Microsoft® Windows Server™ 2003

Performance Tuning Guidelines for Windows Server 2003

Microsoft Corporation

Published: October 2003

Abstract

This document describes important tuning parameters and settings that can result in improved performance for your Microsoft® Windows Server™ 2003 system. Each setting and its potential effect are described to help you make an informed judgment about its relevance to the system, workload, and performance goals.

Contents

Introduction

Microsoft® Windows Server™ 2003 should perform very well out of the box for most customer workloads. However, it is possible to tune the server settings and see incremental performance gains, especially when the nature of the workload will not vary much over time.

The most effective tunings take into account the hardware, the workload, and the performance goals. This document describes important tuning parameters and settings that can result in improved performance. Each setting and its potential effect are described to help you make an informed judgment about its relevance to the system, workload, and performance goals.

Note: Registry settings and tuning parameters may have changed significantly from Microsoft Windows® 2000 Server to Windows Server 2003. Please keep this in mind as you tune your server—using earlier or out-of-date tuning guidelines may produce unexpected results.

As always, care must be taken when manipulating the registry directly. If you must edit the registry, back it up first.

Performance Tuning for Networking

The network architecture covers many components, interfaces, and protocols; Figure 1 illustrates some of them. The sections below discuss tuning guidelines for some of the components for server workloads.

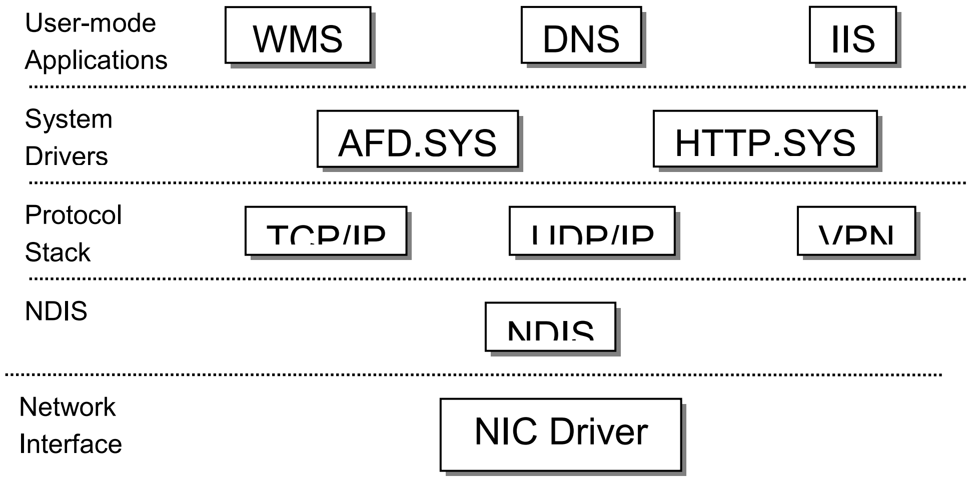

Figure 1   Network Stack Components

The network architecture is layered, and the layers can be broadly divided into:

- **The network driver and NDIS.** These are the lowest layers. NDIS exposes interfaces for the driver below it and for the layers above it; for example, TCP/IP.
- **The protocol stack.** This implements protocols such as TCP/IP and UDP/IP. These layers expose the TDI interface for layers above them.
- **System Drivers.** These are typically TDI clients and expose interfaces to user-mode applications. The WinSock interface is exposed by Afd.sys.
- **User-mode applications.**

Tuning for network-intensive workloads can involve tuning for each of the layers. Some of the tunings are described below.

Choosing a Network Adapter

Network-intensive applications need high-performance network adapters. This section covers some considerations for choosing network adapters.

WHQL Certification

Choose a network adapter with Microsoft Windows Hardware Quality Labs (WHQL) certification.

Offload Capabilities

Offloading tasks can help lower CPU usage on the server, improving overall system performance. The Microsoft TCP/IP transport can offload one or more of the following tasks to a network adapter that has the appropriate task-offload capabilities:

- **Checksum tasks.** The TCP/IP transport can offload the calculation and validation of IP and TCP checksums for sends and receives.
- **IP security tasks.** The TCP/IP transport can offload the calculation and validation of encrypted checksums for authentication headers (AH) and encapsulating security payloads (ESP). The TCP/IP transport can also offload the encryption and decryption of ESP payloads.
- **Segmentation of large TCP packets.** The TCP/IP transport supports large send offload (LSO). With LSO, the TCP/IP transport can offload the segmentation of large TCP packets.
- **Stack offload.** The entire network stack can be offloaded to a network adapter that has the appropriate capabilities.

Interrupt Moderation

Some network adapters are capable of moderating how frequently they interrupt the host processors to indicate network activity (or its completion). Some network adapters are also capable of making such decisions in an adaptive manner, taking into account network and host-processor load. Moderating interrupts can often result in reduction in CPU load on the host, but unless interrupt moderation is performed intelligently and adaptively, the CPU savings may come at the cost of increases in latency.

64-bit Capabilities

Network adapters that are 64-bit capable can perform direct memory access (DMA) operations to and from high physical memory locations (above 4 GB).

Copper and Fiber Network Adapters

Copper network adapters have the same performance as their fiber counterparts, but may be less expensive to purchase. The cost of the transceiver on copper network adapters is lower.

Dual or Quad Port Network Adapters

These network adapters are good for failover scenarios but share a single interrupt among all the ports on the network adapter. Using two single-port network adapters usually yields better performance than using one dual-port network adapter for the same workload.

Tuning the Network Adapter

You can optimize network throughput and resource usage by using network adapter tunings (when available and exposed by the network adapter). Keep in mind that the correct set of tunings depends on the network adapter, workload, host-computer resources, and performance goals.

Enable Offload Features

It is almost always beneficial to turn on network adapter offload features. In some instances, however, the network adapter may not be powerful enough to handle the offload capabilities at high throughput.

For example, enabling LSO can lower the maximum sustainable throughput on some network adapters. However, if the reduced throughput is not expected to be a limitation, offload capabilities should be enabled even for such network adapters. Note that some network adapters require offload features to be enabled for send and receive paths independently.

Network Adapter Resources

Several network adapters allow the configuration of resources by the administrator. Receive buffers and send buffers are among the parameters that may be set. Some network adapters actively manage their resources, and there is no need to set such parameters for these network adapters.

Interrupt Moderation

Some network adapters expose *buffer coalescing* parameters (sometimes separately for send and receive buffers) for control over interrupt moderation. It is important to consider buffer coalescing when the network adapter does not perform adaptive interrupt moderation.

TCP Parameters

TCP parameters that can be adjusted for high throughput scenarios are listed in Table 1.

Table 1. TCP Parameters

Parameter	Description
TCPWindowSize	This value determines the maximum amount of data (in bytes) that can be outstanding on the network at any given time. It can be set to any value from 1 to 65,535 bytes by using the following registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip \Parameters\TcpWindowSize (REG_DWORD)` The default for a gigabit interface is set to approximately 65,535 (rounded down to the nearest multiple of full TCP packets), 16,384 for a 100 Mbps link, and 8,192 for all interfaces of lower speeds (for example, modems), again rounded down. Ideally, this value should be set to the product of end-to-end network bandwidth (in bytes/s) and the round-trip delay (in seconds), also referred to as the bandwidth-delay product. This value should be set according to the amount of TCP data expected to be received by the computer.
Window Scaling	On a link with high bandwidth-delay product (for example, satellite links), there may be a need to increase the window size to above 64 K. For that, you need to enable TCP Options as specified in RFC 1,323 by appropriately setting the following registry entry: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters \Tcp1323Opts (REG_DWORD)` To enable window sizes of greater than 65,535, this registry entry should be set to 1 (one). After this change has been made, the registry entry controlling TCPWindowSize can be set to values larger than 64K (up to 1GB).
MaxHashTableSize	This value determines the size of the hash table holding the state of TCP connections. The default value is 128 multiplied by the square of the number of processors in the system. When a large concurrent connection load is expected on the system, set the following registry entry to a high value to improve the performance of the hash table: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip \Parameters\MaxHashTableSize (REG_DWORD)` The maximum value is 0x10000 (65,536). It is recommended that you consider using the maximum value for large servers which you expect to carry high connection load. Keep in

	mind that the table uses nonpaged pool, so do not set too high a value for the parameter if the server does not have much available nonpaged pool or if you do not anticipate a high-connection load.
NumTcbTablePartitions	By default, the table holding TCP connection states has as many partitions as the square of the number of processors. In most cases, the setting is appropriate and results in lowered contention on the tables. However, the default may be too high for 16 or more processors, and may result in too much CPU usage. In that case, set the following registry entry to a value lower than the square of the number of processors: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip \Parameters\NumTcbTablePartitions (REG_DWORD)`
MaxUserPort	A port is used whenever an active connection is used from a computer. Given the default value of available user mode ports (5,000 for each IP address) and TCP time-wait requirements, it may be necessary to make more ports available on the system. You can set the following registry entry to as high as 0xfffe (65534): `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip \Parameters\MaxUserPort`

Performance Tuning for Storage

The storage architecture covers many components in the driver stack as Figure 2 shows. The sections below discuss tuning guidelines for storage workloads.

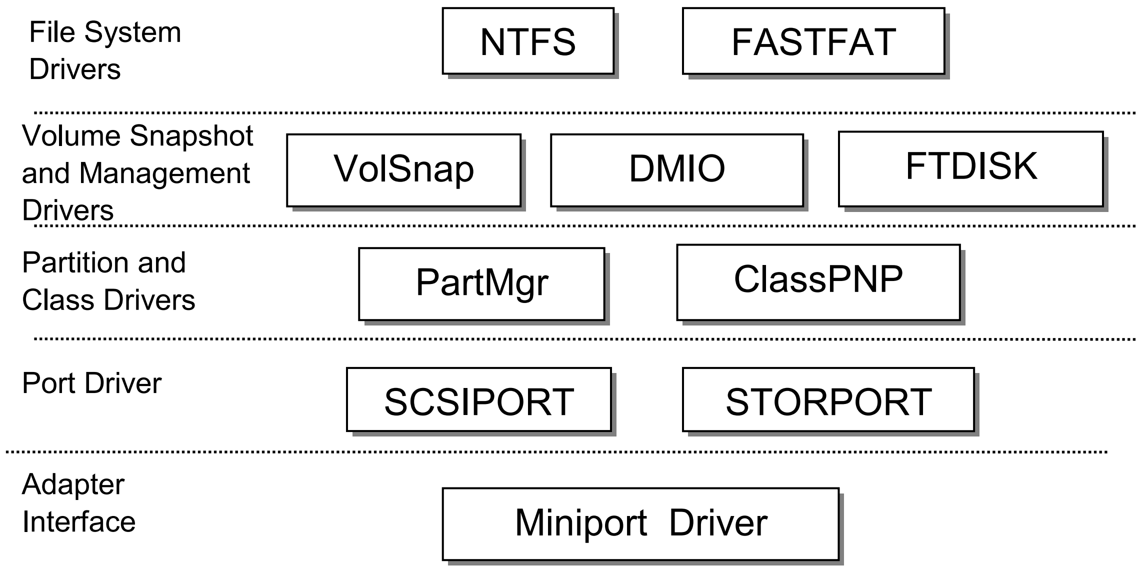

Figure 2 Storage Driver Stack

Choosing Storage

The most important considerations in choosing storage systems are:

- Providing necessary storage space, bandwidth, and latency characteristics for current and future needs.
- Selecting an HW RAID type and backup procedure that provide the required performance and data recovery capabilities.

WHQL Certification

Choose a storage adapter with WHQL certification.

Estimate the Amount of Data to be Stored

When you estimate the amount of data to be stored on a new file server, you need to consider these issues:

- The amount of data currently stored on any file servers that will be consolidated onto the new file server.
- If the file server will be a replica member, the amount of replicated data that will be stored on the new file server.

- The amount of data that you will need to store on the file server in the future.

A general guideline is to plan for faster growth in the future than you experienced in the past. Investigate whether your organization plans to hire a large number of people, whether any groups in your organization are planning large projects that will require extra storage, and so on.

You must also take into account the amount of space used by operating system files, applications, RAID redundancy, log files, and other factors. Table describes some factors that affect file server capacity.

Table 2. Factors That Affect File Server Capacity

Factor	Storage Capacity Required
Operating system files	At least 1.5 GB. To allow space for optional components, future service packs, and other items, plan to allow an additional 3 GB to 5 GB for the operating system volume.
Paging file	1.5 times the amount of RAM by default.
Memory dump	Depending on the memory dump file option that you have chosen, the amount of disk space required can be as large as the amount of physical memory plus 1 MB.
Applications	Varies according to the application, which can include antivirus, backup, and disk quota software, database applications, and optional components such as Recovery Console, Services for Unix, and Services for NetWare.
Log files	Varies according to the application that creates the log file. Some applications allow you to configure a maximum log file size. You must ensure that you have adequate free space to store the log files.
RAID solution	Varies; see Choosing the Raid Level later in this document for more information.
Shadow copies	Ten percent of the volume by default, although increasing this size is recommended.

Storage Array Selection

There are many considerations in choosing a storage array and adapters. The choices include the type of storage arrays being used, including the following options.

Table 3. Options for Storage Array Selection

Option	Descriptoin
Fibre Channel or SCSI	• Fibre Channel allows long glass or copper cables to connect the storage array to the system while providing high bandwidth. • SCSI provides very high bandwidth but has cable length restrictions.
HW RAID capabilities	It is important for the storage controllers to offer HW RAID capabilities. RAID levels 0, 1, and 5 are described in Table 4.
Maximum storage capacity	• Total storage area. • Bandwidth at which storage can be accessed which is determined by the number of physical disks in the array, speed of controllers, type of disk (for example, SCSI or Fibre Channel), HW RAID, and adapters used to connect the storage array to system.

HW RAID Levels

Most storage arrays provide some HW RAID capabilities, including the following RAID options.

Table 4. HW RAID Options

Option	Descriptoin
RAID 0	RAID 0 presents a logical disk that stripe disk accesses over a set of physical disks. • Overall this is the fastest HW RAID configuration. • This is the least expensive RAID configuration, because data is not duplicated. • RAID 0 does not provide additional data recovery mechanisms as does RAID 1 and RAID 5.
RAID 1	RAID 1 presents a logical disk that is mirrored to another disk. • RAID 1 is slower than RAID 0 for write operations, because the data needs to be written to two or more physical disks, and the latency is the slowest of the write operations. • In some cases, RAID 1 can provide faster read operations than RAID 0 because it can read from the least busy physical disk. • RAID 1 is the most expensive in terms of physical disks, because two or more complete copies of the data are stored. • RAID 1 is the fastest in terms of recovery time after a physical disk failure, because the second physical disk is available for immediate use. A new mirror physical disk can be installed while full data access is permitted.
RAID 5	RAID 5 presents a logical disk that has parity information written to other disks as FIgure 3 shows. • RAID 5 uses independent data disks with distributed parity blocks. • RAID 5 is slower then RAID 0, because each logical disk write I/O results in data being written to multiple disks. However, RAID 5 provides additional data recovery capabilities over RAID 0, because data can be reconstructed from the parity. • RAID 5 requires additional time (compared to RAID 1) to recovery from a lost physical disk, because the data on the disk needs to be rebuilt from parity information stored on other disks. • RAID 5 is less expensive than RAID 1, because a full copy of the data is not stored on disk.
Other	Other combinations of RAID exist including RAID 0+1, Raid 10 and Raid 50.

The following figure illustrates RAID 5.

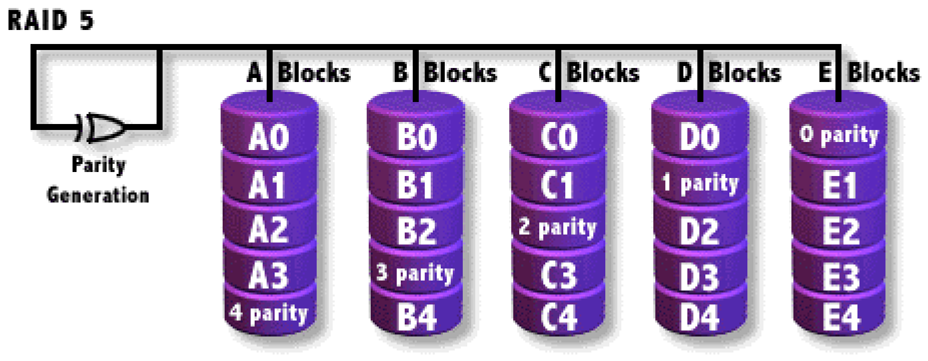

Figure 3 RAID5 Overview

Choosing the RAID Level

Each RAID level is trade-off between the following factors:

- Cost
- Performance

- Availability and reliability

You can determine the best RAID level for your file servers by evaluating the read and write loads of the various data types and then deciding how much you are willing to spend to achieve the performance and availability/reliability that your organization requires. Table describes four common RAID levels, their relative costs, performance, availability and reliability, and their recommended uses.

Table 5. RAID Trade-Offs

	RAID-0 **Striped**	**RAID-1** **Mirrored**	**RAID-5** **Striped with Parity**	**RAID-0+1** **Striped Mirrors**
Minimum number of disks	2	2	3	4
Usable storage capacity	100%	50%	N-1/N where N is the number of disks	50%
Fault tolerance	None. Losing a single disk causes all data on the volume to be lost.	Can lose multiple disks as long as a mirrored pair isn't lost.	Can tolerate the loss of one disk.	Can lose multiple disks as long as a mirrored pair is not lost. Varies according to the number of mirrored pairs in the array. [1]
Read performance	Generally improved by increasing concurrency.	Good read performance	Generally improved by increasing concurrency.	Improvement from increasing concurrency and dual sources for each request.
Write performance	Generally improved by increasing concurrency.	Worse than JBOD (between 20% and 40% for most workloads)	Poor unless full-stripe writes (large requests) Can be as low as ~25% of JBOD (4:1 requests).	Can be better or worse depending on request size, hot spots (static or dynamic), and so on.
Best uses	Temporary data only	Operating system log files	• Operating system • User and shared data • Application files	• Operating system • User and shared data • Application files • Log files

[1]If a disk fails, failure of its mirrored partner prior to replacement will cause data loss. However, the failure of any other member disk does not cause data loss.

If you use more than two disks, RAID 0+1 is almost always a better solution than RAID 1.

When determining the number of disks that should be included in RAID 0, RAID 5, and RAID 0+1 virtual disks, consider the following information:

- Performance increases as you add disks.
- Reliability, in terms of mean time To failure (MTTF) of two disks, decreases as you add disks for RAID 5 or a single disk for RAID 0.
- Usable storage capacity increases as you add disks, but so does cost.
- Stripe unit size. Software solution is fixed at 64 KB. Hardware solutions may range from 4 KB to 1 MB. Ideal stripe unit size maximizes the disk activity without unnecessarily breaking up requests (so that multiple disks are required to service a single request). For example:
    - One stream of sequential requests (large) on JBOD would keep only one disk busy at a time. To keep all disks busy, the stripe unit needs to be equal to 1/N where N is the request size.
    - For N streams of small random requests, if N is greater than the number of disks, and if there are no hotspots, striping will not increase performance. However, if there are hotspots, the stripe unit size needs to maximize the chance that a request will not be split, while minimizing the chance of a hotspot falling entirely within one or two stripe units. You might pick a low multiple of the request size, like 5X or 10X, especially if the requests are on some boundary (for example, 4 KB or 8 KB).
    - For fewer streams than disks, tou need to split the streams so that all disks are kept busy. Interpolate from the previous two examples. For example, if you have 10 disks and 5 streams, split each request in half (use a stripe unit size equal to half the request size).

Determining the Volume Layout

Whenever possible use separate volumes for each data type. For example, use one volume for the operating system and paging space, and one or more volumes for shared user data, applications, and log files.

Place different data types in separate volumes on different virtual disks. Using separate virtual disks is especially important for any data types that create heavy write loads, such as log files, so that a single set of disks (that compose the virtual disk) can be dedicated to handling the disk I/O created by the updates to the log files. Placing the paging file on a separate virtual disk can provide some minor improvements in performance, but typically not enough to make it worth the extra cost.

To gain some performance benefits while minimizing cost, it is often useful to combine different data types in one or more volumes on the same virtual disks. A common method is to store the operating system and paging space on one virtual disk and the user data, applications, and log files in one or more volumes on the remaining virtual disk.

Interrupt Moderation

Some storage adapters are capable of moderating how frequently they interrupt the host processors to indicate disk activity (or its completion). Moderating interrupts can often result in reduction in CPU load on the host, but unless interrupt moderation is performed intelligently; the CPU savings may come at the cost of increases in latency.

Table 6. Options for Interrupt Moderation

Device	Description
64-bit capabilities	Adapters that are 64-bit-capable can perform DMA operations to and

	from high physical memory locations (above 4 GB).
Copper and fiber (glass) adapters	Copper adapters generally have the same performance as their fiber counterparts, and both copper and fiber are available on some fibre channel adapters. Some environments are better suited for either copper or glass adapters.
Dual or quad port SCSI adapters	Some SCSI adapters provide 2 or 4 SCSI buses on a single adapter card. This is often necessary due to SCSI limitations on the number of disks that can be connected to a SCSI bus. Fibre channel disks generally do not have limits on the number of disks connected to an adapter.

Storage Related Parameters

You can adjust the following registry parameters for high throughput scenarios.

CountOperations

This parameter allows you to turn off system and process level I/O counters. This counter affects system and disk counting of disk and network I/O requests. Physical and logical disk counters—in addition to network interface, IP and TCP counters—are not affected by this parameter. It is useful to turn off the process and system counters by using this registry parameter on systems where there is a measurable cost associated with counting I/O at the process and system level but where I/O rates can still be analyzed at the physical, logical, network interface, IP and TCP levels. To turn off the process and system I/O counters, you need to create a registry value (and I/O System key if one doesn't already exist) and set the value to 0 (REG_DWORD) in the following registry entry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Session Manager\I/O System\CountOperations.
```

A reboot is required for this setting to take effect. Process and system counters can be turned on again either by setting CountOperations to 1 or by removing the CountOperations registry entry.

NumberOfRequests

This parameter allows you to specify the number of SRBs created for a given adapter. This improves performance and allows Windows to give more disk requests to a logical disk, which is most useful for HW RAID adapters that have concurrency capabilities since each logical disk consists of multiple physical disks. However, the default setting is often less than optimal for many high-speed HW RAID disk arrays. Overall disk array performance can be improved by setting NumberOfRequests to a value in the range of 32 to 96 (decimal). Set the following registry entry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\MINIPORT_ADAPTER\Parameters
\DeviceN\NumberOfRequests (REG_DWORD)
```

Replace MINIPORT_ADAPTER with the specific adapter name. Make an entry for each device, and in each entry replace *DeviceN* with Device1, Device2, and so forth, depending on the number of devices you are adding. A reboot is required for this setting to take effect. For example, for two Emulex LP9000 miniport adapters whose miniport driver name is lp6nds35, you would create the following registry entries set to 96:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\lp6nds35\Parameters\Device0\NumberOfRequests
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\lp6nds35\Parameters\Device1\NumberOfRequests
```

DontVerifyRandomDrivers

This parameter prevents the driver verifier from randomly verifying drivers for debugging. To disable the driver verifier set a value of 1 (REG_DWORD) for the following registry entry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Session Manager\Memory
Management\DontVerifyRandomDrivers.
```

Performance Tuning for IIS 6.0

Selecting the Right Hardware for Performance

It is important to select the right hardware to satisfy the expected Web load (keeping in mind average load, peak load, capacity, growth plans, and response times). Hardware bottlenecks limit the effectiveness of software tuning. Consider the issues in Table 7 when selecting hardware.

Table 7. Hardware Considerations for IIS 6.0

Issue	Considerations
Number, type, and speed of processors	• Scripts (ASP or ASP.NET) and encryption (SSL) are CPU intensive. Concurrent client load also drives up CPU usage. Consider this when selecting processor speeds and the number of processors in the system. • Larger L2 processor caches provide better performance. • Two CPUs are rarely as effective as one CPU that is twice as fast (as the slower CPUs).
Amount of physical memory (RAM)	Added memory will help as you add more sites, more dynamic content scripts (in particular ASP.NET scripts), and more application pools (or worker processes).
Number, type, and speed of network adapters	The network adapter should not be a bottleneck. Newer network cards can offload some server functions and help performance. For details, see Performance Tuning for Networking earlier in this document.
Type of disk controller, number and capacity of physical disk platters	• IIS caches frequently accessed files in memory. However, files that are not accessed frequently (the "cold set") must be retrieved from the disk when needed. Handling large sites with a high number of requests from the cold set requires good disk performance (a RAID controller connected to a large number of disk platters). • IIS log file sizes and estimated growth under load should be taken into consideration.
Other servers on which the IIS server might depend	• A slow SQL Server may limit the response rate of the IIS server connected to it, even if the IIS server has good hardware components. • Putting both IIS server and SQL Server (or any other CPU-intensive component) on the same computer limits the resources available to each component and affects overall performance.

Operating System Practices

- If possible, do a clean install of the operating system software. Upgrading could leave outdated, unwanted, or sub-optimal registry settings as well as previously installed services and applications that consume resources if automatically started. If another operating system is installed and needs to be kept, install the new operating system on a different partition, otherwise the new installation will overwrite the settings under Program Files\Common Files.

- To reduce disk-access interference, keep the system pagefile, operating system, Web data, ASP template cache, and IIS log on separate physical disks if possible.

- Avoid installing unnecessary services and applications.

Tuning IIS 6.0

IIS 6.0 employs a new process model. A kernel mode HTTP listener (Http.sys) receives and routes HTTP requests (and can even satisfy requests from its response cache). Worker processes register for URL subspaces, and Http.sys routes the request to the appropriate process (or set of processes, in the case of application pools).

Figure 4 shows the difference between the IIS 5.0 and IIS 6.0 process models. IIS 5.0 uses WinSock to accept connections on port 80. Requests are received by the *inetinfo* process, which then either executes the request in-process, or hands it to a *dllhost* process for out-of-process handling (to provide isolation). The response is sent back by the *inetinfo* process.

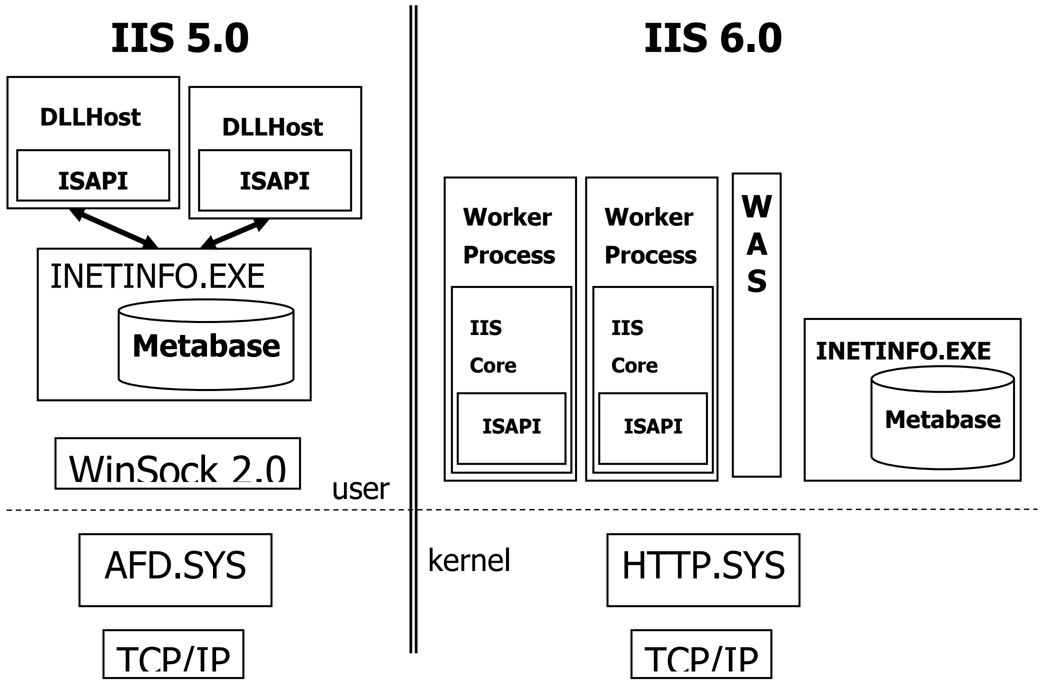

Figure 4   Process Models for IIS 5.0 and IIS 6.0

The IIS 6.0 process relies on the kernel-mode Web driver, Http.sys. In the new model, Http.sys is responsible for connection management and request handling. The request may either be served from the Http.sys cache or handed to a worker process for further handling (see Figure 5). Multiple worker processes may be configured, providing isolation at lower cost.

Http.sys includes a response cache. When a request matches an entry in the response cache, Http.sys sends the cache response directly from kernel-mode. Figure 5 shows the request flow from the network through Http.sys (and possibly up to a worker-process).

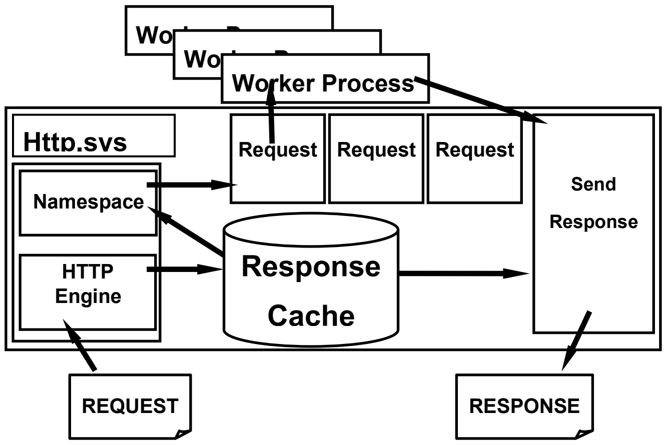

Figure 5   Request Handling in IIS 6.0

Because a Web server has a kernel-mode as well as a user-mode component, both must be tuned for optimal performance. Therefore, tuning IIS 6.0 for a specific workload includes configuring:

- Http.sys (the kernel mode driver) and the associated kernel-mode cache.
- Worker processes and user-mode IIS, including application pool configuration.

Additionally, some tuning parameters that affect performance are discussed in the following sections .

Kernel-mode Tunings

Performance-related Http.sys settings fall into two broad categories: cache management, and connection and request management. All registry settings are stored under the following entry:
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Http\Parameters`

If it is already running, the HTTP service must be stopped and restarted for the changes to take effect.

Cache Management Settings

One of the benefits that Http.sys provides is a kernel-mode cache. If the response is in the kernel-cache, it is possible to satisfy an HTTP request entirely from kernel-mode, which significantly lowers the CPU cost of handling the request. However, the kernel-mode cache of IIS 6.0 is a physical-memory based cache, and the cost of an entry is the memory it occupies.

An entry in the cache is of benefit only when used. However, the entry uses physical memory at all times, whether the entry is in use or not. The usefulness of an item in the cache (the difference being able to serve it from the cache makes) and its cost (physical memory occupied) over the lifetime of the entry need to be evaluated taking into account the resources available (CPU, physical memory) and the workload requirements. Http.sys attempts to keep only useful (actively accessed) items in the cache, but it is possible to increase the performance of the web server by tuning the Http.sys cache for particular workloads.

Below are some useful settings for the Http.sys kernel-mode cache:

- **UriEnableCache.** Default Value: 1. A non-zero value enables the kernel-mode response and fragment cache. For most workloads, the cache should remain enabled. Consider disabling the cache if you expect very low response and fragment cache utilization.
- **UriMaxCacheMegabyteCount.** Default Value: 0. A non-zero value specifies the maximum memory available to the kernel cache. The default value, 0, allows the system to automatically adjust the amount of memory available to the cache. Note that specifying the size only sets the maximum, and the system may not allow the cache to grow to the specified size.
- **UriMaxUriBytes.** Default Value: 262144 bytes (256 KB). This is the maximum size of an entry in the kernel cache. Responses or fragments larger than this will not be cached. If you have enough memory, consider increasing this limit. If memory is limited, and large entries are crowding out smaller ones, it may be help to lower this limit.
- **UriScavengerPeriod.** Default Value: 120 seconds. The Http.sys cache is scanned by a scavenger periodically and entries not accessed between scavenger scans are removed. Setting the scavenger period to a high value reduces the number of scavenger scans. However, the cache memory usage may grow as older, less frequently accessed entries are allowed to stay in the cache. Setting this period to too low a value causes more frequent scavenger scans, and may result in excessive flushes and cache churn.

Request and Connection Management Settings

Http.sys also manages inbound HTTP/HTTPS connections and is the first layer to handle requests on these connections. It uses internal data structures to keep information about connections and requests. Although such data structures can be created (and freed) on demand, it is more CPU-efficient to keep some in reserve in look-aside lists. Keeping such reserves help Http.sys handle fluctuations in load with less CPU usage. Note that load fluctuations are not necessarily the result of fluctuations in externally applied load. Internal optimizations to promote batch processing, and even interrupt moderation may result in load fluctuations and spikes.

The reserves help reduce CPU usage and latency, and increase Web server capacity but increase memory usage. When tuning the request and connection management behavior of Http.sys, it is important to keep in mind the resources available to the server, performance goals, and the characteristics of the workload. Use the following request and connection management settings:

- **MaxConnections.** This value controls the number of concurrent connections Http.sys will allow. Each connection consumes non-paged-pool, a precious and limited resource. The default is determined quite conservatively to limit the amount of non-paged-pool used for connections. On a dedicated web server with ample memory, the value should be set higher if a significant concurrent connection load is expected. A high value may result in increased non-paged-pool usage, so care should be taken to use a value appropriate for the system.
- **IdleConnectionsHighMark, IdleConnectionsLowMark, and IdleListTrimmerPeriod.** These values control the handling of connection structures not currently in use: how many must be available at any time (to handle spikes in connection load), the low and high watermarks for the free list, and the frequency of connection structure trimming and replenishment.

- **RequestBufferLookasideDepth and InternalRequestLookasideDepth** These values control the handling of data structures related to buffer management, and how many are kept in reserve to handle load fluctuations.

User-mode Settings

IIS Registry Settings

The following registry settings are found under this entry:
`HKLM\System\CurrentControlSet\Services\Inetinfo\Parameters\`

- **MaxCachedFileSize** (REG_DWORD) in bytes. Allows file sizes up to the specified size to be cached (default 256 KB). The actual value depends on the number and size of the largest files in the dataset versus the available RAM. Caching large, frequently requested files can reduce CPU usage, disk access, and associated latencies.
- **MemCacheSize** (REG_DWORD) in MB. Limits the IIS user-mode cache size to the specified size (default is adjusted by IIS depending on available memory). Choose the value carefully based on the size of the hot set (set of frequently accessed files) versus the amount of RAM or the IIS process address space, which is normally limited to 2 GB.
- **DisableMemoryCache** (REG_DWORD). Disables the user-mode IIS cache when set to 1 (default is 0). When the cache hit rate is very small you can disable the cache altogether to avoid the overhead associated with the cache code path.
- **MaxPoolThreads** (REG_DWORD). Sets the maximum number of pool threads that can be created per processor (default 4, range unlimited). Each pool thread watches for network requests and then processes them. The MaxPoolThreads count does not include threads that are currently processing ISAPI applications. This parameter should be increased if the CPU shows sub-optimal average usage because all existing threads are busy and there is no available thread to process new requests.
- **PoolThreadLimit** (REG_DWORD). Sets the maximum number of pool threads that can be created in the system (default is four times the number of processors, range unlimited). PoolThreadLimit must be greater than or equal to MaxPoolThreads. Normally PoolThreadLimit = MaxPoolThreads × number of processors. Setting only one of these parameters is sufficient. If both MaxPoolThreads and PoolThreadLimit are specified, the more stringent limit is used.
- **ObjectCacheTTL** (REG_DWORD) in seconds. Controls the length of time that objects are allowed to stay in the IIS user-mode cache without being accessed (default 30 seconds, 0xFFFFFFFF disables the object cache scavenger thread). This parameter can be increased if there is no memory pressure on the system and if the content being served does not change frequently. Lower the parameter if the system is under memory pressure and the user-mode cache is growing. See also ActivityPeriod later in this section.
- **ActivityPeriod** (REG_DWORD) in seconds. Allows a file to be cached only if it was hit repeatedly within the activity period (default 10 seconds, a value of zero will disable this check). This parameter reduces caching overhead caused by caching files that are not accessed frequently You can increase the activity period if the cache is not under heavy churn and there is enough free memory, or you can decrease the activity period if there is a heavy request load on the cache.

- **DataSetCacheSize** (REG_DWORD) default 50. Sets the maximum number of virtual directory entries in the metabase dataset cache. Increase if the number of installed virtual directories exceeds the default. The impact of an under-sized dataset cache is increased latency (accompanied by reduced throughput and low CPU usage) when serving static content.

IIS Metabase

The following settings are found under W3SVC/.

- **AspMaxDiskTemplateCacheFiles**. Allows disk caching of ASP script templates. Compiling the ASP templates is a processor-intensive task. Memory constraints limit the number of templates that can be cached in-memory. Fetching compiled templates from the disk template cache incurs less cost than compiling templates that do not fit into the ASP memory cache. See also AspScriptEngineCacheMax later in this section.
- **AspDiskTemplateCacheDirectory**. If possible, set to a platter not in heavy use (for instance, not shared with the operating system, pagefile, IIS log or other frequently-accessed content). The default directory is "%windir%\system32\inetsrv\Template Disk cache\ASP Compiled Templates".
- **AspScriptEngineCacheMax**. Set to as many script engines as memory limits allow (default 125).
- **AspScriptFileCacheSize**. Set to as many ASP templates as memory limits allow (default 250). See also AspMaxDiskTemplateCacheFiles earlier in this section.
- **AspExecuteInMTA**. Set to 1 (enable) if errors or failures are detected while serving some of the ASP content. This can happen, for example, when hosting multiple isolated sites where each site runs under its own worker process. Errors are typically reported from COM+ in the event viewer. This setting enables the multithreaded apartment model in ASP (the default 0 means disable).
- **AspProcessorThreadMax**. Increase if the current setting (default 25) is not sufficient to handle the load (possibly resulting in errors serving some requests)
- **CentralBinaryLoggingEnabled**. Enable central binary logging by setting this parameter to TRUE. Binary IIS logging reduces CPU usage, disk I/O and disk space usage. Central binary logging is directed to a single file, in binary format, regardless of the number of hosted sites. Parsing binary-format logs requires a post-processing tool.

IIS Worker Process Options (IIS Admin UI, Application Pool Properties)

The options for recycling IIS worker processes under the IIS Admin UI provide practical solutions to acute situations or events without the need for administrator intervention, service reset, or even computer reset. Such situations and events include memory leaks, increasing memory load or non-responsive or idle worker processes. Under normal conditions, recycling options may not be needed and can be turned off (or the system can be configured to recycle very infrequently). In the following sections, bold names are per-app-pool variables. When using scripts to set these variables, use the path /LM/W3SVC/AppPools/*n*, where *n* is the application pool index.

There are three options, as the following tables describe:

- **Recycling options.** These are found on the **Recycling** tab.
- **Performance options.** These are found on the **Performance** tab.
- **Worker process health monitoring options.** These are found on the **Health** tab.

Table 8. Recycling Options

Parameter	Description
PeriodicRestartRequests, DWORD, option disabled by default, default value 35000	Periodic recycling based on time
PeriodicRestartRequests, DWORD, option disabled by default, default value 35000	Periodic recycling based on the (cumulative) number of requests
PeriodicRestartSchedule, MULTISZ, disabled by default, empty string value	Recycling at given time settings
• (**PeriodicRestartMemory,** DWORD, default value 512 MB • **PeriodicRestartPrivateMemory**, DWORD, default value 192 MB	Memory-based recycling (disabled by default) allows recycling of a worker process if it has reached either: • A maximum amount of virtual memory • A maximum amount of used memory Under continuously growing memory pressure it could be helpful to allow frequent recycling of worker processes based on a tight-memory criterion, using either or both of these parameters.

Table 9. Performance Options

Parameter	Description
IdleTimeout, DWORD, in minutes, default 20	Shut down a worker process after being idle for more than a specified amount of time. This can save some resources on limited-memory systems but it is not recommended in situations that will require frequent spawning of new worker processes under heavy CPU load, because of the overhead associated with process creation.
AppPoolQueueLength, DWORD, default 2000	Limit the kernel request queue length per App-Pool. Requests consume paged-pool, and this limit should be lowered under high demand for paged-pool. Exceeding the designated length will cause the server to reject the request with a non-customizable error 503.
CpuAccounting, BOOLEAN, disabled (0) by default, 1 for enabled	Monitor CPU usage. You can set the maximum CPU use by percentage (CpuLimit, DWORD, default 0) and the refresh period to monitor it in minutes (CpuResetInterval, DWORD, default 0). The options upon reaching the CPU usage limit are to take no action (but write an event to the event log) or to shut down the worker process (CPUAction, DWORD, default 0 for "No action", maximum 1 for "Shutdown Worker Process").
MaxProcesses, default: 1 worker process to handle all requests	You can control the total number of worker processes in Web Garden mode of operation. In Web Garden mode, several worker processes handle the request load under a single application pool. There is no pre-assignment of worker processes to Web sites via different app-pools. In some cases one worker process is not enough to handle the load (indicated by poor CPU usage and long response times) and increasing the number of worker processes may improve throughput and CPU usage. One case where Web Garden mode may be considered is with hosting multiple sites. Multiple worker processes can also offer more reliability in case of an incidental crash of one of them, with little chance of total service disruption. Web garden mode is easier

	to set up and control than multiple pre-assigned application pools.

Table 10. Health Options

Parameter	Description
PingingEnabled, BOOLEAN, default 1 **PingInterval**, DWORD, default 30 seconds	Pinging worker processes (PingingEnabled) periodically (PingInterval). If not responding, the worker process is considered to be in a bad state and IIS will attempt to terminate it and spawn another.
RapidFailProtection, BOOLEAN, default **RapidFailProtectionMaxCrashes**, DWORD, default 5 failures **RapidFailProtectionInterval**, DWORD, default 5 minutes	Control a rapid failure rate situation (RapidFailProtection) by setting the maximum number of failures allowed (RapidFailProtectionMaxCrashes) within a given time span (RapidFailProtectionInterval). If such a rate of failures is recorded the application pool will be disabled, and will write related event log messages.
StartupTimeLimit, DWORD, default 90 seconds	Control the worker process startup limit period before considering it a failure
ShutdownTimeLimit, DWORD, default 90 seconds	Control the worker process shutdown limit period before considering it nonresponsive.

Secure Sockets Layer Tuning Parameters

Secure Sockets Layer (SSL) use imposes extra CPU cost. The most expensive component of SSL is the session establishment cost (involving a full handshake), then reconnection cost and encryption/decryption cost. For better SSL performance, do the following:

- Enable keep-alives for SSL sessions. This eliminates the session establishment costs.
- Reuse sessions when appropriate (especially with non-keep-alive traffic).
- Note that larger keys provide more security but also use more CPU time.
- Note that not all components of your page may need to be encrypted. However, mixing plain HTTP and HTTPS may result in a pop-up warning on the client-browser that not all content on the page is secure.

ISAPI

No special tuning parameters are needed for ISAPI. If writing a private ISAPI extension, make sure to code it efficiently for performance and resource use. See also Other Issues Affecting IIS Performance later in this document.

Managed Code Tuning Parameters

- Make sure to precompile all scripts. This can be accomplished by calling one .NET script in each directory. Reset IIS after compilation is complete. Recompile after changes to Machine.config, Web.config or any .aspx script.
- If session state is not needed, make sure to turn it off in each page.
- When running multiple hosts containing ASP.NET scripts in isolated mode (one application pool per site) monitor the memory usage. Make sure the IIS server has enough RAM for the expected number of

concurrently running application pools. Consider using multiple app-domains in place of multiple isolated processes.

Other Issues Affecting IIS Performance

- **Installing Non Cache-Aware Filters.** The installation of a filter that is non-HTTP-cache-aware will cause IIS to disable caching altogether, resulting in a poor performance. Old ISAPI filters (written before IIS 6.0) can cause this behavior. Filters that can operate with the HTTP cache can be marked cache-aware in the metabase.
- **CGI Requests.** Use of CGI applications for serving requests is not recommended under IIS for performance reasons. Frequent creation (and deletion) of CGI processes involves significant overhead. Better alternatives include use of ISAPI application and ASP or ASP.NET scripts. Isolation is available for each of these options.

NTFS File System Setting

Under `HKLM\System\CurrentControlSet\Control\FileSystem\` is **NtfsDisableLastAccessUpdate** (REG_DWORD) 1.

This system-global switch reduces disk I/O load and latencies by disabling the updating of the date and time stamp for the last file or directory access. This key needs to be added; it does not exist by default. Disabling the updates is effective when used with large data sets (or a large number of hosts) containing thousands of directories. It is recommended that you use IIS logging instead if you maintain this information for Web administration only.

Warning: Some applications such as incremental backup utilities rely on this update information and cease to function properly without it.

Tcpip.sys Performance Settings for IIS

See Performance Tuning for Networking earlier in this document.

Network Adapter Tuning and Binding for IIS

- Make sure all network adapter settings are optimal.
- Bind each network adapter to a CPU (the method to use depends on the number of network adapters, the number of CPUs and the number of ports per network adapter).

See also Performance Tuning for Networking earlier in this document.

Performance Tuning for File Servers

General Considerations

It is important to select the right hardware to satisfy the expected file server load, keeping in mind average load, peak load, capacity, growth plans, and response times. Hardware bottlenecks will limit the effectiveness of software tuning.

Consider the following issues when selecting hardware and setting up the operating system.

Table 11. Selecting the Right Hardware for Performance

Issue	Recommendation
Number, type, and speed of processors	• Larger L2 processor caches will provide better performance. • Two CPUs will not be as fast as one CPU that is twice as fast.
Amount of physical memory (RAM)	When your computer is running low on memory and more is needed immediately, Windows Server 2003 uses hard drive space to simulate system RAM. This is known as virtual memory, and is often called the paging file. • Try to avoid having a pagefile on the same drive as the operating system files. • Avoid putting a pagefile on a fault-tolerant drive, such as a mirrored volume or a RAID-5 volume. Pagefiles don't need fault-tolerance, and some fault-tolerant systems suffer from slow data writes because they write data to multiple locations. • Don't place multiple pagefiles on different partitions on the same physical disk drive.
Number, type, and speed of network adapters	• The network adapter should not be a bottleneck. Newer network adapters can offload some server functions and help performance. • Make sure all network adapter settings are optimal. • Bind each network adapter to a CPU (the method depends on the number of network adapters, the number of CPU's and the number of ports per network adapter). • For details, see Performance Tuning for Networking earlier in this document.
Type of disk controller, number of physical disks and their overall capacity	• File Servers cache frequently accessed files in memory. However, files that are not accessed frequently must come from disk. Handling large amounts of data with a high number of requests to a high number of files require good disk performance (RAID controller connected to a large number of disks). • Keep the system page file, the operating system and the data on separate physical disks if possible. • Make sure the allocation unit size is appropriate for the size of the volume.

Server Message Block Server Model

The Server Message Block (SMB) model consists of two entities, the client and the server.

The client receives requests through the redirector (Rdbss.sys and SMB mini-redirector Mrxsmb.sys) for files located on the server. It uses the SMB protocol to send its request through TCP/IP.

The server receives multiple requests from the clients through TCP/IP and routes the requests to the local file system (Ntfs.sys), which accesses the storage stack.

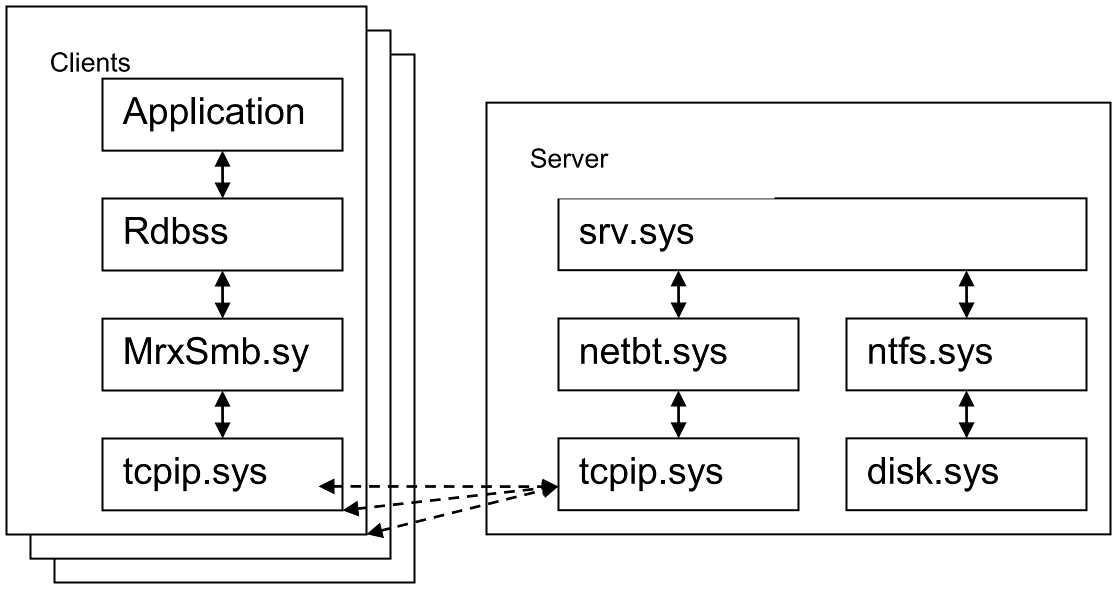

Figure 6   The SMB Model for Client-Server Communication

General Tuning Parameters for File Servers

The following registry tuning parameters could affect performance of file servers.

PagedPoolSize

`HKLM\System\CurrentControlSet\Control\SessionManager\MemoryManagement\ (REG_DWORD)`

File cache space and paged pool space share a common area in system virtual address . Limiting the paged pool allows for a larger system cache, which causes more content to be cached and allows faster serving of files.

NtfsDisable8dot3NameCreation

`HKLM\System\CurrentControlSet\Control\FileSystem\ (REG_DWORD)`

Default is 0. This parameter determines whether NTFS generates a short name in the 8.3 (DOS) naming convention for long file names and for file names that contain characters from the extended character set. If the value of this entry is 0, files can potentially have two names: the name that the user specifies and the short name that NTFS generates. If the name the user specifies conforms to the 8.3 naming convention, NTFS does not generate a short name.

Changing this value does not change the contents of a file, but it avoids the short-name attribute creation for the file, also changing the way NTFS displays and manages the file.

Disablelastaccess

```
HKLM\System\CurrentControlSet\Control\FileSystem\. (REG_DWORD)
```

By default, this registry key is not created.

If you have an NTFS volume with a high number of folders or files, and a program is running that briefly accesses each of these in turn, the I/O bandwidth used to generate the Last Access Time updates can be a significant percentage of the overall I/O bandwidth. To increase the speed of access to a folder or file, you can set disablelastaccess to disable updating the Last Access Time. After you use this command and restart the computer, the Last Access Time is no longer updated. If you create a new file, the Last Access Time remains the same as the File Creation Time.

NumTcbTablePartitions

```
HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\. (REG_DWORD)
```

By default this key is not created.

This parameter controls the number of TCB table partitions. The TCB table can be partitioned to improve scalability on multiprocessor systems by reducing contention on the TCB table.

TcpAckFrequency

Note: TcpAckFrequency applies only to Windows Server 2003. The recommended setting for TcpAckFrequency is between one-third and one-half of TcpWindowSize.

For Gigabit cards:

```
HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces
```

For each Gigabit adapter, add:

```
TcpAckFrequency (REG_DWORD) = 13 (decimal)
```

By default this entry is not in the registry. If only acking data and not any control packets, ack once every 13 packets, instead of the default of two. This helps reducing packet processing costs for the Network Stack, in the case of large writes (uploads) from the client into the server.

For FastEthernet cards:

```
HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces
```

For each FastEthernet adapter, add:

```
TcpAckFrequency (REG_DWORD) = 5 (decimal)
```

By default this entry is not in the registry. If only acking data and not any control packets, ack once every five packets, instead of the default of two. This helps reducing packet processing costs for the Network Stack, in the case of large writes (uploads) from the client into the server.

Interrupt Affinity

Interrupt affinity means binding of interrupts from a specific device to specific processor(s) in a multiprocessor server. This enforces running the ISR and DPC routines on the said processor(s).

Because network connections and file server sessions all stay on the same network adapter, binding interrupts from the network adapter to a processor allows for processing incoming packets (SMB requests, data) on a specific set of processors, improving locality and scalability. You cannot configure affinity on single-processor computers.

The Interrupt-Affinity Filter (IntFiltr) tool allows you to change the CPU-affinity of the interrupts in a system.

Using this utility, you can direct any device's interrupts to a specific processor or set of processors (as opposed to always sending interrupts to any of the CPUs in the system). Note that different devices can have different interrupt-affinity settings. This utility will work on any server running Windows Server 2003, regardless of what processor or interrupt controller is used.

General Tuning Parameters for Client Computers

DormantFileLimit

`HKLM\system\CurrentControlSet\Services\lanmanworkstation\parameters\  (REG_DWORD)`

By default this registry key is not created. (Windows XP client computers only.)

Specifies the maximum number of files that should be left open on a share after the application has closed the file.

ScavengerTimeLimit

`HKLM\system\CurrentControlSet\Services\lanmanworkstation\parameters\ (REG_DWORD)`

Windows XP client computers only.

The amount of time in seconds the redirector waits before it starts scavenging dormant file handles (cached file handles that are not currently used by any application).

DisableByteRangeLockingOnReadOnlyFiles

`HKLM\System\CurrentControlSet\Services\LanmanWorkStation\Parameters\ (REG_DWORD)`

Windows XP client computers only.

Some distributed applications that lock portions of a read-only file as synchronization across clients require that file-handle caching and collapsing behavior be off for all read-only files. This parameter can be set if such applications will not be run on the system and collapsing behavior can be enabled on the client computer.

TcpAckFrequency

Note: TcpAckFrequency applies only to .XP Clients . The recommended setting for TcpAckFrequency is between one-third and one-half of TcpWindowSize.

For Gigabit cards:

`HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces`

For each Gigabit adapter, add:

`TcpAckFrequency (REG_DWORD) = 13 (decimal)`

By default this entry is not in the registry.

If only acking data and not any control packets, ack once every 13 packets, instead of the default of 2. This helps reducing packet processing costs for the network stack in the case of large writes (uploads) from the client into the server.

For FastEthernet cards:

```
HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces
```

For each FastEthernet adapter, add:

```
TcpAckFrequency (REG_DWORD) = 5 (decimal)
```

By default this entry is not in the registry. If only acking data and not any control packets, ack once every 5 packets, instead of the default of 2. This helps reducing packet processing costs for the network stack in the case of large writes (uploads) from the client into the server.

Performance Tuning for Active Directory

Large Active Directory® environments have a few special tuning requirements.

Using the /3GB Switch in the Boot.ini file

On server computers, a large quantity of memory is helpful in reducing disk I/O activity. Use of the /3GB switch gives x86 servers more user mode virtual space and allows Active Directory to improve its caching.

Windows 2000 includes two settings:

- Using the /3GB switch allows the main Active Directory cache a maximum of 1024 MB.
- Without the /3GB switch, the main Active Directory cache is limited to 512 MB.

For Windows Server 2003, the Active Directory cache is allowed to grow more freely but remains limited by virtual address space.

Turning Off Signing and Sealing

Client computers running Windows XP with Service Pack 1 (SP1) and higher, and servers running Windows Server 2003 are capable of signing and sealing for improved security, and this is enabled by default. Windows 2000 clients do not enable signing and sealing by default, although Windows 2000 with Service Pack 3 (SP3) has the option to turn it on. Production environments with a secured network do not require this setting to be enabled. The Windows Server 2003 family of operating systems provides an option for disabling signing and sealing. You can find this setting at:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ldap\ ldapclientintegrity = REG_DWORD 0x0
```

Benchmarking Web Workloads (WebBench)

Consider using the following guidelines for benchmarking Web workloads

- Isolate the IIS server and other related computers from corporate network traffic.
- Allow sufficient warm-up time to get to a steady state.
- Synchronize client clocks with the IIS server clock to ensure proper benchmarking of time-dependent requests.
- For best performance, turn all recycling, performance, and power options off in the IIS Admin UI, unless you encounter an acute situation where these options may help. For more information, see Performance Tuning for IIS 6.0 earlier in this document.
- If using SSL, select a reasonable and consistent key size.

WebBench 4.1 provides a way to measure the performance of Web servers. WebBench uses client computers to simulate Web browsers. However, unlike actual browsers, the clients don't display the files that the server sends in response to their requests. Instead, when a client receives a response from the server, it records the information associated with the response and then immediately sends another request to the server.

The following three tables list high-end and low-end server settings and client computer tuning parameters.

Table 12. High-End Server Settings

Type	Setting
IIS settings	• **Registry** (under HKLM\System\CurrentControlSet/Services\Inetinfo\Parameters\) • **MaxCachedFileSize** (REG_DWORD) 1048576 • **IIS Metabase** (under W3SVC/) • Use central binary logging by setting **CentralBinaryLoggingEnabled** = TRUE • SSL tuning parameters: Key size 1024 bytes. For competitive benchmarking, use the same key size for all servers.
Http.sys settings	• **Registry** (under HKLM\System\CurrentControlSet/Services\HTTP\Parameters\) **UriMaxUriBytes** (REG_DWORD) 1048576 (largest file in the set).
NTFS File System setting	• **Registry** (under HKLM\System\CurrentControlSet\Control\FileSystem\) NtfsDisableLastAccessUpdate (REG_DWORD) 1
TCPIP.SYS performance settings for IIS	• **Registry** (under HKLM\System\CurrentControlSet\Services\tcpip\parameters\) **MaxHashTableSize** (REG_DWORD) 0xffff See also Performance Tuning for Networking earlier in this document.
Network adapter tuning and binding for IIS	• Each network adapter bounded to a CPU. See also Performance Tuning for Networking earlier in this document.

Characteristics of low-end server settings include the following:

- Single processor, single network adapter.
- Limited physical memory—at least 256 MB; typically 512 MB RAM.

- Paging activity expected.
- Not recommended in the case of large number of ASP files or for memory-heavy dynamic content.

Table 13. Low-End Server Settings

Type	Setting
IIS settings	• **Registry** (under HKLM\System\CurrentControlSet/Services\Inetinfo\Parameters\ ) **MaxCachedFileSize** (REG_DWORD) 1048576 **MemCacheSize** (REG_DWORD) 10 • **IIS Metabase** (under W3SVC/) Use central binary logging by setting **CentralBinaryLoggingEnabled** = TRUE
Http.sys settings	• **Registry** (under HKLM\System\CurrentControlSet\Services\http\parameters\) **UriMaxUriBytes** (REG_DWORD) 1048576 **RequestBufferLookasideDepth** (REG_DWORD) 256 **InternalRequestLookasideDepth** (REG_DWORD) 256 **LargeMemMegabytes** (REG_DWORD) 150
NTFS file system setting	• **Registry** (under HKLM\System\CurrentControlSet\Control\FileSystem\) **NtfsDisableLastAccessUpdate** (REG_DWORD) 1

Table 14. Client Computer Tuning Parameters

Type	Setting
My Computer Performance Settings	• Processor scheduling optimized for Programs • Memory usage optimized for Programs
TCPIP.SYS performance settings for IIS	• **Registry** (under HKLM\System\CurrentControlSet\Services\tcpip\parameters\) **MaxUserPort** (REG_DWORD) 0xfffe **MaxHashTableSize** (REG_DWORD) 0xffff **TcpWindowSize** (REG_DWORD) 65536 (make the registry change on clients equipped with 100 BaseT Ethernet network adapters) See also Performance Tuning for Networking in this document.

Benchmarking File Server Workload (NetBench)

NetBench 7.02 is a eTesting Labs benchmark program that lets you measure the performance of file servers as they handle network file requests from clients. NetBench provides you with an overall I/O throughput score and average response time for your server and individual scores for the client computers. You can use these scores to measure, analyze, and predict how well your server can handle file requests from clients. The data volumes are always formatted between tests to flush and clean up the working set to ensure a fresh start. For improved performance and scalability, it is recommended that client data be partitioned over multiple data volumes.

Registry Tuning Parameters for NetBench on Windows Server 2003

Key	Setting
HKLM\System\CurrentControlSet\Control\SessionManager\MemoryManagement\	**PagedPoolSize** = 192000000 (decimal) (default=0)
HKLM\System\CurrentControlSet\Control\FileSystem\	**NtfsDisable8dot3NameCreation** = 1 (default is 0) Add Disablelastaccess = 1 By default this registry key is not created.
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\	Add **NumTcbTablePartitions** = 8 By default this registry key is not created.
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\	Add **TcpAckFrequency** (REG_DWORD) = 13 (decimal) for each Gigabit network adapter. By default this registry key is not created. For FastEthernet adapters set this parameter to 5.

Registry Tuning Parameters for NetBench on Client Computers

Key	Setting
HKLM\System\CurrentControlSet\Services\LanmanWorkStation\Parameters\	**DisableByteRangeLockingOnReadOnlyFiles** = 1; Windows XP client computers
HKLM\system\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\	**Add TcpAckFrequency** = 13 (decimal) for each Gigabit network adapter. By default this registry key is not created. For FastEthernet adapters set this parameter to 5.
HKLM\system\CurrentControlSet\Services\lanmanworkstation\parameters\	Add **DormantFileLimit** = 100 (decimal). By default this registry key is not created; Windows XP client computers.
HKLM\System\CurrentControlSet\Services\lanmanworkstation\parameters\	**ScavengerTimeLimit** = 100 (decimal); Windows XP client computers.

Benchmarking Active Directory Workload (DirectoryMark)

The following tunings are useful for benchmarking the DirectoryMark Workload. DirectoryMark tests are best run from a powerful client to a big server. This allows the operator to start large numbers of threads, and still receive central data reports. This setup requires a Gigabit network adaptor, approximately equal-powered clients and servers, and a server with at least 2 GB of memory.

Add Index for Description Attribute (Server)

Use schema editor to add an index for the description attribute, which is used in the DirectoryMark Addressing and Messaging Search Mixes.

Turn Off Auto Defragmenter

On a server, Auto Defragmenter starts 15 minutes after startup, runs for an hour, and then restarts every 12 hours. Benchmarking environments require reproducible results, so it is recommended that Auto Defragmenter be turned off to avoid any possible interference with a running benchmark. The defragmenter pass can be seen in the event logs if the Auto Defragmenter remains enabled.

The following registry parameter is used for turning off Auto Defragmenter:

```
HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters\DSA Heuristics = REG_SZ 000001
```

Increase MaxUserPorts and TcpWindowSize in TCP/IP

Heavy use of LDAP Binds requires extensive use of dynamic ports. TCP is required on server and client computers to hold these ports open for a few minutes, thus requiring more MaxUserPorts available than are actually used.

You can adjust the following registry parameters:

```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort = REG_DWORD 0xfffe
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize = REG_DWORD
0xffff
```

Benchmarking Networking Workloads (Ttcp, Chariot)

Tuning for NTttcp

NTttcp is a Winsock–based port of ttcp to Windows. It helps measure network driver performance and throughput on different network topologies and hardware setups. It provides the customer with a multithreaded, asynchronous performance benchmark for measuring achievable data transfer rate on an existing network setup.

Options:

- A single thread should suffice for optimal throughput.
- Multiple threads are needed only in the case of single to many clients.
- Posting enough user receive buffers (using the “-a” option) alleviates TCP copying.
- You should not excessively post user receive buffers, because the first ones posted would return before you have the need to use other buffers.
- It’s best to bind each set of threads to a processor (second delimited parameter in “-m” option).
- Each thread creates a socket that connects (listens) on a different port.

Table 15. Example Syntax for NTttcpSender and Receiver

Syntax	Details
Example Syntax for a Sender NTttcps –m 1,0,10.1.2.3 –a 2	• Single thread • Bound to CPU 0 • Connecting to computer with IP 10.1.2.3 • Posting two send overlapped buffers • Default buffer size: 64 KB • Default buffer number: 20 KB
Example Syntax for a Receiver NTttcpr –m 1,0,10.1.2.3 –a 6 –t 1000	• Single thread • Bound to CPU 0 • Connecting to computer with IP 10.1.2.3 • Posting two send overlapped buffers • Default buffer size: 64 KB • Default buffer number: 20 KB

Network Adapter

Make sure you enable all offloading features.

TCP

Set **TcpWindowSize** to something larger than the default value for Gigabit Ethernet (64 KB) only if you have a large bandwidth-delay product.

For example, using the Intel MT Gigabit card on a LAN, you would leave all network adapter and TCP settings at their default values for NTttcp.

- The Intel MT network adapter offloads LSO and checksum (send as well as receive) by default.
- The Intel MT network adapter adaptively manages its resources and you will not need to change any network adapter resource values.
- *Coalesce Buffers* is not exposed, but the default interrupt moderation scheme works well.

Tuning for Chariot

Chariot is a networking workload generator from NetIQ. It stresses the network to help predict networked application performance.

The High_Performance_Throughput script workload of Chariot may be used to simulate the NTttcp workload. The tuning considerations for this workload would the same as those for NTttcp.

Related Links

See the following resources for further information:

- Transaction Processing Performance Council Web site at www.tpc.org.
- Lab Report: Windows Server 2003 Outperforms Predecessors at http://www.microsoft.com/windowsserver2003/evaluation/performance/etest.mspx.
- Performance and Scalability on the Windows Server 2003 Web site at http://www.microsoft.com/windowsserver2003/evaluation/performance/default.mspx.

For the latest information about Windows Server 2003, see the Windows Server 2003 Web site at http://www.microsoft.com/windowsserver2003.