Sun™ ONE Web Proxy Server Deployment Guide

This guide is intended to assist customers who have decided to deploy Sun™ Open Net Environment (Sun ONE) Web Proxy Server (formerly, iPlanet Web Proxy Server) on their intranet or extranet. It assumes you are familiar with the product and therefore does not cover its features in depth. The Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide is the best resource for detailed information on proxy server configuration. For more detailed information on product features and functionality, refer to the Sun ONE Web Proxy Server data sheet or evaluation guide available at http://www.sun.com.

This guide concentrates on the information you need to plan and deploy the proxy server in your organization. It covers the deployment process sequentially, from beginning to end, and is designed to answer the questions you may have at each stage. The guide is organized as follows:

- Deciding Which Services You Want to Provide
- Determining Where to Deploy the Servers
- Deciding Which Architecture to Use
- Determining Number of Servers to Use
- Deciding What Type of Hardware to Use
- Configuring the Servers
- Tuning the Servers
- Monitoring the Servers
- Planning for Growth
- Contacting Sun Microsystems Technical Support

- Further Information

Deciding Which Services You Want to Provide

Although this guide provides much of the information you will need to implement your proxy solution, it does not attempt to cover every possible scenario. The most common setups and configurations are addressed. Customers planning to deploy more unusual proxy server architecture may need to seek additional resources.

A proxy server can bring various capabilities to your intranet and extranet. Understanding these capabilities will assist you in developing your deployment strategy. Sun ONE Web Proxy Server is designed to provide the following core capabilities:

- Proxying
- Caching
- Filtering and Access Control
- Logging

Proxying

The server provides access for internal clients, through a firewall, to the Internet. This service is often provided as part of a larger intranet security strategy and is known as "forward proxying." Forward proxying allows your clients to go outside the firewall without compromising the integrity of your private network. A server can also provide access for external clients, through a firewall, to internal content. This service is often used for secure web publishing and is known as "reverse proxying."

Caching

The server can cache web content locally, conserving bandwidth at network bottlenecks by storing frequently requested content locally. This content can be downloaded and regularly checked for changes in order to return up-to-date documents to all requests.

Filtering and Access Control

The server provides fine-grain access control to web content from your intranet. Network administrators can use filters to block access to any Internet URL or to alter the actual content stream. Using an access control list, filters can be applied to specific addresses, groups of addresses, individual users or groups of users.

Logging

The server records all errors and accesses for reporting purposes. Logs provide useful information to network administrators and group managers. Network administrators often analyze log files to monitor server usage and performance. Group managers find log-based reports useful in tracking Internet usage among their employees.

You may find that your organization has a particular need for one or more of these services. Perhaps you will fully exploit all of them. In any case, the deployment strategy you choose and your ultimate proxy configuration should facilitate the services you will use most often.

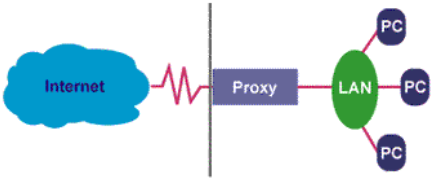
Determining Where to Deploy the Servers

The most common place to deploy a proxy server is at a network bottleneck. Bottlenecks are often created by slow connections at network gateways. Managing bandwidth at these locations is imperative as your business grows and network traffic continues to increase. The Internet gateway and the branch office connection are two likely bottleneck locations and are therefore prime candidates for a proxy server deployment.

Internet Gateway - Forward Proxy

Placing one or more proxy servers at the Internet gateway is the most common deployment scenario for the enterprise. In this location, Sun ONE Web Proxy Server provides gateway services at the application level with a web proxy as well as at the circuit level through SOCKS. The benefit of this type of deployment is enhanced Internet access. Web content caching reduces response times, facilitates bandwidth conservation, and helps reduce your overall communications expense. In addition, content filtering and access control allow you to manage the material on your intranet.

Figure 1 Forward Proxy Server Deployed at the Internet Gateway, Inside the Firewall

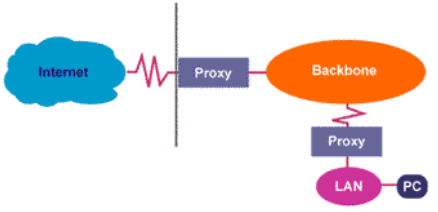


A variety of architectures can be used to deploy proxy servers at the Internet gateway. These implementations of Sun ONE Web Proxy Server will be discussed in the section . “Deciding Which Architecture to Use,” on page 18.

Branch Office - Forward Proxy

Corporations are deploying proxy servers in increasing numbers on their intranets, both in remote locations and on major subnetworks. Proxy servers deployed at major subnetwork connections can drastically reduce the traffic on your corporate backbone. At remote offices, which are often connected via slow links to the corporate network, proxy servers can provide a quick mechanism for replicating content, providing better company integration, and increasing network performance - all of which can be achieved without large capital and communications expense. Outside the United States, proxy servers offer even more savings potential because of the great expense of communications bandwidth overseas.

Figure 2 Forward Proxy Servers Deployed at the Internet Gateway and at a Remote Location



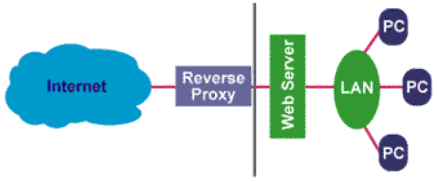
Many organizations are seeing the value of deploying proxy servers throughout their intranet. Types of deployments that use multiple servers can take advantage of the proxy routing capabilities of Sun ONE Web Proxy Server. Proxy routing allows you to chain proxies together to create a hierarchical caching system that can better serve the various organizations within your enterprise.

Sun ONE Web Proxy Servers to cache content locally, setting up a hierarchy of servers for client access. The result is a managed network of proxy servers that is completely transparent to the user. In a typical implementation, smaller, local proxies might be situated near end user communities, with larger proxies near the firewall and external connections. For most installations, two levels of hierarchy is optimum, but you may benefit from adding more levels, depending on the size of your organization and where the bottlenecks occur on your network.

Internet Gateway - Reverse Proxy

Reverse proxying is a special deployment case in which a proxy server is placed outside the firewall to represent a content server to external clients. This type of deployment allows you to expose selected content without exposing the web servers that host it or other elements of your private network.

Figure 3 Reverse Proxy Server Deployed at the Internet Gateway, Outside the Firewall



In reverse proxy mode, the proxy server functions more like a web server with respect to the clients it services. Unlike internal clients, external clients are not reconfigured to access the proxy server. Instead, the site URL routes the client to the proxy as if it were a web server. Replicated content is delivered from the proxy cache to the external client without exposing the origin server or the private network residing safely behind the firewall. Multiple reverse proxy servers can be used to balance the load on an over-taxed web server in much the same way.

Reverse proxy servers are commonly used for secure web publishing. Having a proxy server accepting and filling outside requests allows you to keep your web server behind the firewall. You can then use the web server as a protected web site, staging documents for testing before they are published externally. When you are ready, you can publish selected content to the reverse proxy server's cache.

Deciding Which Architecture to Use

Your Sun ONE Web Proxy Server can be deployed independently or in conjunction with a firewall. Or it can be used with a firewall at the Internet gateway but independently at a branch office. The Sun ONE Web Proxy Server by itself does not constitute a firewall and does not eliminate the need for one.

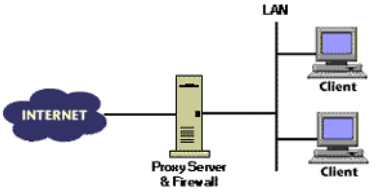
Because proxy servers are often deployed as part of a firewall solution, this section discusses the various firewall architectures you may use at your site. The special case of reverse proxy servers and their implementation are also addressed.

The proxy server can be deployed in conjunction with almost any firewall architecture. However, the firewall architecture you choose may affect the way you implement your proxy server. Three common firewall architectures are described below, but there are many variations. All architectures use some combination of proxy servers, firewall software, and hardware routers.

Dual-Homed Host Architecture

A dual-homed host is a computer that has two network interfaces, one connected to an internal LAN and the other to the Internet. As a firewall architecture, the dual-homed host usually incorporates a firewall software package. This firewall basically acts as a software router providing secure connectivity through packet filtering. The proxy deployed in conjunction with a packaged firewall on a dual-homed host provides a complete firewall solution. In addition to caching, Sun ONE Web Proxy Server brings fine-grain filtering and virus scanning to the solution.

Figure 4 Proxy Server Implemented With a Dual-Homed Host Firewall



One drawback to this solution is that a security breach on the single host machine could jeopardize the whole network. For this reason, many security experts recommend firewall solutions made up of multiple redundant components. Still, a dual-homed host solution might appeal to small offices on a budget or organizations that do not require redundant security measures.

Screened Hosts

A screened host consists of a router deployed in front of a server that is hosted on a private network. This router can be a traditional hardware router or a firewall software application providing packet-filtering capabilities and restricting inbound access to the internal network.

Figure 5 Sun ONE Proxy Server Implemented Behind a Screening Router

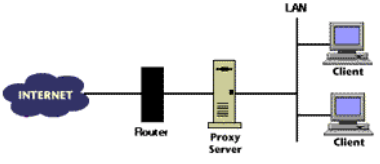
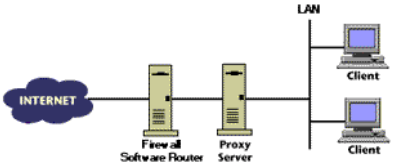


Figure 6 Sun ONE Proxy Server Implemented Behind a Screening Firewall



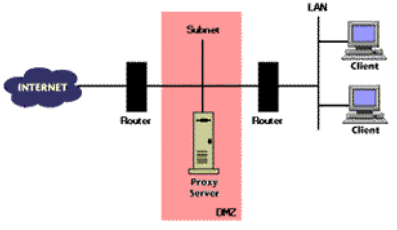
Proxying allows network traffic to gain Internet access through the router. A screening router could also support multiple hosts such as multiple proxy servers or web servers.

One drawback to the screened host deployment is a loss of security should the router fail. This scenario has encouraged the use of multiple routers and the screened subnet architecture. The screened host architecture is appropriate for small to medium-size intranets that require a simple, yet effective security solution.

Screened Subnetwork

A screened subnetwork consists of multiple routers sandwiching a nonsecure network that is outside or part of the firewall solution. This subnetwork is commonly referred to as a DMZ (demilitarized zone). In this scenario, the proxy is deployed in the DMZ and is allowed access to both internal and external networks through the routers. Both internal and external traffic can enter the DMZ but neither can pass through without the assistance of the proxy server and the packet filtering routers.

Figure 7 Sun ONE Web Proxy Server Implemented in a DMZ Between Two Screening Routers



The screened subnetwork is a popular architecture choice for larger organizations with heavily trafficked gateways. For these customers, security is critical and therefore redundancy is imperative. The protected subnetwork also provides an ideal location for other servers that must interface with the secure network and the Internet.

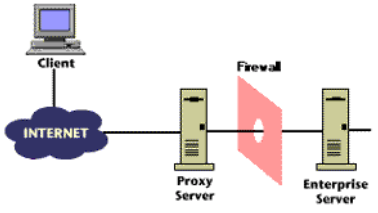
Reverse Proxy

Independent of your firewall architecture, you may want to implement some type of reverse proxy. Reverse proxies are generally deployed in one of two configurations: alone, as a server stand-in; or in groups, for load balancing.

Server Stand-in

In the server stand-in mode, the proxy receives requests for a web server that is protected behind the firewall. Server stand-in facilitates secure web publishing because it allows content on the web server to reside inside the firewall for protection.

Figure 8 Sun ONE Web Proxy Server Implemented in Reverse Mode as a Stand-in for a Web

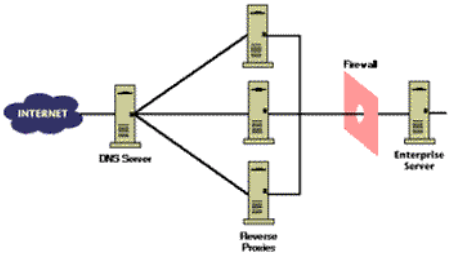


Server stand-in prevents direct, unmonitored access of internal resources from outside the enterprise. In its stand-in role, the proxy server acts like a virtual server mirror. The proxy is positioned similarly to a web server, and is usually placed in the DMZ or on an external subnetwork. As a server mirror, the proxy server provides replication only. The contents of the secure server will be replicated or mirrored in the proxy server cache.

Load Balancing

Multiple reverse proxy servers can be use to balance the load on an overtaxed web server. In this configuration, DNS round-robin is used to route incoming requests to one of a bank of servers. Load balancing helps the host machine handle high-volume requests while reducing the impact on overall performance.

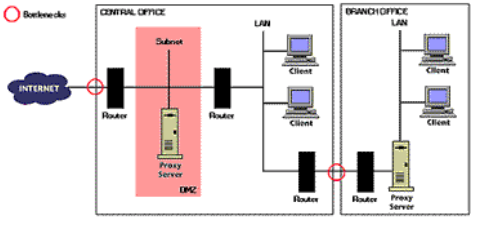
Figure 9 Multiple Sun ONE Web Proxy Servers Implemented in Reverse Mode to Balance the Load on a Web Server



Determining Number of Servers to Use

You may have already assessed where the bottlenecks occur on your network. The number of bottlenecks is a good starting point for the number of proxy servers you will want to deploy. Below is a diagram of one possible enterprise implementation. Sun ONE Web Proxy servers have been deployed at the network bottlenecks.

Figure 10 Sun ONE Web Proxy Servers Implemented at Common Network Bottlenecks



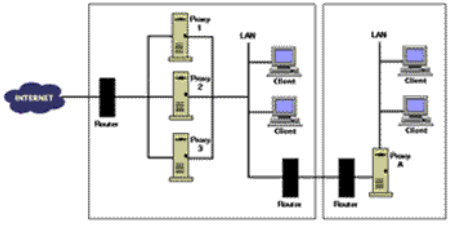
For each bottleneck location you will need to decide whether to deploy a single proxy or multiple proxy servers. This decision will depend on your user load and requirements for redundancy. While a single proxy may be easier to maintain and manage, multiple proxies provide greater reliability and maximize the use of your available bandwidth.

Assuming yours are standard bottlenecks, we recommend deploying an Sun ONE Web Proxy Server at any site where ten or more people connect to your intranet through a WAN. In this situation, a proxy server easily pays for itself with reduced network traffic and increased user performance. And, if the WAN slows down or stops, the regional site has a local copy of the latest content.

Load Balancing and Fail-Over

If proxy availability is critical to your enterprise, you may want to consider deploying multiple proxies to provide fail-over capability. In the diagram below, three proxy servers are deployed at the Internet gateway to balance the load of internal client requests. In this situation, Proxies 1 and 2 might share the load under normal conditions and Proxy 3 might be kept in reserve should one of the other proxies go offline.

Figure 11 Chained Sun ONE Web Proxy Servers Providing Load Balancing and Fail-Over Capabilities



In addition to load balancing between the client and the proxy server, you can also balance the load between proxy servers in a hierarchical chain. Load balancing between Proxies 1, 2, or 3 and Proxy A would be accomplished through an NSAPI plug-in discussed in the section . “Configuring the Servers,” on page 27.

Deciding What Type of Hardware to Use

While planning the number of proxy servers to deploy, you need to anticipate growth and consider how your proxy services will scale. The number and type of users you support and the bandwidth available to them will greatly affect your ability to meet the growing demand.

Capacity Planning

Knowing the total number of users you need to support is important; however, scaling of the proxy server is really dependent on the number of users active at any given time. Each concurrent user is represented by an individual HTTP transaction. In most installations, the concurrent user count is much smaller than the total user base.

You also need to consider the type of use the proxy server will see. A full-time web surfer can generate thousands of requests per day. On the other hand, someone who uses the web less frequently may only generate a few hundred requests in the same period. You need to consider the types of users in your organization and how they will use the proxy services.

Users request different types of content as well. The average size of a web object is somewhere between 10K and 20K; however, your organization may have a much higher average if the majority of your downloads contain rich graphics. The increase in network traffic associated with larger objects must be accounted for when determining capacity requirements.

Estimating Load

Whatever type of users your organization has and whatever type of content they access, your proxy server will need to handle the load. Your highest load may occur at a few peak times. For most businesses, the peak times for web traffic are between 10:30 and 11:30 in the morning and between 1:30 and 4:00 in the afternoon. The proxy server must be sized to accommodate your peak load, but you need not calculate the peak load to do so. You just need to estimate the average load for a given day. Use the following steps to estimate your load and the bandwidth you will need to accommodate it:

1. Record the number of requests for a normal business day, *N*, and divide this number by the length of that business day in seconds, *T*. The result, A^{total}, is your average number of accesses per second.

   $A^{total} = N / T$

2. Experience has shown that you can roughly size your peak capacity by looking at your total accesses per second, A^{total}, and multiplying by 2. The resulting number, C^{peak}, will be the capacity required to handle the accesses at peak times expressed in requests per second.

   $C^{peak} = 2 \times A^{total}$

3. Once you know the number of accesses you must accommodate, you can estimate the necessary bandwidth for the corresponding data that will be transferred. Assuming a typical web object size, such as 15K, you can determine the bandwidth required between the proxy and clients, B^{client}.

   $B^{client} = C^{peak} \times 15$

4. Requests that are serviced by the cache require less bandwidth than those that must go to the origin server. To account for this difference, you must include a factor that assumes a cache hit rate appropriate for your network and user base. Typical cache hit rates are between 30 and 60 percent. A factor, *F*, of 7 or 4 assumes a 30 or 60 percent hit rate, respectively. By assuming a factor to adjust your data transfer rate, you can estimate the amount of bandwidth required between the proxy and origin servers, B^{server}.

   $B^{server} = B^{client} \times F$

5. Once you know the bandwidth required on either side of the proxy server, you know the total bandwidth required, B^{total}.

   $B^{total} = B^{server} + B^{client}$

   or

   $B^{total} = C^{peak} \times 15 (1 + F)$

6. Next you should examine your uplink bandwidth utilization, since this is your ultimate limiting factor. Your available uplink bandwidth will depend on the type of Internet connection you have. Various network analysis tools will allow you to see how much bandwidth you are using. From this information you can determine the percentage of available bandwidth and take this into account in your capacity planning.

Reverse proxy loads may be more difficult to predict than those for the traditional forward proxy server. Since you cannot know with certainty the number of external users, you must rely on an average usage profile. You can get a good idea of your potential reverse proxy load by analyzing the load on the web server currently hosting the content you plan to cache. Gathering this type of information is discussed in the section . “Monitoring the Servers,” on page 34.

Hardware Sizing

An estimate of your required capacity will help you size your hardware. After deployment, you will be able to tune your proxy server to optimize performance, but in the meantime, you must determine an appropriate hardware configuration. Sun ONE Web Proxy Server will provide the best performance when run on a dedicated machine. If at all possible, consider that implementation.

Ideally, hardware sizing is based on the number of incoming connections and the average transaction time of those connections; however, most deployments start with an entry-level or typical hardware setup, such as those outlined below.

Variables	Entry-level Sun ONE Web Proxy Server	Typical Sun ONE Web Proxy Server
Users	Up to 1500	1500 to 3000
Operating System	Entry to mid-level UNIX® or NT server	High-end NT or UNIX server
CPU	120MHz or greater	1 or 2 Pentium Pro processors, UltraSPARC.
RAM	Minimum 32MB, 64MB to 128MB for heavy traffic	128MB to 256MB
Server Hard Disk	Minimum 15MB; 100MB recommended	200MB
Caches	2GB to 4GB	5GB to 9GB

The speed of the CPU you choose is important, but not as important as RAM and disk size. The CPU is normally not a bottleneck for server-grade machines; however, proxy performance does scale with more or faster CPUs on lower end hardware.

The table above suggests a minimum amount of RAM for your proxy server, but you will generally need more RAM as your user base expands. The following table suggests RAM sizes based on the number of users accessing the proxy server. Large deployments should also consider a logging file system or nonvolatile RAM to allow the server to perform asynchronous writes to the cache and greatly improve performance in high-traffic environments.

Users	RAM (MB)
0 to 300	32
300 to 500	64
500 to 1000	96
More than 1000	128

For a UNIX system, each process uses about 200K of RAM for listening and 300K to 500K for working. Estimate approximately 700K in total per process or concurrent user. (As mentioned before, the number of concurrent users will be much less than the total number of users.) It is critical that you have enough actual RAM to hold all the processes in memory when they are active.

Typical cache sizes, may range from 1MB to 20MB per user. An estimate of 10MB is a good place to start. After deployment, continue to monitor the cache performance, watching for increases in the cache hit ratio. You can do this using tools such as sitemon on UNIX. You should keep increasing your cache size as long as the cache hit ratio continues to increase.

There is a tradeoff in selecting the type of disk for your cache. Consider spreading your cache across multiple disks whenever possible. One 10GB disk will store as much content as ten 1GB disks. However, the 10GB disk, while less expensive to purchase and maintain, will not perform as well as the 1GB disks. In general, multiple disks will perform better than a single disk, and multiple disk controllers will always be faster than a single controller.

Configuring the Servers

Once you have purchased your Sun ONE Web Proxy Server, you will need to install the software and begin to configure the services. In this section you will find some general comments on configuring Sun ONE Web Proxy Server and some tips to make the process run smoothly.

Automatic Client Configuration

To manage your proxy deployment efficiently, you should enable automatic proxy configuration in Netscape Navigator clients on your intranet. Client configuration is administered by a Proxy Automatic Configuration (PAC) file, which is downloaded from the server at restart. The PAC file allows you to specify which proxy server, if any, Navigator uses when accessing various URLs. This allows you to do load balancing across multiple proxy servers and to alter your proxy architecture without modifying end user settings. When you add additional proxy servers, you can even specify that they share the same URL. When one proxy server is down, the backup will respond.

You can easily create your own custom PAC files or use one of the examples outlined in the Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide .

Caching

Proper cache setup is critical to the performance of Sun ONE Web Proxy Server. The most important rule to remember when laying out your proxy cache is to distribute the load. Caches should be set up with approximately 1GB per partition and should be spread across multiple disks and multiple disk controllers. This type of arrangement will provide faster file creation and retrieval than is possible with a single, larger cache.

The Cache Batch Update feature in Sun ONE Web Proxy Server allows you to proactively download content from a specified web site or perform scheduled up-to-date checks on documents already in the cache. This gives you the ability to cache content in large quantities at times when traffic on the server is low. Use batch updates to download the most commonly accessed sites at the end of each business day for quick access the following morning. You can use the log files to help determine which sites are frequently accessed. Refer to the Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide for in-depth instructions on creating batch update configurations.

SOCKS

While your web proxy server provides caching and filtering capabilities suitable for web protocols, SOCKS provides a tunneling mechanism for protocols that cannot be proxied or for which is there is no benefit from proxying. SOCKS is firewall software that establishes a connection from inside a firewall to the outside when a direct connection would otherwise be prevented by your security

measures. SOCKS is a circuit-level proxy and is indifferent to the protocols it serves at the application level. For this reason, an application-level proxy server is often configured to use SOCKS for protocols it does not support. Refer to the Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide for instructions on configuring SOCKS.

Templates

Sun ONE Web Proxy Server can use templates to assign unique procedures to specific URLs. You can make the server behave differently depending on the URL the client tries to retrieve. You can also configure different cache refresh settings based on the file type.

The template is just an object that is created in the proxy server's object configuration file, `obj.conf`. Templates allow you to customize how Sun ONE Web Proxy Server interacts with clients by allowing you to do such things as name a set of directives for later reference, simplify complex configurations, or associate named objects with URI patterns.

Experience has shown that many proxy services do not require templates, especially in the early stages of deployment. However, templates become very helpful when it is necessary to edit multiple objects on a regular basis. Check out a "Sample Object Configuration File," on page 39. The template "josh" at the end of this file is called by the "default" object in order to enable certain cache settings.

Instead of making the change several times in each object, the administrator can set up a template that is called by multiple objects and edit the template only once. The payoff increases as your `obj.conf` file becomes more complex and contains objects with a high degree of commonality.

Filtering

Sun ONE Web Proxy Server allows you to filter URLs as well as content. URL filters are applied to requested URLs to determine whether they meet a predetermined set of criteria. URLs can either be allowed or denied based on this criteria. URL filters can also be used in conjunction with access control lists (ACLs) to filter the content that is requested by each client. Filtering with ACLs gives you the ability to restrict or allow access to selected users and groups in your organization. Several URL filters provided by third parties are supported by Sun ONE Web Proxy Server.

Content filters allow you to actually scan and modify the content stream. Virus scanning and HTML tag filtering are achieved through content filtering. Keep in mind that content filtering occurs out of process and can therefore significantly affect performance when applied to large volumes of data. For example, virus scanning places a significant additional CPU load on the server hardware, since the proxy server must compare all incoming data against the known virus patterns.

To create filters and ACLs using the administration server, refer to the Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide.

Server Plug-in Functions

You can create plug-in functions to extend the capabilities of your proxy server by using the Netscape Server Plug-in Application Programming Interface, NSAPI. The server plug-in API is a set of functions and header files that will help you create functions to use with the directives in the server configuration files.

Using the NSAPI directive classes, you can override server functionality, add to it, or create your own custom functions. For example, you could create functions that use a custom database for access control or create custom log files with special entries.

NSAPI also allows you to build advanced configurations with extensions, such as an external filter. The external filter capability, available on UNIX, allows you to use an out-of-process program to filter web content. Currently, this type of out-of-process program is the only way to alter the content stream through the proxy server. With an external filter, all content is piped through your program and filtering occurs before any caching or transfer to the client

NSAPI Directive Classes

Class	Description
AuthTrans	Check user / password
PathCheck	Check validity of the URL
NameTrans	Map URI to another URI
DNS	Use your own DNS function
Connect	Use your own connect() function
Addlog	Perform user-specified logging

Here is what a call to an external filter would look like in `obj.conf`:

```
Filter fn="pre-filter"
path="/path/to/your/filter"
```

Tuning the Servers

As you operate your proxy server and your organization continues to grow, you will probably need to tune your proxy server to get the optimum performance for your particular implementation. Below are a number of tuning tips to help you. Sun ONE Web Proxy Server also provides online forms that can help you tune many of the settings that affect your server's performance.

Time-Outs

There are two time-out settings that significantly affect the performance of the proxy server. These time-outs are the proxy time-out ("timeout") and the time-out after interrupt ("timeout-2"), which is particular to UNIX proxy servers. Here are some tips to help you use these time-outs correctly:

- **timeout**

  This time-out is the proxy time-out and tells the server how long to wait before aborting an idle connection. A long time-out commits a valuable proxy process to a potentially dead client, whereas a time-out that is too short will abort CGI scripts that take a long time to produce their results, such as a database query gateway. For these reasons, we suggest a time-out of 2 to 5 minutes, with an absolute maximum of one hour. To determine the best proxy time-out for your server, consider whether your proxy will be handling many database queries or CGI scripts or whether it will be handling a small amount of requests. In the latter case, you may opt for a higher proxy time-out value because you are less process constrained.

- **timeout-2**

  This time-out is the time-out after interrupt and is used only on UNIX platforms. When a client has aborted a transaction while the proxy is writing a cache file, this time-out allows the proxy to continue writing the cache file; timeout-2 is the idle time-out for a connection in this state. The highest recommended value for this time-out is 5 minutes.

Up-to-Date Checks

The proxy server performs cache up-to-date checks to determine whether requested content in the cache is still valid or needs to be refreshed. You can tune proxy server performance by controlling the number of up-to-date checks. Under the Caching tab in the administration server, specify that documents are not always checked. Choose a reasonable lifetime, somewhere between 8 and 24 hours, that balances caching with the need for fresh data. The value you choose translates to the longest time the proxy server will wait before performing an up-to-date check. The proxy will only check on documents that have not been checked within the specified lifetime.

Last-Modified Factor

A last-modified header is returned from the server with the time the document was last modified. The document is updated based on its freshness and the last-modified (LM) factor. The LM factor is a floating point number that is multiplied by the age of the document since its last modification. The effect here is that recently changed documents are checked more often than old documents. A recommended value for the LM Factor is between 0.1 and 0.2.

DNS Lookups

Domain Name Service (DNS) is the system used to associate standard IP addresses with host names. The proxy server can use forward DNS lookups to resolve an origin server name to an IP address and reverse DNS lookups to resolve client station addresses to names. Excessive DNS lookups can affect the performance of your proxy server and should be avoided. In addition, the load on your DNS servers and their location on your network can also affect performance. Here are some things you can do to avoid a performance hit:

- **Enable DNS Caching.**

  On the NT platform, DNS caching and negative caching are always enabled.

- **Log Only Client IP Addresses.**

  The proxy server has the ability to log client host names; however, you will see better performance if you can get along without it. Set your log preferences to log client IP addresses only.

- **Disable Reverse DNS.**

If you will not be logging client host names, you can disable reverse DNS.

- **Avoid ACLs with Client Host Names.**

  Use client IP addresses instead, if possible.

Number of Processes

On the UNIX platform, the administrator must specify the number of processes that will be preforked on the server. Preforking enhances the performance of the proxy server because the number of processes limits the concurrent requests the proxy can handle. The following table can provide a starting point in determining the number of processes you will need.

Users	Processes	Memory (MB)	Swap (MB)
0 to 300	10	32	64
300 to 500	20	64	128
500 to 1000	30	96	192
More than 1000	40	128	256

While you can estimate the number of processes you will need, the proxy server should be properly scaled to meet your load in the peak periods to minimize delays. Typically, if you need to tune the server, your current process allocation is insufficient. The number of processes in use on sitemon will typically register 100 percent, but this does not show you the number of clients you want - those that are queued by the operating system.

You can estimate the number of clients in waiting by using netstat. Read your system's manual page for netstat, and determine the correct command line options to list all TCP sessions. From this snapshot, count all connections to port 8080, or your designated proxy port, that are in TCP states between SYN_RECVD and CLOSE_WAIT, including ESTABLISHED. (Your system may vary slightly, so check your documentation.) The count you now have should be a snapshot of accepted connections, those that have been established, plus any that have been queued by the system.

Run the log analyzer for a few days to get a good distribution of load on your server. The pstats utility located in `/extras/flexanlg/pstats` and available with Service Pack 2 for Web Proxy Server 3.6, is the fastest way to calculate these statistics. Pick a time when you have a moderate load: when your process

utilization is between 60 and 80 percent. It is important to consider that in peak times, under extreme load, connections will take longer than expected due to things like thrashing, swapping, or OS listen queue overload. You want to make sure that your system is not swapping, so this baseline configuration should not have more processes than can fit in RAM. For this time period, look at your average transaction time from the flex analyzer. Add 10 percent to this estimate if you want to be conservative.

The Sun ONE Web Proxy Server 3.6 SP2 Administrator's Guide contains a chart with the suggested number of processes given as a function of requests per second versus service time per request. Refer to this section for recommendations on determining the optimum number of processes for your server.

HTTP Keep-Alive

The proxy server supports HTTP keep-alive packets in order to provide improved performance on some systems. However, the majority of customers will see better performance with keep-alive connections turned off; this is the default setting.

Experience has shown that the benefit gained from keeping a connection open for a single client does not justify the penalty placed on subsequent requests from other clients. An open connection effectively ties up a process even if it is idle. Unless you have a small user base requesting primarily noncacheable content, the net effect on the proxy server of using keep-alives will be a performance decrease.

Monitoring the Servers

You will need to allocate time and resources to maintain your proxy server. It is important to monitor your proxy server to gauge performance and to identify any signs that you need to tune the server or deploy an additional one. You can monitor the status of your proxy server and keep it running smoothly by analyzing server log files, using server monitoring agents, and installing the latest patches and updates. To troubleshoot your installation, refer to the Sun ONE Web Proxy Server 3.6 SP2 Installation Guide.

Analyzing Logs

The proxy server creates log files to record all errors and access activity. You have the option of logging to several predefined formats, or you can specify a custom format. Of the predefined formats, extended-2 will provide the most information.

A log analyzer is included in the administration server and can be used to generate statistics and examine performance trends. Analyzing log files is the best method for examining cache efficiency, client access requirements, access patterns, throughput, and evaluation of intranet application performance. Log files also provide insight into current and future network requirements and intranet expansion strategies. To make the most of this data, rotate your log files on a regular basis and report usage statistics to end users.

To specifically access cache performance you should track three areas of the log file: cache finish status, routing information, and transfer time:

- Cache finish status can be divided into four categories: cache hits, cache writes, noncacheables, and errors. Record percentages for each of these categories and watch the trends over time.
- Routing information will show you the distribution between direct and indirect connections. Direct connections represent requests to local servers, while indirect requests are routed to the Internet through the web proxy or SOCKS.
- Total transfer time tells you how long the proxy server took to complete the request. Track the average transfer time for each proxy user base as well as an average transfer time by file size range. Breaking up the requests in this manner allows you to identify the percentage of requests by size and the average time for specific size ranges.

Monitoring Performance

Several tools are available to monitor the proxy server's performance. Probably the two most common are sitemon, on UNIX, and event viewer, on Windows NT. There may be other tools that are specific to your operating system. and you should feel free to use them.

As with most network hardware, the Simple Network Management Protocol (SNMP) can be used to monitor the proxy server's status. Details on how to use SNMP to monitor the proxy server are contained in the Administrator's Guide.

If you are running your Sun ONE Web Proxy Server on Windows NT, you can use the Performance Monitor that is automatically installed with the operating system. This tool will enable you to view the activity of your server in chart form.

When monitoring your proxy server with your tool set, watch the following areas:

- **Listenqueue**. On UNIX systems you need to monitor the queue for backups. Regular backups on this queue are an indication that you probably need to prefork additional processes.
- **Cache Utilization**.Keep track of the available space in your cache. If your existing cache begins to fill up, you need to expand the size of your cache by adding disks. You may also consider shortening the life cycle of documents in the cache, effectively eliminating old, seldom-accessed content to make room for new material.
- **CPU Utilization**. Monitor the proxy server's load on the CPU. If your CPU is heavily loaded on a regular basis, consider relieving some of the stress on the unit by adding another CPU to the server. Another alternative is to deploy an additional proxy server and balance the load between multiple machines.
- **Memory Utilization**. Monitor the amount of free RAM you have at peak times. Consider installing additional RAM if your performance begins to deteriorate due to insufficient memory allocation.

Installing Updates

When tracking the performance of your server installation, make sure you are working with the most recent software. Service Pack updates for Sun ONE Web Proxy Server are available online, so you always have access to the latest functionality. For more details, visit http://www.sun.com/software.

Planning for Growth

As your enterprise grows, so will your intranet. When monitoring your proxy server implementation, you can watch for signs that you need to upgrade or expand your current capabilities. Sun ONE Web Proxy Server is designed to allow you to easily add servers to your network.

Growth Issues

The following are some enterprise growth issues you may encounter and how you can handle them:

Have proxy services become strategic for your business?

If the proxy server is strategic to your organization, then high availability is going to be critical. Consider the benefits of deploying additional proxy servers for redundancy.

Have you saturated your network bandwidth?

If your network traffic is causing slowdowns, you can probably improve your situation by deploying additional proxy servers at the major network bottlenecks. The caching capabilities of additional proxy servers can help you reclaim lost bandwidth.

Is your CPU utilization too high?

If your CPU is maxed out, you can add another CPU to the machine or you can balance the load with additional proxy servers. You may be able to leverage existing hardware, without purchasing another CPU or a brand-new system. Sun ONE Web Proxy Server can be deployed on a wide array of platforms and system configurations.

Have you opened a new field office or added a department?

If the location and distribution of your user base has changed, you may need to deploy additional proxy servers. Your users will get the greatest benefit from locally cached content.

Has the type of content your users access changed?

If you see a trend in your organization toward heavy usage of certain content types, you should consider a dedicated proxy server. For example, if your users make a large number of FTP requests, you could benefit from a proxy server dedicated to FTP. Your clients can be configured to route to their regular proxy for other protocols and to a designated proxy for FTP transactions. Such dedicated proxy servers can be assigned by protocol, by site, or by other measures.

Licenses

Deploying another instance of Sun ONE Web Proxy Server may require you to purchase additional user licenses. Refer to the License Agreement for the most up-to-date information on product usage.

Software Updates

As we continue to improve Sun ONE Web Proxy Server, you may want to upgrade your software to take advantage of the latest functionality.

Contacting Sun Microsystems Technical Support

For product-specific Technical Support assistance, please see the Product Support Page at:

`http://www.sun.com/service/sunone/software/index.html`

Further Information

Further information can be found at the following Internet locations:

- Documentation

  `http://docs.sun.com/db/prod/s1.webproxys#hic`

- Consulting Services

  `http://www.sun.com/service/sunps/sunone/index.html`

- Developer Information

  `http://developer.iplanet.com`

- Sofware Training

  `http://www.sun.com/software/training/`

- Software

  `http://www.sun.com/software/`

- Product Data Sheet

  `http://www.sun.com/software/products/web_proxy/ds_web_proxy.html`

Sample Object Configuration File

```
# obj.conf
# You can edit this file, but comments and formatting changes
# might be lost when the admin server makes changes.
Init fn="load-types" mime-types="mime.types"
Init fn="init-proxy" timeout="1200" timeout-2="15"
Init fn="init-dns-cache" status="on" dir="/tmp" semas="4" size="512"
expire="3600"
Init fn="init-cache" status="on" dir="/export/home/cache/d1" ndirs="16"
Init fn="init-partition" status="on" dir="/export/home/cache/d1" max-size="1000"
min-avail="5" name="Part-1"
Init fn="init-urldb" status="on" dir="/export/home/cache/d1/urldb"
Init fn="init-batch-update" conf-file="bu.conf" status="off" dir="/tmp"
Init fn="init-partition" status="on" dir="/export/home/cache/d2" max-size="1000"
min-avail="10" name="Part-2"
Init fn="load-modules" funcs="ns-umatch-init,ns-umatch,ns-umatch-free"
shlib="/opt/ns-home/nsapi/examples/example.so"
Init fn="ns-umatch-init" file="/opt/ns-home/maplist" hashsize="5"
<Object name="default">
NameTrans from="http://.*~http://[^:/]*\\.netscape\\.com.*" fn="assign-name"
name="josh"
NameTrans fn="map" from="file:" to="ftp:"
NameTrans fn="pfx2dir" from="/ns-icons" dir="/opt/ns-home/ns-icons" name="file"
NameTrans fn="pac-map" from="/" to="/opt/ns-home/proxy-pac/pac/proxy.pac"
name="file"
PathCheck fn="url-check"
Service fn="deny-service"
AddLog fn="flex-log" iponly="1" name="access"
AddLog fn="urldb-record"
</Object>
<Object name="file">
PathCheck fn="unix-uri-clean"
```

```
PathCheck fn="find-index" index-names="index.html"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service fn="send-file"
</Object>
<Object ppath="ftp://.*">
ObjectType fn="cache-enable"
ObjectType fn="cache-setting" max-uncheck="21600"
Service fn="proxy-retrieve"
</Object>
<Object ppath="http://.*">
NameTrans fn="ns-umatch" ldomain="mcom.com" mode="handoff"
ObjectType fn="cache-enable"
ObjectType fn="cache-setting" lm-factor="0.100" max-uncheck="7200"
Service fn="proxy-retrieve"
</Object>
<Object ppath="https://.*">
Service fn="proxy-retrieve"
</Object>
<Object ppath="gopher://.*">
ObjectType fn="cache-enable"
ObjectType fn="cache-setting" max-uncheck="14400"
Service fn="proxy-retrieve"
</Object>
<Object ppath="connect://.*:443">
Service fn="connect" method="CONNECT"
</Object>
<Object ppath="connect://.*:563">
Service fn="connect" method="CONNECT"
</Object>
<Object name="josh">
ObjectType fn="cache-enable"
ObjectType fn="cache-setting" max-uncheck="0"
</Object>
```