ViSolve
OPEN SOURCE SOLUTIONS

Transparent Cache

Implementation Using Squid

Prepared By
Visolve Squid Team

Abstract	
	Internet traffic is growing at a phenomenal rate, and such rapid increase in network traffic has created numerous networking challenges for ISPs and enterprises, like increased bandwidth cost for ISP's, bandwidth congestion, poor response time for end users and cost-efficient network / customer base scalability. The most efficient solution to these networking problems is to use your existing network infrastructure to localize traffic patterns, enabling content requests to be fulfilled locally. Increased speed/ decreased latency without the cost of additional bandwidth have catapulted caching software and appliances into a prominent place among the fastest growing segments of Internet technology.

Introduction	
	A transparent cache is so named because it works by intercepting the network traffic transparently to the browser. In this mode, the cache short-circuits the retrieval process if the desired file is in the cache. 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, because they do not require explicit coordination with other caches. The purpose of this white paper is to discuss the various methods of implementing transparent caching using Squid on Linux with a policy based router, an externalL4 switch, and an L4 switch inside the Linux Squid box. First, some basic concepts will be discussed, followed by the advantages of transparent caching, and finally redirecting packets to Squid using IP-Chains.

What is transparent caching?	
	The full explanation about the term "Transparent Caching and Transparent Proxying" depends on the context, but we can assume the context here is HTTP proxy/caches with transparent hijacking of port 80, which is the default HTTP traffic in the internet.

The difference is that the cache includes a cache, while the proxy only proxies without caching. The term transparent is overloaded, having different meanings depending on the situation. To some it means a setup that hijacks port 80 traffic where the client tried to go to other servers, to some it means a semantically transparent proxy that does not change the meaning or content of requests/replies. There is no such thing as a truly transparent proxy, only semitransparent and certainly not such a thing as a truly transparent cache. Squid can be configured to act transparently. In this mode, clients are not required to configure their browsers to access the cache, but Squid will transparently pick up the appropriate packets and cache requests. This solves the biggest problem with caching: i.e. getting users to use the cache server.

Advantages of Transparent Caching

As might be expected, the advantages and disadvantages of transparent caching are largely the reverse of those cited for proxy caching. In the advantages category we |

have the following :

o Simplified administration - The browser does not need to be configured to talk to a cache.
o Central control - The user cannot change his/her browser to bypass the cache.

Disadvantages of Transparent Caching

• Not Robust - Because transparent caching relies on stable routed path between the client and the origin server which happens to pass through a "cached path," it is susceptible to routing changes in the Internet. In other words, if a connection between a client and a cache is established and a routing change occurs which causes the client to take a path which no longer flows through the "diverting" network device, the session will break and the user will have to reload the page. If routes in the Internet are flapping, then results will be even more unpredictable.
• User control - Transparent caching takes control away from the user. Many users have very strong biases about caching and will actually change ISPs to either avoid it or get it.

Browser dependency - For successful operation, many transparent caches rely on the browser supplying the host name of the origin server in the HTTP request header. This is required because these caches cannot access the destination IP address of the origin server from the IP address of the packet. Therefore, upon a cache miss, they cannot determine the origin server address to send the request to. Some early browsers do not provide this information and therefore will not work properly with these transparent caches, but 90% of today's browsers satisfy the above. In the real world, Many network providers have observed that a significant amount of HTTP requests are for non-cacheable content (as much as 35-45%). The hit rate and performance of the cache is inversely proportional to the amount of non-cacheable content sent to the cache.

How to implement transparent caching using squid?	Transparent caching can be implemented by three ways. • Policy based routing. • Using smart switching. • By setting Squid Box as a Gateway.

Transparent caching using policy based routing	This arrangement uses a router to route WWW traffic (via policy routing) to the Squid cache box. Because the router can change only the IP address of a packet, the Squid Linux box must be configured to redirect the destination port of the packet. The Router policy redirects packets with port 80 to the Squid box and redirects other traffic to the Internet directly. To set the router policy rules, refer to your router's manual. Using the IP-Chains tool in the Squid box, one can redirect packets which are sent by router to the Squid application. See later chapters for more details about configuring IP-Chains. Since some routers (e.g. Cisco series) do not recognize Squid cache failures, if Squid does malfunction, service to the WWW breaks. To overcome

this problem, a cache guard (a Perl script running on the computer inside serviced network) can be use to regularly query the Squid box for a cached object. When the cache guard fails repeatedly to retrieve the object from the cache, the cache guard changes the router configuration (by SNMP) to pass the WWW traffic directly to the Internet. In this way, a fail over strategy can be implemented.

Transparent Caching - Policy based routing
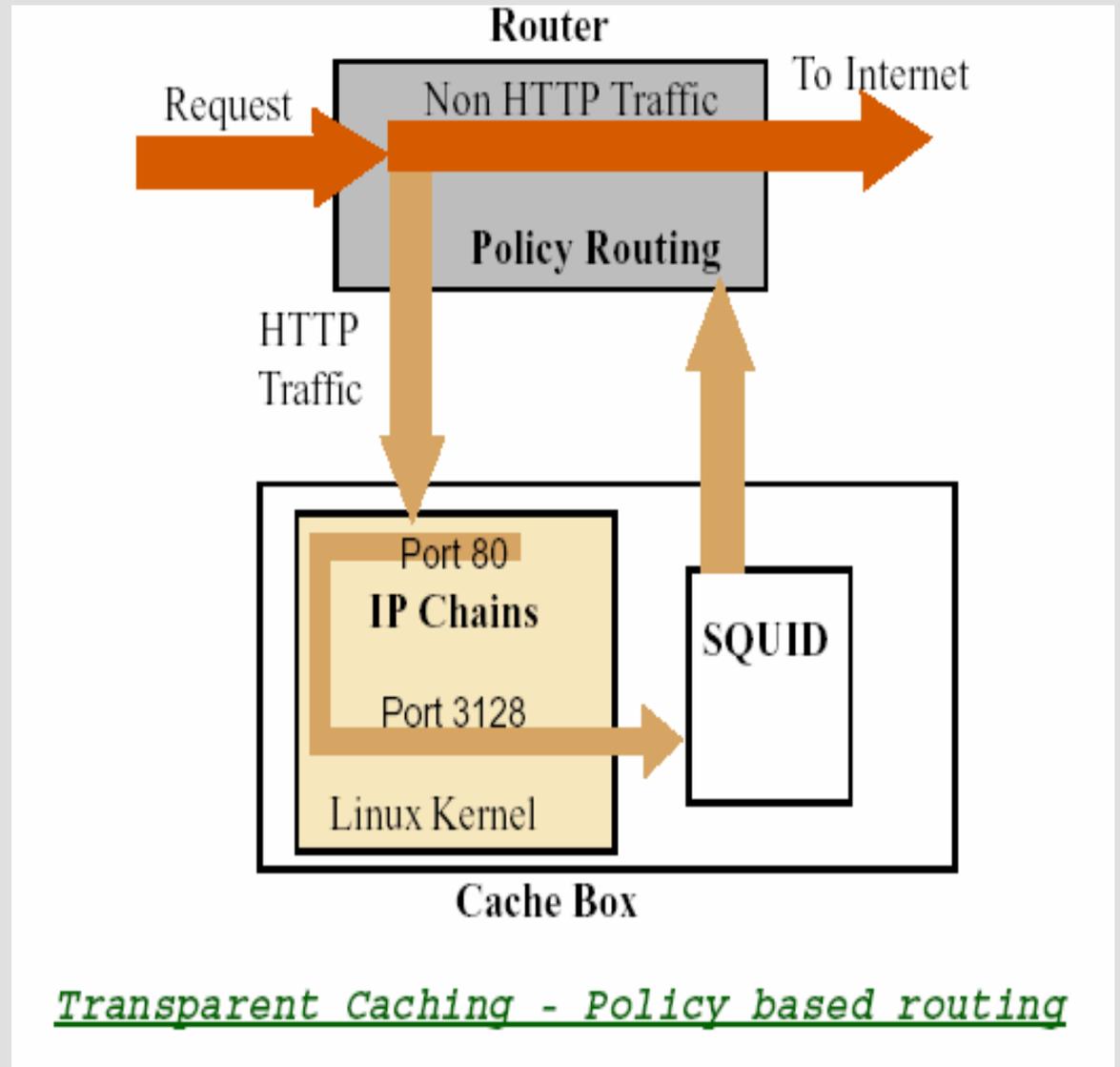
Transparent caching using smart switching	This arrangement uses a Layer 4 or Layer 7 router to route WWW traffic to the transparent Squid cache. Because the router can change only the IP address of a packet, the Squid Linux box must be configured to redirect the destination port of the packet. Both L4 and L7 switches intercept outgoing traffic and pass HTTP requests, typically port 80 traffic, to the squid proxy server that the switch is configured to recognize. The switch forwards non- HTTP traffic to other destinations. The architecture shows a switch passing HTTP traffic to the Squid proxy server and non-HTTP traffic to the Internet

	To Internet **ROUTER** Non HTTP Request **L4 / L7 switch** Rules set in the switch HTTP Port 80 **IP Chains** Port 3128 Linux Kernel **SQUID** **Cache Box**
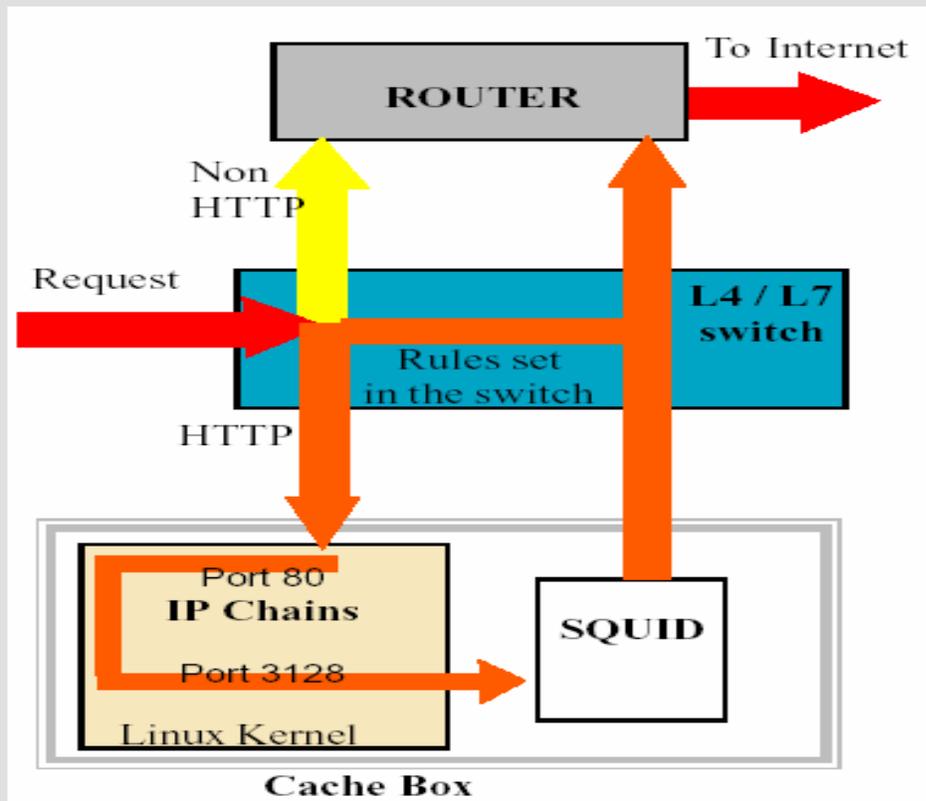
How Switch Operates?	L4 and L7 switches derive their names from the level of the Open Systems Interconnection (OSI) Reference model at which they operate. The capabilities of these switches are determined by the layer in the OSI model at which they operate. • L4 SWITCH - An L4 switch operates at Layer 4 in the OSI model - the Transport layer. L4 switches base their switching decisions on information in the TCP header, and TCP is a protocol that resides at Layer 4 in the OSI seven-layer model. These switches determine where to pass the traffic based on the port number. • L7 SWITCH - At the time of this writing, more sophisticated switches are becoming available. These new switches operate at Layer 7 of the OSI model - the Application layer. Because these switches operate at Layer 7, they can understand URLs and can understand much more about the traffic than an L4 switch can. An L7 switch provides the same features that an L4 switch provides plus additional, more sophisticated features.

Comparing L4 and L7 Switches	An L7 switch has the same features that an L4 switch has, plus additional, more sophisticated features, as described in this section.

Similar features

- Some L4 and L7 switches can switch more than a gigabyte of data.
- For HTTP transparent caching, they partition traffic based on the requested Web server's IP address.

- For HTTP transparent caching, they can be configured to send traffic directly to the Internet if a Web cache fails.

How the L7 switch is different

- An L7 switch can partition HTTP client traffic based on the requested URL.
- For HTTP requests, the L7 switch can look at the request and determine whether the object is cacheable. With an L7 switch, requests for obviously non-cacheable objects, such as URLs with cookies and CGI, will bypass the cache. Non-cacheable objects are then obtained directly from a Web server.

Performance comparison between L4 and L7 switches :

- The performance of L4 and L7 switches is similar. However, because the L7 switch looks more closely at TCP/IP packets for port 80 or port 119, its response time is slightly slower than that of an L4 switch.

Squid box as a gateway	This setup is used in small LAN or WAN where number of clients are less. Here it is mandatory to configure Squid box as a default Gateway in all machines. This method requires more configurations in the Squid box as compared to the other methods.

Squid box Configuration

Steps to be followed to implement Transparent caching.

- Packets headed for port 80 on some computer on the Internet must be redirected by the router or L4 switch (As explained before) to the computer where squid is running. This can be achieved by setting squid box as a Gateway also.

In Squid Box, packets which are redirected by a smart switch or router to the Squid box still need to be redirected to the port where Squid is listening on. Redirecting these packets cannot be done by Squid. Redirecting packets must be done by the Linux kernel, using the IP-chains program. The kernel then receives a packet on port 80, looks at the firewall configuration, and adjusts the packet appropriately i.e. by changing the destination port to 3128, or whatever port Squid is running on. If you need IP Filter redirection, then use the -enable-ipf-transparent configure option in Squid to support certain HTTP clients (HTTP/1.0 clients, NOT sending the Host header). However, normal browsing using the popular browsers will work even without it.
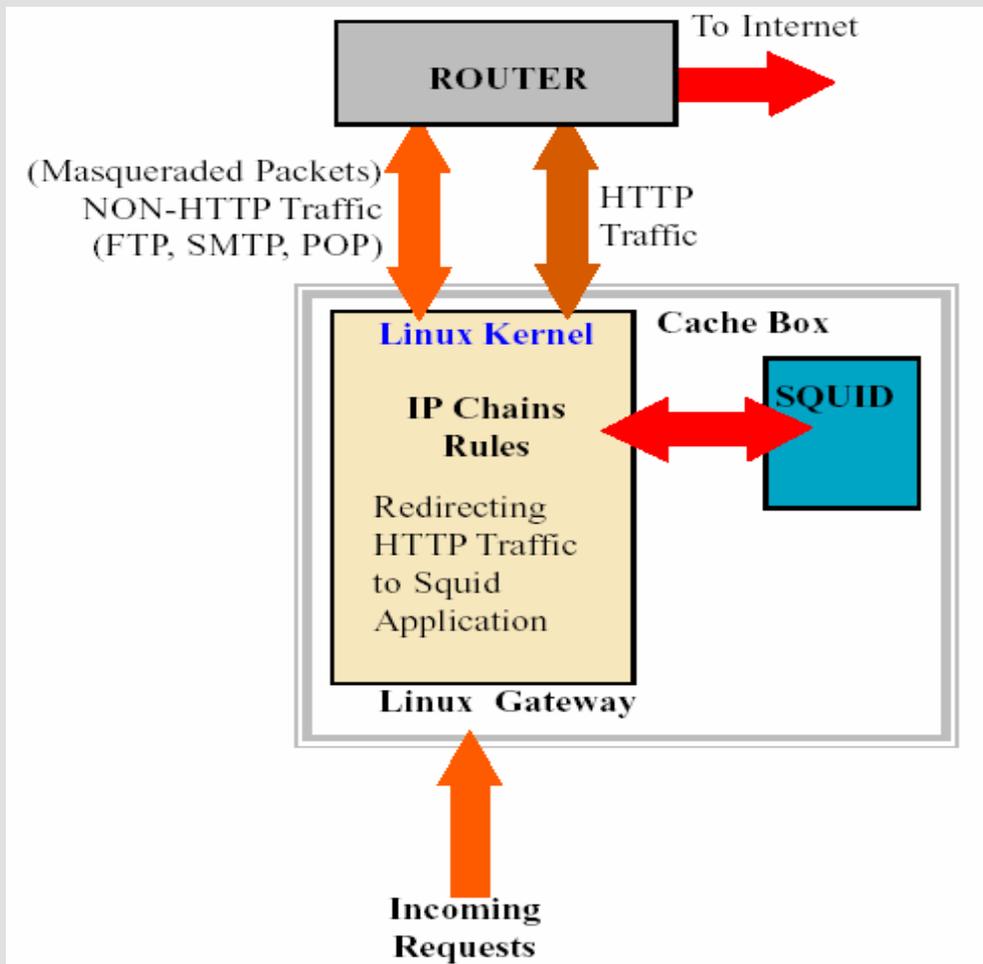
| **About IP Chains** | Ipchains is an extremely powerful program that allows the user to set up complex IP filtering and accounting rules.

Purpose :
To set up a firewall in the Squid/Linux box with the minimal options needed for transparent proxy. Here is the simplest method.

Details :
Make sure that the following options in the kernel are enabled.

- CONFIG_PACKET
- CONFIG_NETLINK
- CONFIG_RTNETLINK
- CONFIG_NETLINK_DEV
- CONFIG_FIREWALL
- CONFIG_FILTER
- CONFIG_UNIX |

- CONFIG_INET
- CONFIG_IP_MULTICAST
- CONFIG_IP_ADVANCED_ROUTER
- CONFIG_RTNETLINK
- CONFIG_NETLINK
- CONFIG_IP_MULTIPLE_TABLES
- CONFIG_IP_ROUTE_MULTIPATH
- CONFIG_IP_ROUTE_TOS
- CONFIG_IP_ROUTE_VERBOSE
- CONFIG_IP_ROUTE_LARGE_TABLES
- CONFIG_IP_ROUTE_NAT
- CONFIG_IP_FIREWALL
- CONFIG_IP_FIREWALL_NETLINK
- CONFIG_NETLINK_DEV
- CONFIG_IP_ROUTE_FWMARK
- CONFIG_IP_TRANSPARENT_PROXY
- CONFIG_IP_MASQUERADE
- CONFIG_IP_MASQUERADE_ICMP
- CONFIG_IP_ROUTER

Else you must recompile the kernel. Also, make sure IP-forwarding is enabled in the kernel using the following command.

cat /proc/sys/net/ipv4/ip_forward

This should return 1. Else, do the command

echo 1 > /proc/sys/net/ipv4/ip_forward

PORT Redirection

The following command enables transparent caching :

ipchains -A input -j REDIRECT 3128 -p tcp -s 0.0.0.0/0 -d 0.0.0.0/0 80

This command redirects all the requests, irrespective of source IP Addresses, with destination port 80 to destination port 3128 in which Squid (in Transparent mode) is running.

IP-Masquerading

This is essential when the third method is implemented, where as this is not applicable for the other two methods. When squid is in transparent mode, the local network will not be able to access other protocols available on Squid. Squid supports ftp, gopher, and https only when clients of each specified protocol are aware of the cache. Hence this is not possible when squid is in transparent mode. Here, IP-Masquerading can be used to enable access to other protocols. The following are the rules to masquerade the protocols SMTP, FTP, POP, SSH, TELNET, and HTTPS. Here, assume the Squid box is connected to the Internet through a router using the eth1 interface.

- ipchains -N good-bad

// New-User defined Rule is declared

ipchains -A forward -s 172.16.1.0/24 -i eth1 -j good-bad

// good-bad rule is added to the ipchains rule table. This is forwarding all the requests coming from the source 172.16.1.1 -254 to the interface through which internet is connecting to.

// In the following set of lines define the user defined rule good-bad

ipchains -A good-bad -p tcp -dport ssh -j MASQ

ipchains -A good-bad -p tcp -dport telnet -j MASQ

ipchains -A good-bad -p tcp -dport ftp -j MASQ

ipchains -A good-bad -p tcp -dport smtp -j MASQ

ipchains -A good-bad -p tcp -dport 110 -j MASQ

For more information about ipchain configurations visit us at
http://squid.visolve.com/squidconf.html

About IP table	The iptables module (for kernel 2.4.x series and above) which is a part of the Netfilter framework is a good upgrade of old ipchains(for kernel 2.2.x).

Kernel setup

To run the pure basics of iptables the following options are to be configured into the kernel :

CONFIG_PACKET
CONFIG_NETFILTER

And of course your interfaces are needed to be configured properly to work, ie. Ethernet, PPP and SLIP interfaces. The following are to be set in the kernel if more advanced options are needed :

- CONFIG_IP_NF_CONNTRACK
- CONFIG_IP_NF_FTP
- CONFIG_IP_NF_IPTABLES
- CONFIG_IP_NF_MATCH_LIMIT
- CONFIG_IP_NF_MATCH_MAC

- CONFIG_IP_NF_MATCH_MARK
- CONFIG_IP_NF_MATCH_MULTIPORT
- CONFIG_IP_NF_MATCH_TOS
- CONFIG_IP_NF_MATCH_STATE
- CONFIG_IP_NF_MATCH_UNCLEAN
- CONFIG_IP_NF_MATCH_OWNER
- CONFIG_IP_NF_FILTER
- CONFIG_IP_NF_TARGET_REJECT
- CONFIG_IP_NF_TARGET_MIRROR
- CONFIG_IP_NF_NAT
- CONFIG_IP_NF_NAT_NEEDED
- CONFIG_IP_NF_TARGET_MASQUERADE
- CONFIG_IP_NF_TARGET_REDIRECT
- CONFIG_IP_NF_NAT_FTP

Port redirection

iptables -t nat -A PREROUTING -p TCP --dport 80 -j REDIRECT --to-port 3128

The above rule redirects port 80 requests, irrespective of source ip address to port 3128 (or whichever port in which squid is running in transparent mode).

IP-Masquerading

iptables -t nat -A POSTROUTING -p TCP -s 0/0 --dport 21 -j MASQUERADE

iptables -t nat -A POSTROUTING -p TCP -d 0/0 --dport 20 -j MASQUERADE

iptables -t nat -A POSTROUTING -p TCP --dport 25 -j MASQUERADE

iptables -t nat -A POSTROUTING -p TCP --dport 110 -j MASQUERADE

iptables -t nat -A POSTROUTING -p TCP --dport 22 -j MASQUERADE

iptables -t nat -A POSTROUTING -p TCP --dport 23 -j MASQUERADE

The above rules are essential when we connect modem or squid is in between two different network to make TELNET, FTP, SMTP, POP, HTTPS to communicate to INTERNET.

Squid in transparent mode	
	To Run Squid in a transparent mode, enable the following directives in Squid.conf.
	httpd_accel_host virtual
	httpd_accel_port 80
	httpd_accel_with_proxy on

	httpd_accel_uses_host_header on The httpd_accel_port directive tells which port the origin server is listening on (port 80). Squid does not need to know how requests arrive at its listening port (3128). This must be done by the operating system or router. Squid sees a request for a URL and connects to port 80 on the server where it thinks the URL resides. Squid does not have any control over what types of request arrive to it. If Squid is listening to port 3128 then it assumes the data arriving there is a protocol it can handler (HTTP, FTP, etc). The type of packets that are redirected to Squid is determined entirely by the TCP/IP implementation of the host (i.e. ipchains forwarding) and is out of Squid's control.

Recommended Hardware for Transparent Caching	**Processor :**Intel P3 550MHz CPU **Hard Drive :**For high performance and stability, a SCSI disk is highly recommended or use UDMA 66 Drive instead of IDE Disk. Typically 9 GB Disks are preferred. **Ethernet :**High performance Ethernet is preferred. **RAM :**For every 1 GB cache, 10 MB of RAM is required. For the above case, Minimum of 300 MB is required preferably 512 MB RAM.

Comparison	Policy based routing Advantages: • Higher router processor load. On a middle sized site (about 1,000 computers), the CPU load on Cisco 7500 varies from 30 to 70 %. If you have about 3,000 computers, you simply can't use this. • Lack of scalability. • Not too elegant fail over mechanism. Disadvantages : • Low-cost solution. • When we use WCCP in router for fail over and having multiple no of caches, WCCP dynamically balances HTTP requests over available Squid proxy servers. • Using smart switching Advantages : • Fail over : For HTTP transparent caching, if a Squid proxy server is down or is too busy, the switch passes the traffic to the Internet or, if there are multiple Squid Proxy Servers, to another Squid proxy server it is configured to recognize. • HTTP request bypass of a Squid proxy server: For HTTP transparent caching, you can configure an L4 or L7 switch to prevent IP addresses from bypassing the Squid proxy server. It can be also configured a switch or used Squid proxy server

Configuration options to allow requests to some IP addresses to bypass Squid proxy server. For example, a carrier implementing Squid proxy server might want to allow hosts of customers who do not want to pay for caching services to bypass the cache.

- Advanced caching system: Instead of diverting all HTTP (port 80) to the cache, the Arrow Point Content Smart Web switches inspect the HTTP request header, learning what content is being requested. Arrow Point Content Smart Web switching enables Web caching to achieve tremendous potential to improve the response of the Web to users, reduce bandwidth costs, and simplify content replication strategies. The switches provide additional levels of redundancy and scalability, as well as simplified administration, required for the mission-critical deployment of caching Performance measurements for Layer 4 switches include the speed with which the switch can set up new TCP sessions (new flows per second), including latency for the flow setup, and the speed of data transfer in the packet forwarding mode (pps).

Disadvantages:

- High Cost

Comparison of using a router to using an L4 or L7 switch

For many routers, complex filters, such as a filter for intercepting HTTP (port 80) or NNTP

- (port 119) requests, can have a dramatic negative impact on the performance of the router. Conversely, L4 and L7 switches are designed to intercept packets of different types. With a policy-based router (non-Cisco router or a Cisco router not running WCCP), the system administrator must manually set up how requests will be distributed, which might result in less efficient partitioning of requests than if a switch were used.

- With a policy-based router, if the Squid proxy server goes down, caching services are unavailable because a router cannot be configured to fail over to the Internet for HTTP requests. Similarly, the router cannot be configured to fail over the news server for NNTP requests (although this option is not viable typically because many news servers cannot handle the volume of connections that a Squid proxy server can). However, even with the workaround, fail over capabilities are not as good as with an L4 switch.

Squid box as a Gateway

Advantages:

- Low cost of implementation
- Less configuration
- Even without router Squid box could be connected with internet with additional hardware in Linux.
- Firewall support.

	Disadvantages:
	• It is beneficial only for small LAN and WAN users.

Conclusion	This paper has outlined the various methods of implementing Transparent Caching using Squid. Each of these methods has its advantages, the choice is left to the implementation team which has to decide based on their network, data access pattern, volume of data, request rate, criticality and budget available. Web caching is a matured technology and Squid is very widely used web caching application, the choice and method of implementation as said may vary, although other features present in the implementation may continue or be enhanced, the underlying fundamentals will be the same as those discussed here. There are other tools available to supplement the system like reporting tools, configuration and management tools and load balancing for implementing multiple cache boxes. And finally the overall success largely depends on the configuration and fine-tuning of both Squid and Linux.

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.