



ViSOLVE

OPEN SOURCE SOLUTIONS

# *Implementing Reverse Proxy Using Squid*

*Prepared By  
Visolve Squid Team*



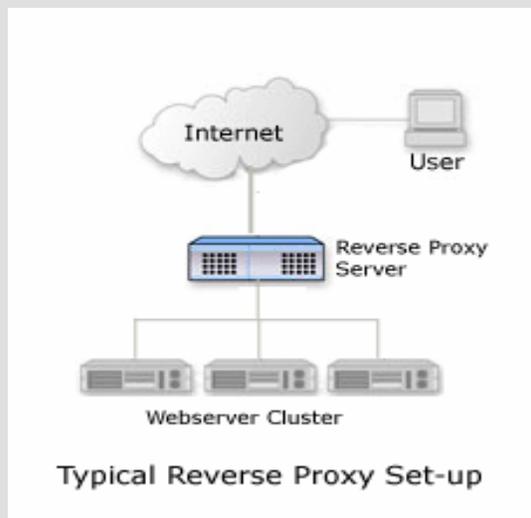
## Introduction

This document describes reverse proxies, and how they are used to improve Web server performance. Section 1 gives an introduction to reverse proxies, describing what they are and what they are used for. Section 2 compares reverse proxy caches with standard and transparent proxy caches, explaining the different functionality each provides. Section 3 illustrates how the reverse proxy actually caches the content and delivers it to the client. Section 4 describes how to configure Squid as a reverse proxy cache.

## What is Reverse Proxy Cache

Reverse proxy cache, also known as Web Server Acceleration, is a method of reducing the load on a busy web server by using a web cache between the server and the internet. Another benefit that can be gained is improved security. It's one of many ways to improve scalability without increasing the complexity of maintenance too much. A good use of a reverse proxy is to ease the burden on a web server that provides both static and dynamic content. The static content can be cached on the reverse proxy while the web server will be freed up to better handle the dynamic content.

By deploying Reverse Proxy Server alongside web servers, sites will:



- Avoid the capital expense of purchasing additional web servers by increasing the capacity of existing servers.
- Serve more requests for static content from web servers.
- Serve more requests for dynamic content from web servers.
- Increase profitability of the business by reducing operating expenses including

the cost of bandwidth required to serve content.

- Accelerate the response time of web and accelerate page download times to end users, delivering a faster, better experience to site visitors. When planning a Reverse-Proxy implementation the origin server's content should be written with the proxy server in mind, i.e. it should be "Cache Friendly". If the origin server's content is not "cache aware", it will not be able to take full advantage of the reverse proxy cache.

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.

The objective of this white paper is to explain the implementation of Squid as a Reverse-proxy also known as Web Server-accelerator. The basic concept of caching is explained followed by the actual implementation and testing of the reverse-proxy mode of squid.

## About Squid

Squid is an Open source high-performance Proxy caching server designed to run on Unix systems. National Science Foundation funds squid project, Squid has its presence in numerous ISP's and corporate around the globe. Squid can do much more than what most of the proxy servers around can do.

Reverse Proxy compared with other Proxy caches

There are three main ways that proxy caches can be configured on a network :

### Standard Proxy Cache

A standard proxy cache is used to cache static web pages (html and images) to a machine on the local network. When the page is requested a second time, the browser returns the data from the local proxy instead of the origin web server. The browser is explicitly configured to direct all HTTP requests to the proxy cache, rather than the target web server. The cache then either satisfies the request itself or passes on the request to the target server.

### Transparent Cache

A transparent cache achieves the same goal as a standard proxy cache, but operates transparently to the browser. The browser does not need to be explicitly configured to access the cache. Instead, the transparent cache intercepts network traffic, filters HTTP traffic (on port 80), and handles the request if the item is in the cache. If the item is not in the cache, the packets are forwarded to the origin web server. For Linux, the transparent cache uses iptables or ipchains to intercept and filter the network traffic. Transparent caches are especially useful to ISPs, because they require no browser setup modification. Transparent caches are also the simplest way to use a

cache internally on a network (at peering-hand off points between an ISP and a larger network, for example), because they don't require explicit coordination with other caches.

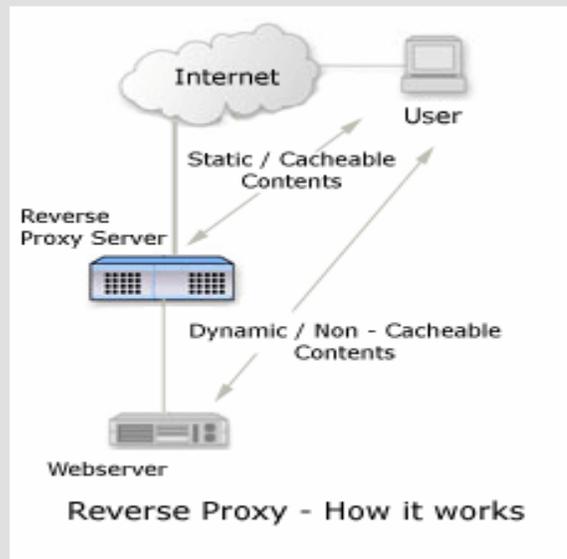
### Reverse Proxy Cache

A reverse proxy cache differs from standard and transparent caches, in that it reduces load on the origin web server, rather than reducing upstream network bandwidth on the client side. Reverse Proxy Caches offload client requests for static content from the web server, preventing unforeseen traffic surges from overloading the origin server. The proxy server sits between the Internet and the Web site and handles all traffic before it can reach the Web server. A reverse proxy server intercepts requests to the Web server and instead responds to the request out of a store of cached pages. This method improves the performance by reducing the amount of pages actually created "fresh" by the Web server.

### **How Reverse proxy caches work**

When a client browser makes an HTTP request, the DNS will route the request to the reverse proxy machine, not the actual web server. The reverse proxy will check its cache to see if it contains the requested item. If not, it connects to the real web server and downloads the requested item to its disk cache. The reverse proxy can only server cacheable URLs (such as html pages and images).

Dynamic content such as cgi scripts and Active Server Pages cannot be cached. The proxy caches static pages based on HTTP header tags that are returned from the web page



The four most important header tags are:

- Last-Modified: Tells the proxy when the page was last modified.
- Expires: Tells the proxy when to drop the page from the cache.
- Cache-Control: Tells the proxy if the page should be cached.

Pragma: Also tells the proxy if the page should be cached. For example, by default all Active Server Pages return "Cache-control: private." Therefore, no Active Server Pages will be cached on a reverse proxy server.

### **Configuring Squid as Reverse Proxy (http accelerator)**

To set up Squid as an httpd accelerator, you simply configure the squid.conf file. Usually it is found in either /usr/local/squid/etc, when installed directly from source code, or /etc/squid when pre-installed on Red Hat Linux systems. The squid.conf file is used to set and configure all the different options for the Squid proxy server. As root open the squid.conf file in your favorite text editor. If the real web server runs on a separate machine than the Squid reverse proxy, edit the following options in the squid.conf file :

```
http_port 80 # Port of Squid proxy
httpd_accel_host 172.16.1.115 # IP address of web server
httpd_accel_port 80 # Port of web server
httpd_accel_single_host on # Forward uncached requests to single host
httpd_accel_with_proxy on #
httpd_accel_uses_host_header off
```

If the web server runs on the same machine where Squid is running, the web server daemon must be set to run on port 81 (or any other port than 80). With the Apache web server, it can be done by assigning the line "Port 80" to "Port 81" in its httpd.conf file. The Squid.conf must also be modified to redirect missed requests to port 81 of the local machine :

```
http_port 80 # Port of Squid proxy
httpd_accel_host localhost # IP address of web server
httpd_accel_port 81 # Port of web server
httpd_accel_single_host on # Forward uncached requests to single host
httpd_accel_with_proxy on #
httpd_accel_uses_host_header off
```

We describe these options in greater detail.

`http_port 80`

The option `http_port` specifies the port number where Squid will listen for HTTP client requests. If this option is set to port 80, the client will have the illusion of being connected to the actual web server. This option should always be port 80.

`httpd_accel_host 172.16.1.115` and `httpd_accel_port 80`

The options `httpd_accel_host` and `httpd_accel_port` specify the IP address and port number of the real HTTP Server, such as Apache. In our configuration, the real HTTP Web Server is on the IP address 172.16.1.115 and on port 80.

If we are using the reverse proxy for more than one web server, then we must use the word `virtual` as the `httpd_accel_host`. Uncached requests can only be forwarded to one port. There is no table that associates accelerated hosts and a destination port. When the web server is running on the same machine as Squid, set the web server to listen

for connections on a different port (8000, for example), and set the `httpd_accel_port` option to the same value.

`httpd_accel_single_host on`

To run Squid with a single back end web server, set `httpd_accel_single_host` option to on. Squid will forward all uncached requests to this web server regardless of what any redirectors or Host headers says. If the Squid reverse proxy must support multiple back end web servers, set this option to off, and use a redirector (or host table or private DNS) to map the requests to the appropriate back end servers. Note that the mapping needs to be a 1-1 mapping between requested and backend (from redirector) domain names or caching will fail, as caching is performed using the URL returned from the redirector. See also `rewrites_host_header`.

`httpd_accel_with_proxy on`

If one wants to use Squid as both an `httpd` accelerator and as a proxy for local client machines, set the `httpd_accel_with_proxy` to on. By default, it is off. Note however that your proxy users may have trouble reaching the accelerated domains, unless their browsers are configured not to use the Squid proxy for those domains. The `no_proxy` option can be used to direct clients not to use the proxy for certain domains.

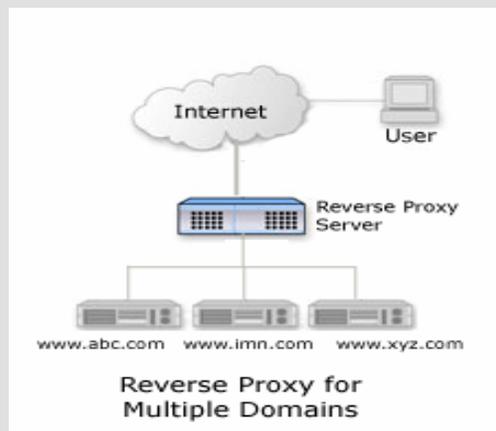
`httpd_accel_uses_host_header off`

Requests in HTTP version 1.1 include a Host header, specifying the host name (or IP address) of the URL. This option should remain off in reverse proxy mode. The only time this option must be set to on is when Squid is configured as a Transparent proxy.

It's important to note that `acls` (access control lists) are checked before this translation. You must combine this option with strict source-address checks, so you cannot use this option to accelerate multiple back end servers.

### **Configuring Squid as Reverse Proxy for Multiple Domains (http accelerator)**

You can configure squid in an accelerator mode for multiple domains also. For example you can configure single squid machine for `www.abc.com`, `www.xyz.com`, `www.lmn.com`.



Squid configuration

httpd\_accel\_host virtual

httpd\_accel\_port 80 (the web server port)

httpd\_accel\_single\_host off (It should be disabled when we are going to the reverse proxy for multiple servers)

httpd\_accel\_uses\_host\_header on

Note: When you compile Squid enable the Internal DNS option

To set the reverse proxy for the domain

www.abc.com 192.168.1.2 www.xyz.com 192.168.1.2 www.lmn.com 192.168.1.2

Let reverse\_proxy server ip be 192.168.1.2

DNS entry in the reverse proxy server

You need to configure Intranet DNS and Internet DNS. You can configure split DNS if you want both the DNS in same machine. Instead of Intranet DNS you can have domain entries in /etc/hosts. You have to configure squid with --disable-internal-dns to use /etc/hosts file lookup.

Internal DNS entry

www.abc.com IN A 172.16.1.2 www.xyz.com IN A 172.16.1.3 www.lmn.com IN A 172.16.1.4

Note: If you have compiled with disable internal dns, then add the entry in the /etc/hosts like

172.16.1.2 www.abc.com 172.16.1.3 www.xyz.com 172.16.1.4 www.lmn.com

Internal DNS entry

www.abc.com 192.168.1.2 www.xyz.com 192.168.1.2 www.lmn.com 192.168.1.2

**References**

- [ViSolve Squid Configuration Manual 2.4](#)
- [ViSolve Squid Configuration Manual 3.0](#)

## Conclusion

Reverse proxying is a special proxy deployment used to reduce load on a web server. The reverse proxy server is placed outside the firewall, acting as the web server to external clients. Cached requests are sent back directly to the clients without any computation from the actual web server. Uncached requests must be forwarded to the backend web server, and the response from the web server is then cached in the reverse proxy.

## About ViSolve.com

ViSolve is an international corporation that provides technical services, for Internet based systems, for clients around the globe. ViSolve is in the business of providing software solutions since 1995. We have experience of executing several major projects and we are now completely focused on leading Internet technologies, Testing QA and support. We are committed to the Open source movement and in the same lines we provide free support for products like Linux, Apache and Squid to the user community.

Document Version : 1.0 | [Created On](#) : 28-01-02 | [Updated On](#) : 29-05-06