[$] ISA Server 2004 Performance Best Practices

Microsoft Internet Security and Acceleration Server 2004

[$] ISA Server 2004 Performance Best Practices

Contents

Introduction

Microsoft® Internet Security and Acceleration (ISA) Server 2004 provides controlled secure access between networks, and serves as a Web caching proxy providing fast Web response and offload capabilities. Its multi-layered architecture and advanced policy engine provide granular control of the balance between the level of security you need and the resources that are required. As an edge server connecting many networks, ISA Server handles large amounts of traffic compared to other servers in an organization. For this reason, it is built for high performance. This article provides guidelines for deploying ISA Server with best performance and adequate capacity.

Executive Summary

In most cases, the available network bandwidth and especially that of the Internet link can be secured by ISA Server running on available entry-level hardware. A typical default deployment of ISA Server securing outbound Web access (HTTP traffic) requires the following hardware configurations for various Internet links. (For details, see Web Proxy Scenarios.)

Internet link bandwidth	**Up to 5 T1** 7.5 megabits per second (Mbps)	**Up to 25 Mbps**	**Up to T3** 45 Mbps
Processors	1	1	2
Processor type	Pentium III 550 megahertz (MHz) or higher	Pentium 4 2.0–3.0 gigahertz (GHz)	Xeon 2.0–3.0 GHz
Memory	256 megabytes (MB)	512 MB	1 gigabyte (GB)
Disk space	150 MB	2.5 GB	5 GB
Network adapters	10/100 Mbps	10/100 Mbps	100/1000 Mbps

Concurrent virtual private network (VPN) remote access connections	150	700	850

Using transport layer stateful filtering instead of Web Proxy filtering improves CPU utilization for the same traffic patterns by a factor of 10. Both stateful filtering and application filtering can be used in parallel to provide granular control over performance.

Planning ISA Server Capacity

Learning your capacity requirements is the first step in determining the necessary resources for an ISA Server deployment. To do this, there are several cases for a broad range of deployments. In general, you are likely to have the following metrics:

- The available and actual bandwidths on every network that is linked to an ISA Server computer.
- The number of users in your organization.
- Various application level metrics. For example, the average mailbox size in a mail server.

The most important metric for ISA Server capacity is the actual network bandwidths, because they usually represent your true capacity needs. In many cases, network bandwidth, and in particular that of the Internet link, can determine ISA Server capacity.

The number of users is less indicative of your capacity needs because users have different usage patterns, depending on their needs and your organization's network policy. In some cases, the number of users as well as application level metrics may prove useful for estimating network traffic.

All ISA Server capacity planning cases are in one of the following categories:

- All network bandwidth can be served by a single entry-level ISA Server computer.
- Network bandwidth is larger than what any single computer can serve, and ISA Server is used for securing enterprise scale applications.

The following sections describe these cases in more detail.

Single Entry-Level Computer

In most situations, a single computer has enough processing power to secure traffic going through standard Internet links. According to market research reports on Internet usage, most corporate Internet link bandwidths are between

2 to 20 Mbps. This indicates that an entry-level computer with a single or dual processor will suffice for most ISA Server deployments.

According to outbound firewall test results, ISA Server running on a single Pentium 4 2.4-GHz processor can provide a throughput of approximately 25 Mbps at 75 percent CPU utilization. This means that for each T1 Internet link (1.5 Mbps), the firewall service will utilize only 4.5 percent CPU. Dual Xeon 2.4-GHz processors can provide a throughput of approximately 45 Mbps (T3) at 75 percent CPU, or 2.5 percent CPU for every T1.

A single entry-level computer also works in branch offices that connect to corporate resources through independent wide area network (WAN) Internet links with the bandwidth limits described in the preceding paragraph.

Enterprise Scale

For large enterprise-scale sites with over 500 users, the situation is more complex. This case requires more elaborate planning, because Internet bandwidth is large enough to shift the performance bottleneck to the system's CPU resource.

Internet connection bandwidth imposes a limit on the number of computers that can fully utilize the connection, and this maximum may be sufficient for most capacity estimations. Initially, planning for maximum network capacity may be conservative, because capacity requirements often increase over time. To accommodate future growth, you should also plan for processing power upgrades. For a description of hardware scaling techniques, their performance characteristics, and other scaling benefits, see Scaling Out ISA Server.

Performance Tuning Guidelines

After you decide which capacity case fits your needs, your next task is to tune it for best performance. For ISA Server on an enterprise scale, this means designing adequate hardware resources to make the system depend on its CPU power as the source for a possible bottleneck. For a single entry-level ISA Server computer, Internet bandwidth is the source for a possible bottleneck, and not the processor you choose.

Tuning Hardware for Maximum CPU Utilization

ISA Server capacity depends on CPU, memory, network, and disk hardware resources. Each resource has a capacity limit, and as long as all resources are consumed below their limit, the system as a whole functions properly, fulfilling its performance objectives. Performance drops considerably when one of these

limits is reached, causing a bottleneck. In this case, the system is said to be *bound* on that resource. Each bottleneck has its symptoms in overall system performance that can help detect the resource that has inadequate capacity. After a bottleneck is discovered, it can be removed by adding more capacity to the resource that has inadequate capacity.

From a cost perspective, it is most efficient to design a system to be bound on CPU resources. This is because it is the most expensive resource to upgrade. Other resource shortages are less expensive to fix: add another disk, add another network adapter, or increase memory. We recommend that you tune the system's hardware to maximize CPU utilization. Make sure that a system will have no performance bottlenecks before reaching full CPU utilization. If the CPU power can sustain the expected load, the bottleneck will never occur. To do this, all other resources must have adequate capacity. The following sections describe how to design a CPU-maximized system with adequate capacity in each resource, how to monitor each resource, and how to troubleshoot a bottleneck in each resource.

Determining CPU and System Architecture Capacity

Like most server applications serving numerous client requests, ISA Server performance also benefits from higher CPU speed, larger processor cache, and improved system architecture:

- **CPU speed.** As in most applications, ISA Server benefits from faster CPUs. However, increasing CPU speed does not ensure a linear increase in performance. Due to the large and frequent memory access effect, increasing CPU speed may cause more wasted idle memory when waiting for CPU cycles.
- **L2/L3 cache size.** Dealing with large amounts of data requires frequent memory access. An L2/L3 cache improves the performance on large memory fetches.
- **System architecture.** Because ISA Server transfers large data loads between network devices, memory, and the CPU, the system elements around the CPU also have an effect on ISA Server performance. A faster memory front side bus (FSB) and faster I/O buses improve overall capacity.

CPU bottlenecks are characterized by situations in which
`\Processor\% Processor Time`
performance counter numbers are high while the network adapter and disk I/O remain well below capacity. In this case, (which is the ideal CPU-maximized system), reaching 100 percent means that the CPU power must be increased, either by upgrading to a faster CPU, or by adding more processors. For

information about CPU scaling options, see Scaling Out ISA Server. If ISA Server continues to have high response times, but low CPU percentages, the bottleneck is elsewhere.

Determining Memory Capacity

ISA Server memory is used for:

- Storing network sockets (mostly from a nonpaged pool)
- Internal data structures
- Pending request objects

In Web Proxy caching scenarios, memory is also used for:

- Disk cache directory structure
- Memory caching

Because ISA Server handles numerous simultaneous connections requiring system nonpaged memory, the limiting memory factor is the size of the nonpaged pool, which is implied by the total memory size. For Microsoft Windows Server™ 2003 and Windows® 2000 Server, minimum and maximum values of nonpaged pool size are shown in the following table.

Physical memory (MB)	128	256	512	1,024	2,048	4,096
Minimum nonpaged pool size	4	8	16	32	64	128
Maximum nonpaged pool size	50	100	200	256	256	256

When Web caching is not enabled, 512 MB should be enough for single processor computers, and 1,024 MB is sufficient for dual processor computers. These amounts can also store the full memory working set capacity.

The most critical evidence that memory is not tuned correctly, is when \Memory\Pages/sec (measuring hard page faults per second) is large (above 10) during peak loads. If this happens, the first action depends on whether Web caching is enabled:

If Web caching is disabled, you must determine if more physical memory is needed by monitoring the memory used by all processes in the system. The following performance counters will assist you:
\Memory\Pages/sec
\Memory\Pool Nonpaged Bytes
\Memory\Pool Paged Bytes
\Process(*)\Working Set

If Web caching is enabled, first try to lower memory cache size to 10 percent of physical memory. If hard page faults still occur, proceed with step 1.

Determining Network Capacity

Every network device that exists on a connection has its capacity limit. These include the client and server network adapters, routers, switches, and hubs that interconnect them. Adequate network capacity means that none of these network devices are saturated. Monitoring network activity is essential for assuring that actual loads on all network devices are below their maximum capacity.

There are two general cases where network capacity impacts ISA Server performance:

- **ISA Server is connected to the Internet using a WAN link**. In most situations, the Internet connection bandwidth sets the limit for the amount of traffic. It is probable that the cause for weak performance during peak traffic hours is over-utilization of the Internet link.
- **ISA Server is connected only to LANs.** In this case, it is important to have an infrastructure to support maximum traffic requirements. However, in most situations, this is not a concern due to the low price of 100-Mbps and 1-Gbps LANs.

To monitor network activity, use the performance counter:
\Network Interface(*)\Bytes Total/sec
If its value is more than 75 percent of the maximum bandwidth of any network interface, consider increasing the bandwidth of the network infrastructure that is not adequate.

Determining Disk Storage Capacity

ISA Server uses disk storage for:

- Logging firewall activity
- Web caching

If both are disabled or if there is no traffic, ISA Server will not perform any disk I/O activity. In a typical setup of ISA Server, logging is enabled and configured to use Microsoft Data Engine (MSDE) logging. For most deployments, a single disk is enough to serve the maximum logging rate. If Web caching is enabled, disk storage capacity must be planned carefully as explained in Web Caching.

The limiting factor of any disk storage system is the number of physical disk accesses per second. This number varies according to how random these accesses are, and how fast the physical disk spins. Usually, the limit is between 100 to 200 accesses per second. The performance counter to use for monitoring

the disk access rate is:
\PhysicalDisk(*)\Disk Transfers/sec
If this limit is reached on a disk for a sustained period of time, you can expect the system to slow down, which you will notice by an increase in system response time. To remove this bottleneck, the immediate solution is to lower disk accesses by adding more physical disks.

Another cause for a high disk access rate is hard page faults. For troubleshooting this situation, see Web Caching.

Application and Web Filters

ISA Server uses application filters to perform application level security inspection. An application filter is a dynamic-link library (DLL) that registers to a specific protocol port. Whenever a packet is sent to this protocol port, it is passed to the application filter, which inspects it according to application logic and decides what to do according to policy. When no application filter is assigned to a protocol, data undergoes TCP stateful filtering. At this level, ISA Server only checks the TCP/IP header information.

In general, application level filtering requires more processing than TCP stateful filtering for several reasons:

- Application filters inspect the data's payload, while TCP stateful filtering looks only at the TCP/IP header information. Application filters can perform other actions with the data's payload, such as looking at it and blocking it, or changing content according to application logic.
- Application filters work in user mode space. Transport level filtering works in kernel mode. This means extra processing overhead for passing the data through the full operating system networking stack.

Because application filters are firewall processing extenders, they can have an impact on performance. We recommend:

- Obtain performance information for the filters you use, and tune them to be as efficient as possible. One example is the HTTP Web filter that can be configured to look at HTTP payload and search for specific signatures. Enabling this feature provides extra processing that will reduce the demands on the ISA Server computer.
- Where applicable, consider using ISA Server rules instead of a filter. For example, site blocking using access rule destination sets may be more efficient than a Web filter that does the same thing.
- If you develop a filter, optimize it for best performance. This is recommended for any software, especially for a mission-critical firewall or proxy server.
- ISA Server allows using application filtering and lower level TCP stateful filtering for the same application port depending on source and destination

networks. For example, you can filter Internet traffic at the application level, while using transport filtering protection on traffic passing between all other networks.

Logging

ISA Server provides two major methods for logging firewall activity:

- **MSDE logging.** This method is the default logging method for firewall and Web activity. ISA Server writes log records directly to an MSDE database to enable online sophisticated queries on logged data.
- **File logging.** With this method, ISA Server writes log records to a text file in a sequential manner.

In comparing the two methods, MSDE has more features, but it uses more system resources. Specifically, you can expect an overall 10 to 20 percent improvement in processor utilization when switching to file logging from MSDE.

MSDE logging also consumes more disk storage resources. MSDE logging performs about two disk accesses on every megabit. File logging will require the same amount of disk accesses for 10 megabits. One way to improve ISA Server performance is to switch from MSDE to file logging. This is recommended only when there is a performance problem caused by saturated processor or disk access.

Scenarios

ISA Server supports a range of deployment and application scenarios. The following sections describe the major scenarios and their performance characteristics.

Deployment Scenarios

Deployment scenarios refer to the location of an ISA Server computer within a corporate intranet. Due to security and performance considerations, several popular scenarios have evolved over the years, and the following sections describe each from a performance and capacity perspective.

Internet Edge Firewall

Organizations with enterprise-scale capacity requirements may consider deploying an ISA Server computer as a dedicated Internet edge firewall acting as the secure gateway to the Internet for all corporate clients. To maintain high throughput levels of hundreds of Mbps between the internal networks and the Internet connection, ISA Server can be configured to provide packet level and stateful transport layer filtering only.

The more advanced application level filtering that ISA Server provides will be enabled on the second layer of defense, which is comprised of back-end firewall ISA Server computers.

Departmental or Back-End Firewall

The next line of defense for enterprise-scale organizations includes several ISA Server computers that are deployed as departmental or back-end network firewalls that provide secure inbound and outbound access control into and out of protected LANs. Organizations with existing firewall infrastructures may keep their current high-performance firewalls at the Internet edge and offload sophisticated application layer filtering to ISA Server firewalls at the LAN edges. This would allow an organization to utilize current high-speed Internet connections while benefiting from the unique level of protection provided by ISA Server 2004 application layer filtering capabilities.

From a performance perspective, a departmental ISA Server firewall is required to sustain only a portion of the total traffic going through the edge firewall, allowing for more resource-consuming security features to be running, such as application filters.

Branch Office Firewall

ISA Server can be used to securely connect branch office networks to a main office using site-to-site VPN connections. In this deployment, ISA Server is placed at a branch office where it acts both as a firewall protecting the branch office network and as a VPN gateway connecting the branch office network to the main office network.

In general, a transport level filtered site-to-site VPN consumes only 25 percent of the processing power per unit of traffic that is required for application level filtered Internet access.

> **Note:** In a transport level filtered site-to-site VPN, the traffic going through the tunnel is not inspected by application level filters. Application level filtering for site-to-site VPN traffic, like any other traffic, is enabled on a per-protocol basis.

Web Proxy Scenarios

Most traffic on the Internet and inside today's corporate networks is HTTP. An analysis of traffic patterns of many protocols indicates that HTTP is demanding in terms of network performance. Therefore, typical Web traffic workload simulations are realistic for measuring any firewall's capacity and performance characteristics.

Note: One typical metric to validate network performance is the amount of transactions that are exchanged per TCP connection. Typical values for HTTP (3 to 5 on the average) are low as compared to other protocols.

The following table summarizes the hardware recommendations for supporting HTTP traffic on three typical single-computer deployments according to Internet link bandwidth.

Internet link bandwidth	Up to 5 T1 (7.5 Mbps)	Up to 25 Mbps	Up to T3 (45 Mbps)
Processors	1	1	2
Processor type	Pentium III 550 MHz (or higher)	Pentium 4 2.0–3.0 GHz	Xeon 2.0–3.0 GHz
Memory	256 MB	512 MB	1 GB
Disk space	150 MB	2.5 GB	5 GB
Network interface	10/100 Mbps	10/100 Mbps	100/1000 Mbps

These requirements in the preceding table are for default ISA Server 2004 installation settings, and a policy configuration containing hundreds of rules. This includes all default application and Web filtering as well as MSDE logging. The following applies to the preceding table:

- **Internet link bandwidth.** The bandwidth figures apply to a demanding workload where ISA Server 2004 is utilized as a transparent Web proxy with full HTTP application layer filtering. Serving as a forward or reverse Web proxy, ISA Server may double the throughput, meaning that the minimum recommended computer for T3 bandwidth is a single Pentium 4 processor, and a dual processor computer for two T3 connections. For details about performance differences between various Web proxy scenarios, see Proxy Scenarios.

  In deployments requiring only stateful filtering (no need for higher application level filtering), the recommended hardware reaches LAN wire speeds. For details, see Stateful Filtering.

  With Web caching enabled, it is possible to lower the Internet link bandwidth by 20 to 30 percent depending on byte hit ratio. For details, see Web Caching.

- **Processors.** The figures were obtained by simulating HTTP traffic on thousands of IP addresses, loading an ISA Server processor to 70 to 80 percent utilization.

- **Processor type.** Other processors emulating the IA-32 instruction set that have comparable power may also be considered.
- **Memory.** The memory requirements do not take into account memory space for Web caching. For information about additional memory for Web caching, see Web Caching.
- **Disk space.** The disk space requirements indicate the amount of free disk space that is recommended for ISA Server logs. For planning disk space requirements for Web caching, see Web Caching.
- **Network interface.** The network interface requirements are for the internal networks (those not connected to the Internet).

ISA Server secures HTTP traffic using its built-in Web Proxy application filter. This application filter supports three different scenarios: forward proxy and transparent proxy for protecting outbound access to the Internet for corporate users, and reverse proxy for protecting inbound access of Internet users to internal websites. The next sections describe each of these scenarios from a performance perspective and explain how caching can be used to improve performance.

Proxy Scenarios

This section provides scenarios for forward proxy, transparent proxy, and reverse proxy.

Forward Proxy

In forward proxy, client Web browsers are aware of the presence of the proxy. In Internet Explorer, for example, this is done by setting **proxy server** or **automatically detect settings** in **Internet Options**. When Web clients are aware of the proxy, they open connections directly to the proxy, and send the proxy requests for locations on the Internet. (For example, Internet Explorer will open two connections to the proxy when sending HTTP 1.1 requests.) When ISA Server receives a request for a server, it opens a connection to this server, and reuses it for other requests coming from other clients to the same server. This leads to a star connection topology.

The performance advantage of this scenario is that it allows for high reuse of connections, which minimizes the number of open connections as well as the connection rate.

Transparent Proxy

In transparent proxy, client Web browsers are unaware of the proxy's presence. They sense that they are routed directly to servers on the Internet with no agent in between. Specifically, Web clients access Internet servers directly by opening

connections with the target websites. This leads to a considerable increase in connection rate, because after a user asks for a page on a new server, the Web browser shuts down its connections with the current Web server and opens new connections with the new Web server. This is typical of transparent proxy and has an effect on ISA Server performance. Typically, the client-side connection rate in transparent proxy is about three times higher than in forward proxy, which consumes about twice as many processor cycles per request.

Transparent proxy is a popular scenario because it is easy to deploy, especially for Internet service providers (ISPs) that have a heterogeneous client base. For this reason, there are considerable performance improvements in this scenario.

In general, ISA Server requires twice the amount of CPU resources for transparent proxy as compared to forward proxy.

Reverse Proxy

Reverse proxy or Web publishing works in the same manner as forward proxy, but the direction is inbound instead of outbound. In this scenario, ISA Server acts as a website accessed by clients on the Internet. The clients do not know that the website they are accessing is actually a proxy. As with forward proxy, the number of connections and connection rate are minimal, due to efficient connection reuse. Reverse proxy is used for secure publishing of Web servers, such as Internet Information Services (IIS), Outlook® Web Access, SharePoint® Portal Server, and many more.

From a performance perspective, reverse proxy has characteristics similar to forward proxy. The main difference is that the major amount of traffic flows from ISA Server to Internet users, requiring a large Internet connection. As explained in the next section, forward proxy and reverse proxy have different performance impacts when Web caching is enabled.

Web Caching

Web caching is a feature for improving the performance of ISA Server in all Web proxy scenarios. But the performance improvement impact is different when enabling the cache for the outbound scenarios (forward and transparent proxy) and the inbound reverse proxy scenario.

The main difference between forward (transparent) and reverse caching is the purpose of the cache. Forward (and transparent) caching is intended to save Internet bandwidth costs and to reduce response time by placing popular cacheable content near users. Reverse caching is used for offloading the back-end Web servers. Reverse caching has no effect on response time, and will even increase latency for objects that are not cached.

In terms of savings, forward caching saves access attempts to Web servers on the Internet by serving those attempts from the cache, thus saving on required Internet link bandwidth. For example, if the cache byte hit ratio is 20 percent and peak throughput on the internal links is 10 Mbps, the peak throughput on the Internet link would be only 8 Mbps.

> **Note:** Cache object hit ratio is the proportion of objects that are served from the cache out of the total objects that are served by the proxy. Likewise, cache byte hit ratio is the proportion of bytes that are served from the cache out of the total bytes that the proxy serves. Common average values are about 35 percent object hit ratio and about 20 percent byte hit ratio.

Reverse caching helps in consolidation of Web servers, reducing both hardware and management costs. For example, if 80 percent of a website's data is static and cacheable, and a dynamic object requires four times more CPU cycles as compared to a static object, utilizing a reverse proxy will reduce the number of Web servers by 100 percent.

> **Note**: Suppose a static object requires X CPU cycles, and a dynamic object requires 4X cycles. If 80 out of 100 requests are static, the total number of cycles required for 100 requests is 80X + (100-80)4X = 160X, and 50% of those utilized for static content that will be served by an ISA Server cache.

Another difference between forward cache and reverse cache is the magnitude of the cached working set. In reverse cache, the size of the client space is unlimited, but the server space contains only several websites and a relatively small number of objects. In most cases, ISA Server can be designed with reasonable memory and disk space to store all the hosted cacheable content in its cache, so that only dynamic uncacheable content is directed to the hosted Web servers. Preferably, all cache can be kept and served in memory.

In forward cache, the server space contains a limitless number of websites and Web objects, so the cache working set is limitless. To hold such a large working set, you must define large disk caches. The next sections describe how to plan and tune Web cache capacity for forward and reverse caching.

Tuning Forward Cache Memory and Disks

In forward caching, object hit ratio and peak HTTP request rate are used to determine the number of necessary disks according to the following formula:

$$number\_of\_disks = \left\lceil \frac{peak\_request\_rate \times object\_hit\_ratio}{100} \right\rceil$$

For example, if peak request rate is 900 requests per second and object hit ratio is 35 percent, four disks are required.

Note: The number 100 in the preceding formula is empirical and means that the average performing physical disk (spinning up to 10,000 rounds per minute) can serve 100 I/O operations per second. A faster disk spinning at 15,000 rounds per minute can do 130-140 I/O operations per second.

We recommend using dedicated disks of the same type and of equal capacity. If a RAID storage subsystem is used, it should be configured as RAID-0 (no fault tolerance). Small disks, preferably no more than 40 GB, are recommended.

Tuning cache memory is more complicated. In cache scenarios, memory is used for:

- **Pending request objects**. Number of pending request objects is proportional to the number of client connections to the ISA Server computer. In most cases, it will be less than 50 percent of client connections. Each pending request requires about 15 KB. For 10,000 simultaneous connections, the Web Proxy memory working set has no more than 50% × 10,000 × 15 KB = ~ 75 MB allocated for pending request objects.
- **Cache directory.** Directory containing a 32-byte entry for each cached object. The size of the cache directory is directly determined by the size of the cache and the average response size. For example, a 50-GB cache holding 7,000,000 objects (~7 KB each on the average) requires 32 × 7,000,000 = 214 MB.
- **Memory caching**. The purpose of memory caching is to serve requests for popular cached objects directly from memory lowering disk cache fetches. But because cacheable content is unlimited in forward caching, the memory cache size has a limited effect on performance.

By default, the memory cache is 10 percent of total physical memory, and is configurable. In general, we recommend using the default setting unless hard page faults occur. Hard page faults cause severe performance degradation. The easiest way to fix this situation when using caching is to lower the size of the memory cache.

Considering this information, use the following process for tuning cache memory size:

Tune disk cache size, as explained in the preceding section.

Estimate required memory as the total of:

Pending request objects (10% × 15 KB * peak-established-connections).

Cache directory size (32 * URLs-in-cache).

Memory cache size (by default, 50 percent of total memory).

System memory will require about 2 KB per connection, with an extra 50 MB to start with (50 MB + 2 KB * peak-established-connections).

At least 100 MB for other processes running in the system.

Monitor memory usage and change memory cache size accordingly. The informative performance counters are:
\ISA Server Cache\Memory Cache Allocated Space (KB)
\ISA Server Cache\Memory URL Retrieve Rate (URL/sec)
\ISA Server Cache\Memory Usage Ratio Percent (%)
\ISA Server Cache\URLs in Cache
\Memory\Pages/sec
\Memory\Pool Nonpaged Bytes
\Memory\Pool Paged Bytes
\Process(WSPSRV)\Working Set
\TCP\Established Connections

Tuning Reverse Cache Memory and Disks

In reverse caching, working set size is so much smaller as compared to forward caching, that it is relevant to try to put it all in memory. The size of the working set is the total amount of cacheable objects in the website that the cache hosts. The size of the disk and memory cache is recommended to be about twice the size of the working set to hold all cacheable objects, and to account for fragmentation in disk allocation and cache refresh policy. For example, a working set of 500 MB requires 1,000-MB disk cache and 1,500-MB memory with memory cache size set to 66 percent.

Because most cache fetches are served from the memory cache, the I/O rate on the disk is low. In most cases, a single physical disk is sufficient, without being a bottleneck.

Using the /3GB Boot.ini Switch

For large systems with over 2 GB of memory, Windows Server 2003 and Windows 2000 Advanced Server offer the 4GT RAM tuning feature. This feature divides a process memory space into 3 GB for application memory and 1 GB for system memory. This feature enables processes to benefit from more than 2-GB RAM in user space, and is enabled by adding the switch /3GB to the Boot.ini file. (For details, see article Q171793 in the Microsoft Knowledge Base.)

This feature may be beneficial for ISA Server, especially for reverse caching hosting a large website. However, using this feature reduces the maximum size of the nonpaged pool (to 128 MB instead of 256 MB), hence the maximum number of concurrent TCP connections.

Web Authentication

There are many methods for performing Web authentication, and each has its own performance impact. The following table summarizes the advantages and disadvantages of each method.

Authentication scheme	Strength	When authentication is performed	Overhead per request	Overhead per batch
Basic	Low	Per request	Low	None
Digest	Medium	Per time/count	None	High
NTLM	Medium	Per connection	None	High
NTLMv2	High	Per connection	None	High
Kerberos	High	Per connection	None	Medium
Secure ID	High	Per browser session	None	Medium
RADIUS per request (default)	High	Per request	High	None
RADIUS per session	Medium	Once	None	Low

From a performance perspective, an authentication scheme performs best with no per request overhead, and a low per batch overhead. Deciding which authentication scheme to use depends on strength and infrastructure.

Also, Web Proxy authentication can be configured on the Web Proxy listener level or on a rule level. Choose the listener level only if authentication is required for all Web access. Otherwise, choose the rule level, which means that authentication will be performed only when necessary according to rules.

Web Filters

Like application filters, Web filters may also have an impact on performance, depending on what they do. ISA Server incorporates several Web filters that perform specified tasks. Of these, the most CPU consuming are the HTTP filter and the link translation filter.

HTTP filter inspects every Web request and response, checking that they comply with normal HTTP protocol usage. It is enabled by default and its default configuration provides size limits to HTTP headers and the URL. Other available features include blocking by methods, extensions, headers, and HTTP payload signatures. These functions have no performance impact when selected, except for signature blocking, which requires 10 percent more CPU cycles. HTTP filter is recommended for protecting Web traffic.

Link translation is used specifically in Web publishing scenarios. It looks in HTML response bodies, searching for absolute hyperlinks, and changes them to point to the ISA Server computer instead. By default, link translation scans only HTTP headers and does not scan response bodies, so there should be no noticeable performance impact. Also, when body scanning is enabled, it scans by default only HTML content, causing an overall 5 percent increase in CPU utilization.

Stateful Filtering

Stateful filtering inspects data at transport level and is implemented in the ISA Server Firewall Packet Engine kernel-mode driver. Stateful filtering evaluates source and destination IP addresses, TCP/UDP flag port numbers and options, and Internet Control Message Protocol (ICMP) types and codes. It uses this information to determine the state of the connection, allowing packets that conform to this state, and denying packets that do not conform.

Stateful filtering requires only a small amount of the resources that application level filtering requires. The same HTTP traffic amount that utilizes 75 percent of the CPU power with Web Proxy filtering will utilize only 8 percent of CPU power with stateful filtering (a performance increase factor of 10).

VPN

A virtual private network (VPN) consists of two basic scenarios: remote access VPN and site-to-site VPN. Both can use several protocols and work in conjunction with application filtering or stateful filtering. Internet Protocol security (IPSec)-based protocols can also utilize hardware offloading capabilities available in many network adapters, improving overall processor utilization. Some protocols can work with compression for increasing throughput or saving bandwidth. All of these features impact performance, as explained in the next sections.

Remote Access VPN

Remote clients dialing in from the Internet use VPN remote access to access their corporate networks. Protocols that are used in remote access are Point-to-Point Tunneling Protocol (PPTP) and Layer Two Tunneling Protocol (L2TP) over Internet Protocol security (IPSec). Both of these protocols support compression, which is recommended because it saves bandwidth and processing power required for encryption.

To determine adequate capacity for an ISA Server VPN server, you first need to evaluate the maximum number of concurrent remote connections that your ISA Server computer needs to support. For example, if you expect to have no more than 5 percent of your organization's employees establishing remote connections simultaneously, and your organization has 5,000 employees, 250 concurrent VPN remote access connections is the capacity you need.

The table below indicates the maximal number of concurrent VPN remote access connections supported by each hardware platform. These figures assume out-of-the-box ISA Server setup incorporating Web Proxy filtering, MSDE logging, and compression for both PPTP and L2TP/IPSec protocols.

Protocol	Connections and bandwidth	Single Pentium III 550 MHz processor	Single Pentium 4 3 GHz processor	Dual Xeon 3 GHz processors
PPTP	Connections	150	600	760
	Bandwidth	2.25 Mbps	9 Mbps	11.4 Mbps
L2TP/IPSec	Connections	150	700	850
	Bandwidth	2.25 Mbps	10.5 Mbps	12.75 Mbps

The following applies to the preceding table:

- **Bandwidth.** Bandwidth figures are the required Internet link bandwidth. The actual bandwidth is twice the amount shown in the preceding table, due to compression.

  The bandwidth figures assume an average throughput of 30 Kbps per connection, approximately equivalent to a 56-KB dial-up connection.

In deployments where VPN clients can be trusted to a higher degree, Application level filtering may be disabled, improving total capacity on the account of loosening the security level. The next table states the same figures for the case where the Web Proxy filter is disabled.

Protocol	Connections	Single	Single Pentium 4

	and bandwidth	Pentium III 550 MHz processor	3 GHz processor
PPTP	Connections	375	1,000
	Bandwidth	5.6 Mbps	15 Mbps
L2TP/IPSec	Connections	330	1,000
	Bandwidth	5 Mbps	15 Mbps

The following applies to the preceding table:

- **Connections.** The single Pentium 4 3 GHz processor is capable of reaching the maximum number of concurrent connections shown in the preceding table.

Site-to-Site VPN

In site-to-site VPN, there are two main choices from a performance and capacity perspective. One choice is using either PPTP or L2TP over IPSec. These protocols provide compression of the application traffic, which doubles the throughput that can be transferred through the site-to-site link. For example, sending a 2-MB file through a PPTP or L2TP tunnel will actually pass only 1 MB. The other choice is using IPSec tunneling, which does not incorporate compression. So in effect, PPTP and L2TP over IPSec save site-to-site throughput by 50 percent, as compared to IPSec tunneling.

With Web Proxy filtering disabled, L2TP over IPSec requires a single Pentium III 550-MHz processor for 15-Mbps application traffic. Passing this traffic in one direction requires only 7.5-Mbps link capacity due to compression. A single Pentium 4 3-GHz processor can handle up to 90-Mbps application traffic requiring T3 link capacity (45 Mbps). When Web Proxy filtering is enabled, a Pentium III 550-MHz processor can sustain 7-Mbps application traffic requiring 3.5-Mbps Internet link bandwidth, while a single Pentium 4 3-GHz processor handles 34-Mbps application traffic corresponding to 17-Mbps Internet bandwidth. Dual Xeon 3-GHz processors can handle 53-Mbps application traffic requiring 26.5-Mbps Internet link bandwidth. PPTP can handle about 15 to 20 percent more throughput for the same CPU consumption.

The second choice is using IPSec tunneling, which does not support compression, meaning that Internet link traffic is the same as application traffic. When working in conjunction with stateful filtering (Web Proxy filter is disabled), IPSec tunneling can handle 10 Mbps on a single Pentium III 550-MHz processor and 52 Mbps on a single Pentium 4 3-GHz processor. With Web Proxy filtering

enabled, the throughput figures are 4 Mbps, 18 Mbps, and 30 Mbps for the single Pentium III, single Pentium 4, and dual Xeon platforms respectively.

Scaling Out ISA Server

There are several ways to scale out an ISA Server system:

- **Using high-level network switching hardware gear.** These switches are often called L3, L4, or L7 switches (layer 3, layer 4, or layer 7) because they provide switching capabilities based on various information available at different networking layers. L3 switching is based on packet layer information (IP), L4 is based on transport layer information (TCP), and L7 performs switching based on application data (HTTP headers). The information available at these levels can provide sophisticated load balancing, according to IP source or destination addresses, TCP source or destinations ports, URL, and content type. Because the switches are implemented as hardware appliances, they have a relatively high throughput, and are highly available and reliable, but also expensive. Most switches can detect server-down conditions, enabling fault tolerance.
- **Using DNS round-robin name resolution.** A cluster of servers can be assigned the same name in DNS. DNS responds to queries for that name by cycling through the list. This is an inexpensive (no-cost) solution, but has drawbacks. One problem is that the load is not necessarily distributed evenly between servers in the cluster. Another problem is that it provides no fault tolerance.
- **Using Windows Network Load Balancing.** Network Load Balancing works by sharing an IP address with all the servers in a cluster, and all data sent to this IP is viewed by all servers. However, each packet is served by only one of the servers, according to some shared hash function. Network Load Balancing is implemented at the operating system level. It provides evenly distributed load balancing and supports fault tolerance. (Other servers in the cluster can detect a failing server and distribute its load between them.) However, it requires CPU processing overhead (about 10 to 15 percent for common ISA Server scenarios), and has a limit to the number of members in the cluster (about 8 computers as the recommended maximum). Network Load Balancing is available on Windows Server 2003 and Windows 2000 Advanced Server.

Because ISA Server maintains a state for each stream that passes through, all scale-out methods must support "stickiness" so that all data goes through the ISA Server computer.

Scaling is used for increasing the capacity of a system. Each scaling method has its benefits and drawbacks, and for ISA Server it also depends on the scenario. When deciding which scale method to use, consider the following:

- **Performance factor.** Amount of throughput that can be gained by adding another computer.
- **System cost.** Initial cost of buying the system, and not the cost of ownership.
- **System administration.** Level of complexity in administering the system. This has a direct impact on the system's cost of ownership.
- **Fault tolerance.** Method used by the system to enable high availability and reliability.
- **System growth.** Method used to increase the processing power of the system. The cost of upgrades is also an important consideration.

The following are some tradeoffs to consider when deciding to scale out:

- **Single point of failure vs. fault tolerance.** The availability of a single computer deployment is more susceptible to hardware failures than a multiple computer cluster. A failure in the system board or disk controller will cause the entire system to fail, requiring repair. This is also true for a hardware load balancer that has a malfunction.
- **Growth.** Upgrading a single computer solution from one processor to two processors is simple, provided there is an empty processor slot in the computer (or available ports in the hardware load balancing switch). In multiple computer clusters, adding another computer is more complicated.

The following table summarizes the scale-out methods.

Features	Hardware switch	DNS round-robin	Windows Network Load Balancing
Scale factor	2	2	1.8 (for 1 to 8 computers in a cluster)
System cost	Expensive	No added cost	Requires Windows Server 2003 or Windows 2000 Advanced Server
Fault tolerance	Depends on switch (most detect failing	None	By mutual detection of failing computer

	computer and load the others)		

References

Bandwidth Needs of Enterprises, SMBs and Teleworkers Through 2006, Gartner Report R-18-3617, September 30, 2002.

Do you have comments about this document? Send feedback.