocation within Virtual Machines

- Carefully select the amount of virtual memory you allocate to your virtual machines:
  - On the one hand, you should allocate enough memory to hold the working set of applications you will run in the virtual machine.
  - On the other hand, unlike native systems, on which more physical memory is almost always better, allocating too much memory to a virtual machine can reduce overall system performance, and can even reduce the performance of the overallocated virtual machine. This is because allocating to a virtual machine much more memory than its working set requires can increase memory pressure at the host level (as the host balances VMware Workstation's memory requests against the system's other memory needs).

This performance penalty due to overallocation may be exacerbated if you select the **Fit all virtual machine memory into reserved host RAM** option (to find this option, choose the VMware Workstation **Edit** menu, then choose **Preferences > Memory**).

#### Memory Trimming and Page Sharing

- VMware Workstation uses memory-trimming to deallocate unused virtual machine memory for the host to reallocate. Trimming usually has little impact on performance, and it may be needed in low-memory situations. However, memory trimming can reduce disk performance in a guest. To disable memory trimming, use one of the following two methods:
  - Add `MemTrimRate=0` to the `.vmx` file for the virtual machine.
  - With the guest virtual machine opened, choose the VMware Workstation **VM** menu, then choose **Settings > Options > Advanced**. In the **Settings** section, select **Disable memory page trimming**.
- VMware Workstation uses page-sharing to allow guest memory pages with identical contents to be stored as a single copy-on-write page. Page sharing decreases host memory usage, but consumes other system resources, potentially including I/O bandwidth. You may want to avoid this overhead for guests for which host memory is plentiful and I/O latency is important. To disable page sharing, add the following line to the `.vmx` file for the virtual machine:



```
sched.mem.pshare.enable = "FALSE"
```

**Note:** Because memory trimming and page sharing can increase disk I/O demands, it may be helpful to turn them off when benchmarking disk-intensive applications. In other situations, however, these two features may improve performance.

## Disks

- For the best and most repeatable performance, store virtual disks on local disk drives instead of on network drives.
- VMware Workstation can emulate either IDE or SCSI disks. Virtual SCSI disks perform better than virtual IDE disks, and are therefore recommended.
- Use preallocated disks instead of growable disks. When creating a new virtual machine, the disk wizard offers the option **Allocate all disk space now?** Selecting this option preallocates all the space at the time you create the virtual disk. It requires as much disk space as the size you specify for the virtual disk. The alternative is a growable disk which starts small and grows as needed. Preallocated disks are recommended for performance experiments since they yield better performance and more reproducible results than growable disks.
- **Independent virtual disks** may be either **persistent** or **nonpersistent**. With persistent disks, changes are immediately and permanently written to the disk. All changes to a persistent disk remain, even if a snapshot is reverted. With nonpersistent disks, changes to the disk are discarded when you power-off or revert to a snapshot. Independent, persistent disks have the highest performance and are recommended for most workloads. You can configure a virtual machine for independent persistent disks as follows:
  1. With the guest virtual machine opened but powered off, choose the VMware Workstation **VM** menu, then choose **Settings > Hardware**.
  2. In the Device window, select **Hard Disk** and click the **Advanced** button in the lower right.
  3. In the **Mode** section, be sure **Independent** and **Persistent** are both selected.

## Networking

- You should remove or disable any virtual networking devices that are not required for your tests.

## Miscellaneous

- Make sure the system under test is running a general availability (GA) release of VMware Workstation (rather than a beta or debug version).
- VMware Workstation can run in two modes: normal mode and debug mode. The debug mode is slower than normal mode, and thus should not be used when performing benchmarking tests.

Ensure that you are not running in debug mode as follows:

1. With the guest virtual machine opened but powered off, choose the VMware Workstation **VM** menu, then choose **Settings > Options > Advanced**.
  2. In the **Settings** section, be sure **Run with debugging information** is not selected.
- VMware Workstation 5.5 has logging enabled by default. Although this generates disk I/O, the impact is extremely small, and we recommend that logging always be left enabled. If



you want, though, you can disable logging by adding the following line to the `.vmx` file for the virtual machine in question:

```
logging = "FALSE"
```

More information about virtual machine logging, including how to change the location of the log file, may be found in VMware knowledge base article 1028 "Disabling the Workstation Log File or Storing the Log File in a Different Location," at:

[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1028](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1028)

- Make sure the log files are stored on a local disk (rather than on a network drive).
- Due to changes in VMware Workstation beginning with version 5.5, full screen and normal (windowed) mode have nearly the same performance (despite the recommendations included in the "Performance Tuning" section of the *VMware Workstation 5 User's Manual*).
- Make sure you have selected the correct guest operating system in the virtual machine settings editor, as an incorrect selection could reduce performance. To verify the current selection, choose the VMware Workstation **VM** menu, then choose **Settings > Options**. Look in the Version window. (Note that to change the selection you must first power off the virtual machine.)
- Check that any non-default VMware Workstation configuration settings are intentional.
- Some operating systems, including Windows NT and Windows 98, poll the optical drives (i.e., CD or DVD) every second or so to see if a disc is present. This polling allows the operating systems to execute autorun programs. This polling can also cause VMware Workstation to connect to the host optical drive, which can make the virtual machine appear to pause while the drive spins up. To avoid such pauses, remove the optical drive from your virtual machine unless you need it for your experiment.

Another option is to configure your virtual machine to start with the optical drive disconnected. This way, the virtual machine seems to have a virtual optical drive containing no disc (and VMware Workstation does not connect to your host optical drive).

To make this change, choose the VMware Workstation **VM** menu, then choose **Settings**, click the CD-ROM item in the **Device** list, and clear the **Connect at power on** check box.

When the virtual machine is configured not to automatically connect to the CD-ROM at power on you can still use the CD-ROM from within the virtual machine by selecting the **Connected** box after the machine is powered on.

### Use Purpose-Built Virtual Machines

It is good practice to run benchmarking tests on virtual machine that were created specifically for your performance tests (that is, purpose-built virtual machines) rather than reusing existing virtual machines. This is because existing virtual machines may have unnecessary applications or services installed that could impact performance.

Creating a purpose-built virtual machine typically involves some or all of the following steps:

1. Create a fresh virtual machine.
2. Install an operating system.
3. Install any operating system upgrades or patches desired.
4. Install any applications or benchmarks that will be used.
5. Start and stop the machine to verify its correct functioning and to ensure that all hardware has been discovered and that all software installations have completed.



6. Check the log files for this virtual machine to make sure no errors or unexpected warnings are being reported.

Use this clean virtual machine, or **clones** of it, for all tests.

**Note:** If you will be using clones, remember that **linked clones** do not perform as well as full **clones**. Therefore, when creating a clone, make sure to choose **From current state**, then **Create a full clone**.

Avoid running virtual machines from snapshots or linked clones.



## Guest Operating Systems

This section provides detailed guidance regarding the guest operating systems running in virtual machines.

### CPU

The following guidelines relate to guest operating systems and CPUs in virtual machines.

#### UP Versus SMP HAL/Kernel

- Most guest operating systems can be configured to use either a UP HAL/kernel or an SMP HAL/kernel. The UP operating system versions are for single-processor systems. If used on a multiprocessor system, a UP operating system will recognize (and use) only one of the processors. The SMP versions, while required in order to fully utilize multiprocessor systems, may also be used on single-processor systems. Due to their extra synchronization code, however, SMP operating systems used on single-processor systems are slightly slower than UP operating systems.

For best performance, a single-processor system should therefore be configured to use a UP operating system. For more details on this topic, see VMware knowledge base article 1077 "High CPU Utilization of Inactive Virtual Machines," at: [http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1077](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1077)

- Most operating systems automatically select an appropriate HAL/kernel when they are first installed in a virtual machine: UP for single-processor virtual machines, SMP for dual-processor virtual machines. When a virtual machine with a UP HAL/kernel is reconfigured to have two processors, the typical behavior of the guest operating system is to automatically switch to an SMP HAL/kernel. If that virtual machine is later reconfigured to have a single processor, however, it typically does *not* automatically switch to a UP HAL/kernel.

For this reason, among others, it is best to create fresh virtual machines specifically for the benchmarking tests (as detailed in [Use Purpose-Built Virtual Machines on page 21](#)), rather than to reuse machines created previously.

**Note:** Some newer versions of Windows, with appropriate BIOS and hardware support, may be able to seamlessly switch between UP and SMP HALs. More information about this topic can be found in the section entitled "Virtual SMP Best Practices" within the white paper "Best Practices Using VMware Virtual SMP," at: [http://www.vmware.com/pdf/vsmp\\_best\\_practices.pdf](http://www.vmware.com/pdf/vsmp_best_practices.pdf)

(this white paper primarily addresses VMware ESX Server, but the section referenced above also applies to VMware Workstation).

- When conducting SMP scaling experiments, it may be desirable to compare a system running an SMP operating system version on a single processor to a system running an SMP operating system version on two processors. If you keep the HAL/kernel the same in both systems you change only one variable (the number of processors). Note that this consideration applies to native, host, and guest systems.
- When conducting experiments intended to help determine whether to run an application in a single-processor virtual machine or in a dual-processor virtual machine, it may be desirable to configure the single-processor machine with a UP HAL/kernel (since this provides the best single-processor performance) and the dual-processor machine with an SMP HAL/kernel (since this is the only choice available for a dual-processor system). This way, you configure each virtual machine to provide the best performance possible.



- If your benchmarks don't involve processor scaling, it may be best to stick with single-processor configurations (and UP HALs/kernels), thus avoiding unnecessary SMP overhead.
- If comparing a native system to a virtual machine, make sure the HAL/kernel types in both are the same (that is, either both UP or both SMP).
- When running a single-threaded benchmark, make sure you are using a UP HAL/kernel. Use of an SMP HAL/kernel in this case incurs SMP overhead without any benefits.
- When running a multithreaded benchmark with an SMP HAL/kernel, make sure there is enough parallelism to keep both CPUs busy.
- For more information about working with SMP virtual machines, see:  
[http://www.vmware.com/pdf/vsmp\\_best\\_practices.pdf](http://www.vmware.com/pdf/vsmp_best_practices.pdf)

### 32-bit Versus 64-bit CPUs

- 64-bit versions of both operating systems and application software (databases, application servers, and so forth) are becoming increasingly common. Make sure you use similar types of operating system and application software (that is, 32-bit or 64-bit), both natively and within the guest (for example, compare a 64-bit operating system running a 32-bit application natively to a 64-bit operating system running a 32-bit application within the guest, not to a 64-bit operating system running a 64-bit application within the guest).
- VMware Workstation 5.5 supports 64-bit guest operating systems. These 64-bit operating systems, however, run only in 64-bit virtual machines. 64-bit virtual machines, in turn, have a number of hardware requirements:
  - A 64-bit virtual machine does not run on a 32-bit processor.
  - Some early 64-bit processors did not include segmentation support in long mode, and thus do not support 64-bit virtual machines.
  - Some newer 64-bit processors include hardware virtualization assist features. These features are called **Virtualization Technology** (VT) by Intel, and, when available, will be called **Secure Virtual Machine** (SVM) by AMD. Processors that include hardware virtualization assist must have the features enabled (typically done through the BIOS) in order to support 64-bit virtual machines.

More information about this topic can be found under **Virtualization Assist** in the [Glossary on page 28](#) or in VMware knowledge base article 1901 "Hardware Requirements for 64-Bit Guest Operating Systems," at:

[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1901](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1901)

VMware also provides a processor check utility that you can use to determine if a processor supports 64-bit virtual machines. The utility is available at:

<http://www.vmware.com/download/ws/#utilities>

Instructions for the utility are available at:

[http://www.vmware.com/pdf/processor\\_check.pdf](http://www.vmware.com/pdf/processor_check.pdf)

### Idle Loops

- When the system is otherwise inactive, some operating systems may spin in their idle loops for some time before issuing halt instructions, whereas other operating systems may halt immediately. Typically, operating systems that spin in idle loops before halting are running SMP HALs/kernels.



In our experiments, we've found that some versions of Windows behave in this way when running SMP HALs, whereas other versions (or even the same version, but with a different service pack installed) halt quickly when they are idle.

The impact of this behavior may be reduced by adding (or modifying) the `monitor.idleLoopSpinUS=` variable in the `.vmx` configuration file, which overrides the idle loop handler's behavior on a per-virtual machine basis:

```
monitor.idleLoopSpinUS = "n"
```

In this case `n` is the number of microseconds the virtual machine is allowed to spin in its idle loop before it is halted by VMware Workstation. The default value is 2000, and the recommended range of settings is between 100 and 4000 (values outside this range are unlikely to result in resource-utilization improvements).

The `monitor.idleLoopSpinUS` variable has the following effects:

- A lower value means that the virtual machine is halted more quickly when idle (and thus uses fewer physical CPU cycles).
- A higher value means that the virtual machine is allowed to spin longer before being halted (and thus uses more physical CPU cycles). A higher value can also result in better virtual machine performance, however, because the CPU can wake up more quickly from spinning in its idle loop than from a halted state.

For further details please refer to VMware knowledge base article 1077, "High CPU Utilization of Inactive Virtual Machines," at:

[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faaid=1077](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faaid=1077)

and VMware knowledge base article 1730, "ESX Server Reports Increased CPU Utilization for Idle Microsoft Windows Server 2003 SP1 SMP Virtual Machines," at:

[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faaid=1730](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faaid=1730)

(the latter article specifically refers to VMware ESX Server 2.5.1, but the information applies equally to VMware Workstation 5.5).

## Memory

- The performance impact of page faults can be higher in virtual machines than in native systems. Therefore, for best performance, set guest memory size to accommodate the guest's working set, thus avoiding excessive page faults.

To determine if page faults are an issue in a particular virtual machine:

- In Windows:

Use the **Pages/sec** counter (which may be found in the **Memory** Performance object within Perfmon).

For more information refer to <http://support.microsoft.com/kb/889654>

- In Linux:

Run `vmstat` to display the `--swap--` counters including:

`si`: Amount of memory swapped in from disk (KB/s)

`so`: Amount of memory swapped out to disk (KB/s)

Ensure these two counters are as close to zero as possible.

For additional information about how page faults affect performance, and how to measure them, see VMware knowledge base article 1687, "Excessive Page Faults Generated By



Windows Applications May Impact the Performance of Virtual Machines,” at:  
[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1687](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1687)

## Disks

- On Windows systems, the Disk Properties tab associated with each hard drive provides a check box enabling write caching on the disk and, in some cases, a check box enabling advanced performance on the disk. Selecting one or both of these features can improve the disk performance, especially in workloads that make heavy use of the disk.
- It is best to use virtual SCSI hard disks in the virtual machine. If you are using virtual IDE hard disks, however, and the guest operating system is Windows 95 OSR2 or later, you should make sure DMA access for the IDE hard disks is enabled.

You can enable DMA access in Windows 95 family operating systems as follows:

1. Right-click **My Computer** and choose **Properties** from the pop-up menu. The Device Manager appears.
2. Click the + sign beside **Disk Drives** to show your guest’s individual drives.
3. Right-click the entry for each IDE drive and open its Properties dialog box.
4. Under **Settings**, select the box labeled **DMA** and accept any warning Windows displays.
5. Restart Windows for the new settings to take effect.

To enable DMA access in Windows NT family operating systems, see the instructions in [Disks \(General\) on page 17](#).

## Defragment Disks

- Defragment your disks before running your experiments. This defragmentation must be done in stages, from the inside out:
  1. Defragment the virtual disk in the booted-up virtual machine. For example, in a Windows XP guest virtual machine, use the Windows XP Disk Defragmenter tool from within that virtual machine.
  2. Use the VMware Workstation Defragmentation tool option.
    - With the guest virtual machine opened but powered off, choose the VMware Workstation **VM** menu, then choose **Settings > Hardware > Hard Disk > Defragment**.
    - Note:** This is required only for growable virtual disks, and not for preallocated virtual disks.
    - Or use the `vmware-vdiskmanager` command-line utility, which is found in the directory where VMware Workstation was installed (see the *VMware Workstation 5 User’s Manual* for more information about this command).
  3. Defragment the host disk (for example, in a Windows 2003 Enterprise host, use the Windows 2003 Enterprise defragment tool).

## Networking

- Make sure you are using one of the high-performance virtual network interface devices (vmxnet or e1000), instead of the default vlnace device. In addition to improved performance, this avoids a process called “NIC morphing,” in which the vlnace device is automatically replaced with the vmxnet device. Such morphing can have a nondeterministic impact on networking performance.



## Miscellaneous

- Make sure the system under test is running general-availability (GA) releases of all operating systems, applications, and benchmarking software (rather than beta or debug versions) with all applicable patches and updates installed.
- Make sure you are using a guest operating system version that is supported by the VMware software you are using. This includes not just the type of operating system (e.g., Linux), the variant (e.g., Red Hat Enterprise Linux), and the version (e.g., release 4), but even the specific kernel (e.g., SMP).
- If you have tuned the native system (registry, swap space, and so forth), a similar tuning exercise should be performed on the guest system before comparing native to virtual machine performance. Settings on the native system may not be applicable within the virtual machine and it may therefore be necessary to repeat the full tuning process on the virtual machine.
- Check that any non-default system configuration settings are intentional.
- For workloads that include guest startup (such as automated tests) make sure that the virtual machine is not detecting new hardware. If new hardware is detected, the New Hardware wizard (on Windows) or `kudzu` (on Linux) may start up. In this case, complete the new hardware configuration process then save the virtual disk before performing the tests.
- Disable screen savers, virus checkers, and any unnecessary services. For further information, see [Disable Unnecessary Services on page 18](#).
- Direct the output from the application or benchmark to a log file instead of to the display. This avoids the added overhead associated with the virtualized display.

## VMware Tools

- Make sure the VMware Tools suite is installed on the virtual machine.  
VMware Tools provides a suite of utilities and drivers. These utilities and drivers vary somewhat across guest operating systems, but typically include an optimized SVGA driver, a mouse driver, the VMware Tools control panel, and support for such features as shared folders, defragmenting growable virtual disks, optional synchronization of the virtual machine's clock with the host computer's clock, VMware Tools scripts, and the ability to connect and disconnect devices while the virtual machine is running.

To install VMware Tools, power up the virtual machine, open the VMware Workstation **VM** menu, then choose **Install VMware Tools**.

**Note:** Please check the relevant VMware product end-user license agreement (EULA) before publishing any benchmarking data regarding VMware products.



## Glossary

**Clone:** A copy of a virtual machine. See also Full Clone and Linked Clone.

**Console:** See VMware Virtual Machine Console.

**Full Clone:** A copy of the original virtual machine that has no further dependence on the parent virtual machine. See also Linked Clone.

**Full Screen Mode:** A mode of operation in which a guest virtual machine's display occupies the host computer's entire screen. See also Normal (Windowed) Mode.

**Growable Disk:** A type of virtual disk in which only as much host disk space as is needed is initially set aside, and the disk grows as the virtual machine uses the space. See also Preallocated Disk.

**Guest:** A virtual machine running within VMware Workstation. See also Virtual Machine.

**Guest Operating System:** An operating system that runs inside a virtual machine. See also Host Operating System.

**Hardware Abstraction Layer (HAL):** A layer between the physical hardware of a computer and the software that runs on that computer designed to hide differences in the underlying hardware, thus allowing software to run on a range of different architectures without being modified for each one. Windows uses different HALs depending, among other factors, on whether the underlying system has one CPU (Uniprocessor (UP) HAL) or multiple CPUs (Symmetric Multiprocessor (SMP) HAL). See also Kernel.

**Hardware Virtualization Assist:** See Virtualization Assist.

**Host-Only Networking:** A type of network connection between a virtual machine and the host. Under host-only networking, a virtual machine is connected to the host on a private network, which normally is not visible outside the host. Multiple virtual machines configured with host-only networking on the same host are on the same network. See also Network Address Translation.

**Host Operating System:** The operating system running directly on the host computer. VMware Workstation runs within the host operating system. See also Guest Operating System.

**Host System (or Host):** A computer system on which VMware Workstation software is running.

**Hyper-Threading:** A processor architecture feature that allows a single processor to execute multiple independent threads simultaneously. Hyper-threading was added to Intel's Xeon and Pentium® 4 processors. Intel uses the term "package" to refer to the entire chip, and "logical processor" to refer to each hardware thread.

**Independent Disk:** Independent disks are not included in snapshots. Independent disks can in turn be either Persistent or Nonpersistent.

**Kernel:** The heart of an operating system. The kernel usually includes the functionality of a Hardware Abstraction Layer (HAL). Though applying to any operating system, the term is more often used in reference to Linux than to Windows.

**Linked Clone:** A copy of the original virtual machine that must have access to the parent virtual machine's virtual disk(s). The linked clone stores changes to the virtual disk(s) in a set of files separate from the parent's virtual disk files. See also Full Clone.

**NAS:** See Network Attached Storage.

**Native Execution:** Execution of an application directly on a physical server, as contrasted with running the application in a virtual machine.



**Native System:** A computer running a single operating system, and in which the applications run directly in that operating system.

**Network Address Translation (NAT):** A type of network connection that allows you to connect your virtual machines to an external network when you have only one IP network address, and that address is used by the host computer. If you use NAT, your virtual machine does not have its own IP address on the external network. Instead, a separate private network is set up on the host computer. Your virtual machine gets an address on that private network from the VMware virtual DHCP server. The VMware NAT device passes network data between one or more virtual machines and the external network. It identifies incoming data packets intended for each virtual machine and sends them to the correct destination. See also Host-Only Networking.

**Network-Attached Storage (NAS):** A storage system connected to a computer network. NAS systems are file-based, and often use TCP/IP over Ethernet (although there are numerous other variations). See also Storage Area Network.

**Nonpersistent Disk:** All disk writes issued by software running inside a virtual machine with a nonpersistent virtual disk appear to be written to disk, but are in fact discarded after the session is powered down. As a result, a disk in nonpersistent mode is not modified by activity in the virtual machine. See also Persistent Disk.

**Normal (Windowed) Mode:** A mode of operation in which a guest virtual machine resides within the VMware Workstation console screen. See also Full Screen Mode.

**Pacifica:** A code name for AMD's version of virtualization assist, included in some 64-bit AMD processors. See Secure Virtual Machine.

**Persistent Disk:** All disk writes issued by software running inside a virtual machine are immediately and permanently written to a persistent virtual disk. As a result, a disk in persistent mode behaves like a conventional disk drive on a physical computer. See also Nonpersistent Disk.

**Physical CPU:** A processor within a physical machine. See also Virtual CPU.

**Preallocated Disk:** A type of virtual disk in which all the host disk space for the virtual machine is allocated at the time the virtual disk is created. See also Growable Disk.

**SAN:** See Storage Area Network.

**Secure Virtual Machine (SVM):** AMD's version of virtualization assist, included in some 64-bit AMD processors. See also Virtualization Assist.

**Service Console:** The service console boots the systems and runs support, management, and administration applications.

**Snapshot:** A snapshot preserves the virtual machine just as it was when you took that snapshot — including the state of the data on all the virtual machine's disks and whether the virtual machine was powered on, powered off, or suspended. VMware Workstation lets you take a snapshot of a virtual machine at any time and revert to that snapshot at any time.

**Storage Area Network (SAN):** A storage system connected to a dedicated network designed for storage attachment. SAN systems are usually block-based, and typically use the SCSI command set over a Fibre Channel network (though other command sets and network types exist as well). See also Network-Attached Storage.

**Symmetric Multiprocessor (SMP):** A multiprocessor architecture in which two or more processors are connected to a single pool of shared memory. See also Uniprocessor (UP).



**Template:** A virtual machine that cannot be deleted or added to a team. Setting a virtual machine as a template protects any linked clones or snapshots that depend on the template from being disabled inadvertently.

**Uniprocessor (UP):** A single-processor architecture. See also Symmetric Multiprocessor (SMP).

**Vanderpool:** A code name for Intel's version of virtualization assist, included in some 64-bit Intel processors. See Virtualization Technology.

**Virtual CPU:** A processor within a virtual machine. VMware Workstation 5.5 currently supports up to two virtual CPUs per virtual machine.

**Virtual Disk:** A virtual disk is a file or set of files that appears as a physical disk drive to a guest operating system. These files can be on the host machine or on a remote file system. When you configure a virtual machine with a virtual disk, you can install a new operating system into the disk file without the need to repartition a physical disk or reboot the host.

**Virtual Machine:** A virtualized x86 PC environment in which a guest operating system and associated application software can run. Multiple virtual machines can operate on the same host system concurrently.

**Virtual SMP:** A VMware proprietary technology that supports multiple virtual CPUs in a single virtual machine.

**Virtualization Assist:** A general term for technology included in some 64-bit processors from AMD and Intel that can allow 64-bit operating systems to be run in virtual machines (where supported by VMware Workstation). More information is available in VMware knowledge base article 1901. See also Secure Virtual Machine and Virtualization Technology.

**Virtualization Overhead:** The cost difference between running an application within a virtual machine and running the same application natively. Since running in a virtual machine requires an extra layer of software, there is by necessity an associated cost. This cost may be additional resource utilization or decreased performance.

**Virtualization Technology (VT):** Intel's version of virtualization assist, included in some 64-bit Intel processors. See also Virtualization Assist.

**VMware Tools:** A suite of utilities and drivers that enhances the performance and functionality of your guest operating system. Key features of VMware Tools include some or all of the following, depending on your guest operating system: an SVGA driver, a mouse driver, the VMware Tools control panel, and support for such features as shared folders, shrinking virtual disks, time synchronization with the host, VMware Tools scripts, and connecting and disconnecting devices while the virtual machine is running.

**VMware Virtual Machine Console:** An interface that provides access to one or more virtual machines running on the host. You can view the virtual machines' displays to run programs within them or modify guest operating system settings. In addition, you can change the virtual machines' configurations, install a guest operating system, or choose to run a virtual machine in full screen mode.

---

VMware, Inc., 3145 Porter Drive, Palo Alto, CA 94304

Tel 650-475-5000 Fax 650-475-5001 [www.vmware.com](http://www.vmware.com)

Copyright © 1998-2006 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156 and 6,795,966; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. All other marks and names mentioned herein may be trademarks of their respective companies. Revision: 20060414 Item: WS-ENG-Q106-185

---