Squid @VISOLVE.com

Squid Configuration Manual
for Squid 2.4

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. We'd welcome the opportunity of providing technical services(7x24 Interactive Support), for Internet based systems, for clients around the globe. For more Resources on Squid development kindly refer www.squid-cache.org

Table of Contents

- cache_peer
- cache_peer_domain
- neighbor_type_domain
- icp_query_timeout (msec)
- maximum_icp_query_timeout (msec)
- mcast_icp_query_timeout (msec)
- dead_peer_timeout (seconds)
- hierarchy_stoplist
- no_cache

III. **Cache size**

This section describes parameters related to Cache Memory Size (real memory) as well as Cache Replacement Policy. Squid supports more than one Cache replacement policy. This section also touches briefly on cache interaction with disk, but that subject is covered in greater detail in the next section. Relevant parameters for this section are

- Cache_mem (bytes)
- cache_swap_low (percent, 0-100)
- cache_swap_high (percent, 0-100)
- maximum_object_size (bytes)
- minimum_object_size (bytes)
- maximum_object_size_in_memory (bytes)
- ipcache_size (number of entries)
- ipcache_low (percent)
- ipcache_high (percent)
- fqdncache_size (number of entries)
- cache_replacement_policy
- memory_replacement_policy

IV. **Log file path names and cache directories**

This section describes parameters for the configuration of cache directories (placement of caches in specific files and directories) AND Log file placement on disk (size, name, path, activity). Log files contain runtime information (relevant "successful" events as well as errors). Log files are used for system level debugging and runtime activity.

Certain log file management issues, e.g. backups, restore, recycling etc. are not described here. For more information on controlling the log file size, see logfile_rotate directive, Squid command line option (-k rotate) and man page on logrotate in Linux.

- Cache_dir
- cache_access_log
- cache_log
- cache_store_log
- cache_swap_log
- emulate_httpd_log on|off
- log_ip_on_direct
- mime_table
- log_mime_hdrs on|off
- user agent_log
- referer_log
- pid_filename
- debug_options
- log_fqdn on|off
- client_netmask

V. **Support for External functions**

v. **Support for External functions**
Squid has the ability to invoke certain "externally defined' functions that are NOT part of the Squid binary. Such "external" executables (programs) are usually placed in a contrib directory for source code distribution. The most common "external" programs are FTPUser, DNS, Redirectors and Authenticators, and are usually contributed by sources other than Squid.

External programs are invoked by Squid through the standard fork() and exec(). The number of such fork-able child processes for specific "external" processes can also be defined. Relevant parameters for this section are:

- ftp_user
- ftp_list_width
- ftp_passive
- cache_dns_program
- dns_children
- dns_retransmit_interval
- dns_timeout
- dns_defnames on|off
- dns_nameservers
- unlinkd_program
- pinger_program
- redirect_program
- redirect_children
- redirect_rewrites_host_header
- redirector_access
- authenticate_program
- authenticate_children
- authenticate_ttl
- authenticate_ip_ttl
- authenticate_ip_ttl_is_strict

VI. **Tuning the Squid Cache**
This section describes the important parameters that determine Squid cache performance. Notable among them are: Object refresh algorithm, size of the header and body for both reply and request, policy for aborting (server) connections when client closes connection etc. Relevant parameters are described below:

- wais_relay_host
- wais_relay_port
- request_header_max_size (KB)
- request_body_max_size (KB)
- reply_body_max_size (KB)
- refresh_pattern
- reference_age
- quick_abort_min (KB)
- quick_abort_max (KB)
- quick_abort_pct (percent)
- negative_ttl time-units
- positive_dns_ttl time-units
- negative_dns_ttl time-units
- range_offset_limit (bytes)

VII. **Timeouts**
Timeout parameters in Squid can be based on overall connection timeouts, peer-specific timeouts, site/domain-specific timeouts, request-specific timeouts etc. Proper setting of timeout values is critical to optimal Squid performance. Relevant parameters for timeout settings are listed here.

By adding a *translation* layer into Squid, we can accept (and understand) web requests, since the format is essentially the same. The additional layer can re-write incoming web requests, changing the destination server and port. This re-written request is then treated as a normal request: the remote server is contacted, the data requested and the results cached. This lets Squid to pretend to be a web server, rewriting requests so that, they are passed on to some other web server.

For Transparent caching, Squid can be configured to *magically* intercept outgoing web requests and cache them. Since the outgoing requests are in web-server format, it needs to translate them to cache-format requests. Transparent redirection is prohibited by internet standard #5 "Internet Protocol". And HTTP assumes no transparent redirection is taking place.

This section allows various configurations related to the accelerator mode and the transparent mode.

- httpd_accel_host
- httpd_accel_port
- httpd_accel_single_host
- httpd_accel_with_proxy on|off
- httpd_accel_uses_host_header on|off

XII. **Miscellaneous**

As the title suggests, this section covers parameters that could not be explicitly bundled in with any of the previous categories. Examples of features covered here are:

- Limiting the growth of log fils.
- Displaying customized information to clients upon error conditions or access denial.
- Defining memory pools for Squid.
- Network management by enabling SNMP.
- Co-ordination with neighbor caches by enabling WCCP and
- Directing the requests either to the origin server or to the neighbor cache.

The relevant parameters are:

- dns_test names
- logfile_rotate
- append_domain
- tcp_recv_bufsize (bytes)
- err_html_text
- deny_info
- memory_pools on|off
- memory_pools_limit (bytes)
- forwarded_for on|off
- log_icp_queries on|off
- icp_hit_stale on|off
- minimum_direct_hops
- minimum_direct_rtt
- cachemgr_passwd
- store_avg_object_size (kbytes)
- store_objects_per_bucket
- client_db on|off
- netdb_low
- netdb_high
- netdb_ping_period
- query_icmp on|off
- test_reachability on|off
- buffered_logs on|off

XIII. **Delaypool parameters (all require delay_pools compilation options)**

Conceptually, Delay pools are bandwidth limitations - "pools" of bandwidth that drain out as people browse the Web, and fill up at a rate we specify - this can be thought of as a leaky bucket that is continually being filled. This is useful when bandwidth charges are in place, if we want to reduce bandwidth usage for web traffic.

Delay Pools can do wonders when combined with ACLs. These tags permit us to limit the bandwidth of certain requests, based on any criteria. Delay behavior is selected by ACLs (low and high priority traffic, staff Vs students or student Vs authenticated student or so on). In ISPs, delay pools can be implemented in a particular network to improve the quality of service. To enable this, Squid needs to be configured with the --enable-delay-pools option.

Relevant parameters are described below:

NETWORK

Tag Name **http_port**

Usage
```
http_port port
hostname: port
1.2.3.4 : port
```

Description
This tag name is used to specify the socket addresses where Squid will listen for HTTP client requests. Multiple socket addresses can be specified. There are three forms: port alone, hostname with port, and IP address with port. If hostname or IP address is specified, then Squid binds the socket to that specific address. This replaces the old "tcp_incoming_address" option. Most likely, there is no need to bind to a specific address, so the port number alone can be used. If Squid is to be run in accelerator mode, then it should listen on port 80 also, or instead.

Default
```
http_port 3128
```

Example
Give the port number in which you want squid to listen to http client requests. Like...
http_port 8080
We can override the default port number by '-a ' command line option.
#/usr/local/squid/bin/squid -a 8080

This will start squid with port 8080, which overrides the port number in squid.conf.
However this option cannot be used to override IP address

Caution
Before changing the port number, make sure no application in your box is running in the same port.

Note
http_port can be used to specify the tcp_incoming_address through which the cache listens to requests from other remote servers. http_port can be listed multiple times.

Tag Name **icp_port**
Usage `icp_port port`

Description
This specifies the port number from which Squid sends and receives ICP queries to and from neighbor caches. To disable "0" is used.

ICP is a protocol used for communication among squid caches. ICP is primarily used within a cache hierarchy to locate specific objects in sibling caches. If a squid cache does not have a requested document, it sends an ICP query to its siblings, and the siblings respond with ICP replies indicating a ``HIT'' or a ``MISS.'' The cache then uses the replies to choose from which cache to resolve its own MISS. ICP is currently implemented on top of UDP. Squid also supports ICP via multicast.

Default `icp_port 3130`

Example
The port number is given in which squid has to send and receive ICP queries from neighbor caches. Like...

icp_port 5050
May be overridden by -u command line option.

#/usr/local/squid/bin/squid -u 5050
This will start squid with port 5050, which overrides the port number in squid.conf

Caution
Before changing this port number, make sure no application in the box is running in the same port.

Tag Name **htcp_port**
Usage `htcp_port port`

Description
Used to specify the port number through which Squid sends and receives HTCP queries to and from neighbor caches. To disable "0" is used.

Default `htcp_port 4827`

Example
htcp_port 5089

Caution
To enable this option, you must use --enable-htcp with the configure script.

Tag Name **mcast_groups**
Usage `mcast_groups IPAddress`

Description
This tag specifies a list of multicast groups, with which your server should join to receive multicasted ICP queries.

Multicast is essentially the ability to send one IP packet to multiple receivers. Multicast is often used for audio and video conferencing systems. If you are unsure about multicast, please read the Multicast chapter in the Squid FAQ (http://squid.nlanr.net/Squid/FAQ/).

This option is to be set only if you want to RECEIVE multicast queries.

ICP replies are always sent via unicast, so this option does not affect whether or not you will receive replies from multicast group members.

Be sure you understand the difference between an ICP _query_ and an ICP _reply_.

Use cache_peer Directive for sending ICP queries.

Default `none`
`By default, Squid doesn't listen on any multicast groups`

Example
mcast_groups 239.128.16.128 224.0.1.20

Caution
Should not use a multicast address, which is already in use by another group of caches. We should not set this option to SEND multicast ICP.

Tag Name **tcp_outgoing_address**
Usage `tcp_outgoing_address IPAddress`

Description
It is used for connections made to remote servers. It is also used to communicate with other caches while using HTCP or CARP. Normally tcp_outgoing_address should not be specified. It is better to let the OS select a suitable address. There are some very specific network configurations where tcp_outgoing_address needs to be specified

Default `tcp_outgoing_address 255.255.255.255`

Example
The tcp_incoming_address can be specified using http_port.

Tag Name udp_incoming_address
Usage `udp_incoming_address IPAddress`

Description
It is used for the ICP socket receiving packets from other caches.

Default `udp_incoming_address 0.0.0.0`

Caution
Cannot have the same value, since they both (udp_incoming_address and udp_outgoing_address) use the port 3130.

Tag Name udp_outgoing_address
Usage `udp_outgoing_address IPAddress`

Description
It is used for the ICP packets sent out to the caches.

Default `udp_outgoing_address 255.255.255.255`

Caution
Cannot have the same value, since they both (udp_incoming_address and udp_outgoing_address) use the port 3130.

Peer cache servers and Squid hierarchy

Tag Name cache_peer
Usage `cache_peer hostname type http_port icp_port options`

Description
This tag is used to specify the other caches in the hierarchy. The cache_peer option is split into five fields. The first field is the hostname or IP of the cache that is to be queried. The second field indicates the type of relationship. The third field sets the HTTP port of the destination server, while the fourth sets the ICP (UDP) query port. The fifth field can contain zero or more keywords. Here are the detailed explanations on each field. See cache_peer_access also.

Hostname
Hostname (FQDN) or IP address of the cache to be queried should be mentioned

Hostname (FQDN) or IP address of the cache to be queried should be mentioned.

For ex,

cache_peer sib1.visolve.com sibling 3128 3130 [proxy-only]
cache_peer 172.16.1.100 sibling 3128 3130 [proxy-only]

Type
Here cache hierarchy should be specified. This option plays an important role in deciding neighbor selection.

- parent
- sibling
- multicast

Http_port
The port number where the cache listens for proxy requests. See also http_port

Icp_port
Used for querying neighbor caches about objects. To have a non-ICP neighbor specify '7' for the ICP port and make sure the neighbor machine has the UDP echo port enabled in its /etc/inetd.conf file. See also icp_port

OPTIONS:

proxy-only
To specify that objects fetched from this cache should not be saved locally.

Weight=n
To specify a weighted parent. The weight must be an integer. The default weight is 1, larger weights are favoured more.

ttl=n
To specify a IP multicast Time To Live (ttl) value when sending ICP queries to multicast groups. We do not accept ICP replies from random hosts. So you must configure other group members as peers with the multicast-responder option below.

no-query
This option is set for those peers, which do not support ICP queries. It is obvious to have doubt about the ICP port specified in, while using this option. Squid does not care what digit has been given in the ICP port when no-query is specified. Using any number is fine. It is recommended to use 0 to emphasis the fact that ICP is not used in any way (not even to UDP echo port 7).

This might be the typical example for this option :

cache_peer hostname sibling 8080 0 proxy-only no-query

By default, Port 3130 is typically where an ICP-aware proxy listens for ICP packets. Port 7 is the "echo" port (see /etc/services). It is typically handled by inetd as an internal process and simply "echoes" back what has been sent it. Since option "no-query" specified, port "7" is there so that if peer is

queried, Squid gets an answer and not declares peer as dead and therefore stop using it.

Port 7 is used when Squid has a non-ICP peer but still want to query it before sending requests there (no-query not specified). In such case, Squid will send the ICP queries to port 7 which is the UDP echo port.

default
If this is a parent cache which can be used as a "last-resort." and not ICP enabled then "default" would be the appropriate option. Simply adding default to a parent does not force all requests to be sent to that parent. The term default is perhaps a poor choice of words. If the cache is able to make direct connections, direct will be preferred over default. If needed to force all requests to parent cache(s), use the never_direct option.

round-robin
To define a set of parents which should be used in a round-robin fashion in the absence of any ICP queries.

multicast-responder
Indicates that the named peer is a member of a multicast group. ICP queries will not be sent directly to the peer, but ICP replies will be accepted from it.

closest-only
Indicates that, for ICP_OP_MISS replies, we'll only forward CLOSEST_PARENT_MISSes and never FIRST_PARENT_MISSes.

no-digest
To NOT request cache digests from this neighbor.

no-netdb-exchange
It disables requesting ICMP RTT database (NetDB) from the neighbor.

no-delay
To prevent access to this neighbor from influencing the delay pools.

login=user:password
If this is a personal/workgroup proxy and your parent requires proxy authentication.

connect-timeout=nn
To specify a peer specific connect timeout (also see the peer_connect_timeout directive).

digest-url=url
To tell Squid to fetch the cache digest (if digests are enabled) for this host from the specified URL rather than the Squid default location.

No cache peer is defined

Default `none`

Example
cache_peer proxy.visolve.com parent 3128 3130 default

cache_peer 172.16.1.100 sibling 3128 3130 proxy-only

cache_peer 172.16.1.123 sibling 3129 5500 weight=2

Caution
If you compiled Squid to support HTCP, your cache will automatically attempt to connect to TCP port 4827 (there is currently no option to change this port value). Cache digests are transferred via the HTTP port specified on the cache_peer line. Non-ICP neighbors must be specified as 'parent'.

Tag Name **cache_peer_domain**
Usage `cache_peer_domain cache_host domain [domain ...]`

Description
This tag is used to limit the domains for which the neighbor caches will be queried. It is used to communicate with different caches depending on the domain that the request is destined for

- Prefixing the domain name with '!' means that the cache will be queried for objects NOT in that domain.
- Any number of domains may be given for a cache-host, either on the same or separate lines.
- When multiple domains are given for a particular cache-host, the first matched domain is applied.
- Cache hosts with no domain restrictions are queried for all requests.
- There is also a 'cache_peer_access' tag in the ACL section .

Default `none`

Example
cache_peer_domain parent.foo.net .edu

It has the effect such that UDP query packets are sent to 'bigserver' only when the requested object exists on a server in the .edu domain.

Tag Name **neighbor_type_domain**
Usage `neighbor_type_domain parent|sibling domain domain ...`

Description
Modifying the neighbor type for specific domains is now possible. You can treat some domains differently than the default neighbor type specified on the 'cache_peer' line. Normally it should only be necessary to list domains, which should be treated differently because the default neighbor type applies for hostnames, which do not match domains listed here.

Default `none`

Example
cache_peer proxy.visolve.com parent 3128 3130
neighbor_type_domain proxy.visolve.com sibling .com .net

Tag Name **icp_query_timeout (msecs)**
Usage `icp_query_timeout milliseconds`

Description
Normally Squid will automatically determine an optimal ICP query timeout value based on the round-trip-time of recent ICP queries. If you want to override the value determined by Squid, set this 'icp_query_timeout' to a non-zero value .

Default `icp_query_timeout 0`

Example
This value is specified in MILLISECONDS, so, to use a 2-second timeout (the old default), you would write: icp_query_timeout 2000.

Tag Name **maximum_icp_query_timeout (msecs)**
Usage `maximum_icp_query_timeout milliseconds`

Description
Normally the ICP query timeout is determined dynamically. But sometimes it can lead to very large values (say 5 seconds). Use this option to put an upper limit on the dynamic timeout value. If 'icp_query_timeout' is set to zero, then this value is ignored.

Default `maximum_icp_query_timeout 2000`

Caution
Do NOT use this option to always use a fixed (instead of a dynamic) timeout value.

Tag Name **mcast_icp_query_timeout (msecs)**
Usage `mcast_icp_query_timeout milliseconds`

Description
For Multicast peers, Squid regularly sends out ICP "probes" to count how many other peers are listening on the given multicast address. This value specifies how long Squid should wait to count all the replies.

When Squid sends out a multicast query, it will wait at most mcast_icp_query_timeout seconds (it's perfectly possible that one day a peer will be on the moon: and it would probably be a bad idea to peer with that cache seriously, unless it was a parent for the Mars top-level domain.) It's unlikely that you will want to increase this value, but you may wish to drop it, so that only reasonably speedy replies are considered.

Default `mcast_icp_query_timeout 2000`

Caution
Do NOT use this option to always use a fixed (instead of a dynamic) timeout value.

Tag Name **dead_peer_timeout (secs)**
Usage `dead_peer_timeout seconds`

Description
This controls how long Squid waits to declare a peer cache as "dead." If there are no ICP replies received in this amount of time, Squid will declare the peer dead and not expect to receive any further ICP replies. However, it continues to send ICP queries, and will mark the peer as alive upon receipt of the first subsequent ICP reply .

Default `dead_peer_timeout 10 seconds`

Caution
This timeout also affects when Squid expects to receive ICP replies from peers. If more than 'dead_peer' seconds have passed since the last ICP reply was received, Squid will not expect to receive an ICP reply on the next query. Thus, if your time between requests is greater than this timeout, you will see a lot of requests sent DIRECT to origin servers instead of to your parents.

Tag Name **hierarchy_stoplist**
Usage `hierarchy_stoplist words`

Description
A list of words which, if found in a URL, cause the object to be handled directly by this cache. In other words, use this when to query neighbor caches for certain objects. This option can be listed multiple times. As some times this functionality is affected by the directive never_direct, See also never_direct.

Default `hierarchy_stoplist cgi-bin ?`

Squid will fetch URL's containing 'cgi-bin' or '?' from the origin servers directly without communicating with cache peers.

Example
hierarchy_stoplist jsp asp

If the URL contains the words jsp and asp, which indicate dynamic pages, then Squid will not query peers for the pages and will directly request the origin server.

Note
It is recommended to include all dynamic pages in this tag.

Tag Name **no_cache**
Usage `no_cache deny|allow aclname`

Description
A list of ACL elements, which, if matched, cause the reply to be immediately, removed from the cache. In other words, use this to force certain objects to never be cached.

Default
```
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
The word 'DENY' is to indicate the ACL names, which
should NOT be cached
```

Example
acl DENYPAGE urlpath_regex Servlet
no_cache deny DENYPAGE
The DENYPAGE acl assures that the url containing Servlet will NOT be cached.

Caution
It is recommended to use this directive effectively.

CACHE SIZE

Tag Name **cache_mem**
Usage `cache_mem bytes`

Description
'cache_mem' specifies the ideal amount of memory to be used for :

- In-Transit objects
- Hot Objects
- Negative-Cached objects

Data for these objects are stored in 4 KB blocks. This parameter specifies the ideal upper limit on the total size of 4 KB blocks allocated. In-transit objects have priority over the others. When additional space is needed for incoming data, negative-cached and hot objects will be released. In other words, the negative-cached and hot objects will fill up any unused space not needed for in-transit objects. If circumstances require, this limit will be exceeded. Specifically, if your incoming request rate requires more than 'cache_mem' of memory to hold in-transit objects, Squid will exceed this limit to satisfy the new requests. When the load decreases, blocks will be freed until the high-water mark is reached. Thereafter, blocks will be used to store hot objects.

Default `cache_mem 8 MB`

Example
cache_mem 1 GB

Caution
This parameter does not specify the maximum process size. It places a limit on one aspect of squid's memory usage. Squid uses memory for other things as well. Process will probably become twice or three times bigger than the value put here.

Tag Name **cache_swap_low (percent, 0-100)**
Usage `cache_swap_low percentage`

Description
This tag is used to specify the low-water mark for cache object replacement. Replacement begins when the swap (disk) usage is above the low-water mark and attempts to maintain utilization near the low-water mark. If utilization is close to the low-water mark, less replacement is done each time.

Default `cache_swap_low 90`

Example
cache_swap_low 95

Caution
If you have a large cache, 5% could be hundreds of MB. If this is the case, you may wish to set this number closer together.

Tag Name **cache_swap_high (percent, 0-100)**
Usage `cache_swap_high percentage`

Description
This tag is used to specify the high-water mark for cache object aggressive replacement. As swap utilization gets close to high-water mark, object eviction becomes more aggressive.

Default `cache_swap_high 95`

Example
cache_swap_high 98

Caution
If you have a large cache, 5% could be hundreds of MB. If this is the case, you may wish to set this number closer together.

Tag Name **maximum_object_size (bytes)**
Usage `maximum_object_size (bytes)`

Description
Objects larger than this size will NOT be saved on disk. The value is specified in kilobytes, and the default is 4 MB. If you wish to get a high BYTES hit ratio, you should probably increase this (one 32 MB object hit counts for 3200 10 KB hits). If you wish to increase speed more than your want to save bandwidth, you should leave this low. During the initial downloading of a file, the downloading time seems to be a normal one, but if you download the same file again, the time it takes for download will be very minimum. This indicates that, the file comes from the Cache.

Default `maximum_object_size 4096 KB`

Example
maximum_object_size 2000 KB

Caution
If using the LFUDA replacement policy you should increase this value to maximize the byte hit rate improvement of LFUDA. See replacement_policy for a discussion of this policy.

Tag Name **minimum_object_size (bytes)**
Usage `minimum_object_size (bytes)`

Description
Objects smaller than this size will NOT be saved on disk. The value is specified in kilobytes, and the default is 0 KB, which means there is no minimum. See maximum_object_size for more information.

Default `minimum_object_size 0 KB`

Example
minimum_object_size 2000 KB

Note
See maximum_object_size, which is reversibly applicable.

Tag Name **maximum_object_size_in_memory (bytes**
Usage `maximum_object_size_in_memory (bytes)`

Description
Objects greater than this size will not be attempted to be kept in the memory cache. This should be set high enough to keep objects accessed frequently in memory to improve performance while, low enough to keep larger objects from hoarding cache_mem.

Default `maximum_object_size_in_memory 8 KB`

Default maximum_object_size_in_memory 8 KB

Example
maximum_object_size_in_memory 20 KB

Tag Name **ipcache_size (number of entries)**
Usage ipcache_size (number of entries)

Description
This tag is used to specify the size of the ipcache.

Default ipcache_size 1024

Tag Name **ipcache_low (percent)**
Usage ipcache_low percentage

Description
This specifies the low water mark for caching IP addresses.

Default ipcache_low 90

Tag Name **ipcache_high (percent)**
Usage ipcache_high percentage

Description
This specifies the high water mark for caching IP addresses.

Default ipcache_high 95

Tag Name **fqdncache_size**
Usage fqdncache_size (number of entries)

Description
This specifies the maximum number of FQDN cache entries.

Default fqdncache_size 1024

Tag Name **cache_replacement_policy**
Usage `cache_replacement_policy policy`

Description
The cache replacement policy parameter decides which objects will remain in cache and which objects are evicted (replaced) to create space for the new objects.

- LRU : Squid's original list based LRU policy
- heap GDSF : Greedy-Dual Size Frequency
- heap LFUDA : Least Frequently Used with Dynamic Aging
- heap LRU : LRU policy implemented using a heap

This applies to any cache_dir lines listed below this.
The LRU policies keep recently referenced objects. i.e., it replaces the object that has not been accessed for the longest time.

The heap GDSF policy optimizes object-hit rate by keeping smaller popular objects in cache. So it has a better chance of getting a hit. It achieves a lower byte hit rate than LFUDA though, since it evicts larger (possibly popular) objects.

The heap LFUDA (Least Frequently Used with Dynamic Aging) policy keeps popular objects in cache regardless of their size and thus optimizes byte hit rate at the expense of hit rate since one large, popular object will prevent many smaller, slightly less popular objects from being cached.

Both policies utilize a dynamic aging mechanism that prevents cache pollution that can otherwise occur with frequency-based replacement policies.

For more information about the GDSF and LFUDA cache replacement policies see http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html .

Default `cache_replacement_policy lru`

Example
cache_replacement_policy heap LFUDA
To use this policy, Squid should be built with configure option --enable-removal-policy=heap or simply --enable-removal-policy.

Caution
If using the LFUDA replacement policy, the value of maximum_object_size should be increased above its default of 4096 KB to maximize the potential byte hit rate improvement of LFUDA. If needed to use other policies than default, squid should be built with configure option --enable-removal-policies.

Tag Name **memory_replacement_policy**
Usage `memory_replacement_policy policy`

Description
The memory replacement policy parameter determines which objects are purged from memory, when memory space is needed.

See cache_replacement_policy for details

Default `memory_replacement_policy lru`

LOG FILE PATH NAMES AND CACHE DIRECTORIES

Tag Name **cache_dir**
Usage
```
cache_dir Type Maxobjsize Directory-Name Mbytes Level-1
Level2 [..]
DISKD :
cache_dir diskd Maxobjsize Directory-Name MB L1 L2 Q1 Q2
```

Description

Type specifies the kind of storage system to use. Most everyone will want to use "ufs" as the type. If you are using Async I/O (--enable async-io) on Linux or Solaris, then you may want to try "aufs" as the type. Async IO support may be buggy, however, so beware.

Maxobjsize refers to the max object size this storedir supports. It is used to initially choose the storedir to dump the object. -1 means 'any size'.

'Directory' is a top-level directory where cache swap files will be stored. If you want to use an entire disk for caching, then this can be the mount-point directory. The directory must exist and be writable by the Squid process. Squid will NOT create any directory.

'Mbytes' is the amount of disk space (MB) to use under this directory.

'Level-1' is the number of first-level subdirectories, which will be created under the 'Directory'.

'Level-2' is the number of second-level subdirectories, which will be created under each first-level directory. To create swap directory use /usr/local/squid/bin/squid –z option.

For the diskd type, Q1 specifies the number of unacknowledged I/O requests when

For the diskd type, Q1 specifies the number of unacknowledged I/O requests when Squid stops opening new files. If this many messages are in the queues, Squid won't open new files. Q2 specifies the number of unacknowledged messages when Squid starts blocking. If this many messages are in the queues, Squid blocks until it receives some replies.

Default `cache_dir ufs /usr/local/squid/cache 100 16 256`

Example
cache_dir ufs /cache1 5000 16 256
cache_dir ufs /cache2 7000 16 256

Note
Can specify multiple cache_dir lines to spread the cache among different disk partitions. Click Here to find more informations on file systems and cache_dir.

Tag Name **cache_access_log**
Usage `cache_access_log Directory-path/filename`

Description
This tag is used to specify the path of the access.log file, which logs the client request activity. It contains an entry for every HTTP and ICP queries received. Log Details can be customized using log_mime_hdrs, log_fqdn, client_netmask and emulate_httpd_log.

See for Detailed information about this log file. See also log_icp_queries.

Default `cache_access_log /usr/local/squid/logs/access.log`

Example
cache_access_log /var/log/squid_access.log

Caution
It is safe to set log_mime_hdrs off.

Tag Name **cache_log**
Usage `cache_log Directory-path/filename`

Description
This tag is used to set the path of the Cache logging file. This is where general information about the cache's behaviour goes. Amount of data logged to this file can be increased with the debug_options tag below.

Default `cache_log /usr/local/squid/logs/cache.log`

Example
cache_log /var/log/squid_cache.log

Caution
Do not change the default value of debug_options unless otherwise needed. Because if

debug_options value is high, then logging information goes high. This leads to undesirable growth in log file.

Tag Name **cache_store_log**
Usage `cache_store_log Directory-path/filename`

Description
This tag is used to specify the location of the store.log, the file that logs the activities of the storage manager. The file shows which objects are ejected from the cache, and which objects are saved and for how long.

Default `cache_store_log /usr/local/squid/logs/store.log`

Example
cache_store_log /var/log/squid_store.log

Caution
There are no real utilities to analyze this data. So it is recommended to disable this tag

Note
To disable, enter "none" instead of the filename.

Tag Name **cache_swap_log**
Usage `cache_ swap _log Directory-path/filename`

Description
This tag specifies the location for the cache "swap.log." This log file holds the metadata of objects saved on disk. It is used to rebuild the cache during startup. Normally this file resides in the first 'cache_dir' directory, but you may specify an alternate pathname here. Note, you must give a full filename, not just a directory. Since this is the index for the whole object list you CANNOT periodically rotate it.

If you have more than one 'cache_dir', these swap logs will have names such as:

- cache_swap_log.00
- cache_swap_log.01
- cache_swap_log.02

The numbered extension (which is added automatically) corresponds to the order of the 'cache_dir' lines in this configuration file.

Default `cache_ swap _log /usr/local/squid/logs/ swap.log`

Example
cache_ swap _log /var/log/squid_ swap.log

Caution
If you change the order of the 'cache_dir' lines in this file, then these log files will NOT correspond to the correct 'cache_dir' entry (unless you manually rename them). We

correspond to the correct 'cache_dir' entry (unless you manually rename them). We recommend that you do NOT use this option. It is better to keep these log files in each 'cache_dir' directory.

Tag Name **emulate_httpd_log**
Usage `emulate_httpd_log on|off`

Description
The Cache can emulate the log file format, which many 'httpd' programs use. To disable/enable this emulation, set emulate_httpd_log to 'off' or 'on'.

Default `emulate_httpd_log off`
(By Default Squid Native Log format is used. Since it includes useful information that Squid-specific log analyzers use).

Example
emulate_httpd_log on

Caution
Before setting this to ON, make sure you have httpd_log file analyzers which will analyze log files and give us useful information.

Tag Name **log_ip_on_direct**
Usage `log_ip_on_direct on|off`

Description
This tag is used to enable/disable logging of the destination IP address in the hierarchy log tag when the cache directs the request to the origin server.

Default `log_ip_on_direct on`

Example
log_ip_on_direct off

Note
Earlier Squid versions logged the hostname here. If you prefer the old way, set this to off.

Tag Name **mime_table**
Usage `mime_table Directory-Path/filename`

Description
This tag is used to set the pathname to Squid's MIME table. This file contains Squid's supported mime types.

Default `mime_table /usr/local/squid/etc/mime.conf`

Caution
Shouldn't need to change this, but the default file contains examples and formatting information if done.

Tag Name **log_mime_hdrs**
Usage `log_mime_hdrs on|off`

Description
The Cache can record both the request and the response MIME headers for each HTTP transaction. The headers are encoded safely and will appear as two bracketed fields at the end of the access log (for either the native or httpd-emulated log formats). To enable this logging, set log_mime_hdrs to 'on'.

Default `log_mime_hdrs off`

Tag Name **useragent_log**
Usage `useragent_log Directory-path/filename`

Description
If configured with the "--enable-useragent_log" configure option, Squid will write the User-Agent field from HTTP requests to the filename specified here.

Default `useragent_log none (By default useragent_log is disabled.)`

Example
useragent_log /var/log/useragent.log

Tag Name **referer_log**
Usage `referer_log Directory-path/filename`

Description
If configured with the "--enable-referer_log" configure option, Squid will write the Referer field from HTTP requests to the filename specified here.

Default `referer_log none (By default referer_log is disabled.)`

Example
referer_log /var/log/referer.log

Tag Name **pid_filename**
Usage `pid_filename Directory-path/filename`

Description
This tag specifies the location of the file in which Squid writes its process-ids.

Default `pid_filename /usr/local/squid/logs/squid.pid`

Example
pid_filename /var/lock/squid.pid

Caution
To disable, enter "none".

Tag Name **debug_options**
Usage `debug_options section, level`

Description
Logging options are set as section, level, where each source file is assigned a unique section. Lower levels result in less output. The magic word "ALL" sets debugging levels for all sections. We recommend normally running with "ALL, 1".

Default `debug_options ALL, 1`

Example
debug_options ALL, 9

Caution
Full debugging (level 9) can result in a very large log file, so be careful. Normally, running with "ALL, 1" is recommended.

Tag Name **log_fqdn**
Usage `log_fqdn on|off`

Description
This tag can be set to ON, if you wish to log fully qualified domain names in the access.log.

Default `log_fqdn off`

Example
log_fqdn on

Caution
To do this, Squid does a DNS lookup of all IP's connecting to it. This can (in some situations) increase latency, which makes your cache seem slower for interactive browsing.

Tag Name client_netmask
Usage `client_netmask NETMASK`

Description
A netmask for client addresses in log files and cachemgr output. Change this to protect the privacy of your cache clients. A netmaskof 255.255.255.0 will log all IP's in that range with the last digit set to '0'.

Default `client_netmask 255.255.255.255`

Example
client_netmask 255.255.255.0

Caution
When you enable this tag, then the client's visit pages cannot be identified.

Support for external functions

Tag Name ftp_user
Usage `ftp_user username`

Description
This tag can be used if you want the anonymous login password to be more informative. You can set this to something reasonable for your domain, like squid@squid.visolve.com. The reason why this is domain less by default is that the request can be made on the behalf of a user in any domain, depending on how the cache is used. Some ftp servers also validate the email address. For detailed explanation Click Here

Default `ftp_user Squid@`

Example
ftp_user squid@squid.visolve.com

Tag Name ftp_list_width
Usage `ftp_list_width number`

Description
This tag is used to set the width of ftp listings. This should be set to fit in the width of a standard browser. Setting this too small can cut off long filenames when browsing ftp

sites.

Default `ftp_list_width 32`

Example
ftp_list_width 64

Tag Name **ftp_passive**
Usage `ftp_passive on|off`

Description
If your firewall does not allow Squid to use passive connections, then turn off this option.

Default `ftp_passive on`

Tag Name **cache_dns_program**
Usage `cache_dns_program program`

Description
This tag is used to specify the location of the executable for dns lookup process. This option is only available if Squid is rebuilt with the --disable-internal-dns option.

The external dns program uses the normal resolver libraries which is a much more mature DNS client. The internal DNS client still has some problems with special cases in the DNS protocol. However, things has gotten a lot better compared to the early version so any of these issues are not likely to be noticed, and is heavily out weighted by the improved performance and reliability. But drawbacks of the external DNS helper are likely to be noticed when using external DNS. If DNS lookups are slow then the external DNS helper will hit the roof and no further DNS lookups can complete (some Squid versions even abort in such case).

Recommendation: Use the internal DNS client unless an experience problem which forces to use the external one until a fix is provided.

Default `cache_dns_program /usr/local/squid/libexec/squid/`

Example
cache_dns_program /usr/local/squid/bin/dnsserver

Tag Name **dns_children**
Usage `dns_children number (1 to 32)`

Description
The number of processes spawn to service DNS name lookups are specified here. For heavily loaded caches on large servers, There is probably need to increase this value to

at least 10. The maximum is 32. The default is 5. This option is only available if Squid is rebuilt with the --disable-internal-dns option. The number of processes increases, the performance of DNS lookups also increases. It is recommended to use maximum child processes (32).

The limitation that the external dnsserver helper can only handle one DNS lookup at a time and cannot be aborted prior to the 2 minutes DNS lookup time-out. The internal DNS client does not have this limitation and can handle any number of concurrent lookups. See the description of cache_dns_program.

Default `dns_children 5`

Example
dns_children 10

Caution
You must have at least one dnsserver process

Tag Name **dns_retransmit_interval**
Usage `dns_retransmit_interval time-units`

Description
This tag is used to set the initial retransmit interval for DNS queries. The interval is doubled each time all configured DNS servers have been tried

Default `dns_retransmit_interval 5 seconds`

Tag Name **dns_timeout**
Usage `dns_timeout time-units`

Description
This tag is used to set the DNS Query time-out. If no response is received to a DNS query within this time then all DNS servers for the queried domain is assumed to be unavailable

Default `dns_timeout 5 minutes`

Tag Name **dns_defnames**
Usage `dns_defnames on|off`

Description
Normally the 'dnsserver' disables the RES_DEFNAMES resolver option (see res_init(3)). This prevents caches in a hierarchy from interpreting single component hostnames locally. To allow dnsserver handle single component names, enable this option. This option is only available if Squid is rebuilt with the --disable-internal-dnsoption.

option is only available if Squid is built with the --disable-internal-dnscode.

Default dns_defnames off

Tag Name **dns_nameservers**
Usage dns_nameservers IPaddress

Description
This tag can be used if you want to specify a list of DNS name servers (IP addresses) to use instead of those given in your /etc/resolv.conf file

Default dns_nameservers none

Example
dns_nameservers 172.16.1.102 204.54.6.20

Tag Name **diskd_program**
Usage diskd_program path/to/diskdfile

Description
To specify the location of the diskd executable. Note that this is only useful if you have compiled in diskd as one of the store io modules.

Default diskd_program /usr/local/squid/libexec/squid/diskd

Example
diskd_program /local/squid/bin/diskd

Tag Name **unlinkd_program**
Usage unlinkd_program path/to/unlinkedfile

Description
This tag specifies the location of the unlinkd program. This isn't needed if you are using async-io since it's handled by a thread

Default unlinkd_program /usr/local/squid/libexec/squid/unlinkd

Example
unlinkd_program /usr/local/squid/bin/unlinkd

Tag Name **pinger_program**
Usage pinger_program path/to/pingerfile

Description

This tag is used to specify the location of the executable for the pinger process. This is only useful if you configured Squid (during compilation) with the '--enable-icmp' option

Default `pinger_program /usr/local/squid/libexec/squid/`

Example
pinger_program /usr/local/squid/bin/pinger

Tag Name **redirect_program**
Usage `redirect_program path/to/redirector`

Description
This tag is used to specify the location of the executable for the URL redirector. Since they can perform almost any function there isn't one included. Click here for information on how to write one. By default, a redirector is not used

Default `redirect_program none`

Example
redirect_program /usr/local/squirm/bin/squirm

Tag Name **redirect_children**
Usage `redirect_children number`

Description
This tag is used to set the number of redirect processes to spawn

Default `redirect_children 5`

Example
redirect_children 10

Caution
If you start too few Squid will have to wait for them to process a back log of URLs, slowing it down. If you start too many they will use RAM and other system resources.

Tag Name **redirect_rewrites_host_header**
Usage `redirect_rewrites_host_header on|off`

Description
By default Squid rewrites any Host: header in redirected requests. If you are running a accelerator then this may not be a wanted effect of a redirector

Default `redirect_rewrites_host_header on`

Tag Name **redirect_access**
Usage `redirector_access allow|deny`

Description
If defined, this access list specifies which requests are sent to the redirector processes

Default `All requests are sent`

Example
redirector_access allow aclname

Tag Name **authenticate_program**
Usage `authenticate_program path/to/program path/to/passwdfile`

Description
This tag is used to specify the command for the external authenticator. Such a program reads a line containing "username password" and replies "OK" or "ERR" in an endless loop. If you use an authenticator, make sure you have 1 acl of type proxy_auth. If you want to use the traditional proxy authentication, jump over to the ../auth_modules/NCSA directory and give

```
# make
# make install
```

The source for this program is included in the source distribution, in the auth_modules/NCSA directory. You should now have an ncsa_authprogram in the same directory where your squid binary lives. You may need to create a password file. If you have been using proxy authentication before, you probably already have such a file. You can get apache's htpasswd program from here. Pick a pathname for your password file. We will assume you will want to put it in the same directory as your Squid.conf.

Default `authenticate_program none By default, the authenticator_program is not used`

Example
authenticate_program /usr/local/squid/bin/ncsa_auth /usr/local/squid/etc/passwd

Tag Name **authenticate_children**
Usage `authenticate_children number`

Description
The number of authenticator processes to spawn (default 5).

Default `authenticate_children 5`

Caution
If you start too few Squid will have to wait for them to process a back log of

If you start too few Squid will have to wait for them to process a back log of usercode/password verifications, slowing it down. When password verifications are done via a (slow) network you are likely to need lots of authenticator processes.

Tag Name **authenticate_ttl**
Usage `authenticate_ttl seconds`

Description
This tag is used to specify the time a checked username/password combination remains cached (default 3600). If a wrong password is given for a cached user, the user gets removed from the username/password cache forcing a revalidation.

Default `authenticate_ttl 3600`

Tag Name **authenticate_ip_ttl**
Usage `authenticate_ip_ttl number`

Description
With this option you control how long a proxy authentication will be bound to a specific IP address. If a request using the same user name is received during this time then access will be denied and both users are required to reauthenticate them selves. The idea behind this is to make it annoying for people to share their password to their friends, but yet allow a dialup user to reconnect on a different dialup port. The default is 0 to disable the check. Recommended values if you have dialup users are no more than 60 (seconds). If all your users are stationary then higher values may be used.

Default `authenticate_ip_ttl 0`

Example
`authenticate_ip_ttl 3600`

Tag Name **authenticate_ip_ttl_is_strict**
Usage `authenticate_ip_ttl_is_strict on|off`

Description
This option makes authenticate_ip_ttl a bit stricter. With this enabled authenticate_ip_ttl will deny all access from other IPaddresses until the TTL has expired, and the IP address "owning" the userid will not be forced to reauthenticate.

Default `authenticate_ip_ttl_is_strict on`

TUNING THE SQUID CACHE

Tag Name **wais_relay_host**
wais_relay_port
Usage wais_relay_host
wais_relay_port

Description
Relay WAIS request to host (1st arg) at port (2 arg).

Default wais_relay_port 0

Example
wais_relay_host localhost
wais_relay_port 8000

Tag Name **request_header_max_size**
Usage request_header_max_size (KB)

Description
This specifies the maximum size for HTTP headers in a request. Request headers are usually relatively small (about 512 bytes). Placing a limit on the request header size will catch certain bugs (for example with persistent connections) and possibly buffer-overflow or denial-of-service attacks.

Default request_header_max_size 10 KB

Tag Name **request_body_max_size**
Usage request_body_max_size (KB)

Description
This specifies the maximum size for an HTTP request body. In other words, the maximum size of a PUT/POST request. A user, who attempts to send a request with a body larger than this limit receives an "Invalid Request" error message. If you set this parameter to a zero, there will be no limit imposed.

Default request_body_max_size 1 MB

Tag Name **reply_body_max_size**
Usage `reply_body_max_size (KB)`

Description
This option specifies the maximum size of a reply body. It can be used to prevent users from downloading very large files, such as MP3's and movies. The reply size is checked twice. First when we get the reply headers, we check the content-length value. If the content length value exists and is larger than this parameter, the request is denied and the user receives an error message that says "the request or reply is too large." If there is no content-length, and the reply size exceeds this limit, the client's connection is just closed and they will receive a partial reply.

Default `reply_body_max_size 0`
If this parameter is set to zero (the default), there will be no limit imposed.

Caution
Downstream caches probably cannot detect a partial reply if there is no content-length header, so they will cache partial responses and give them out as hits. You should NOT use this option, if you have downstream caches.

Tag Name **refresh_pattern**
Usage `refresh_pattern [-i] regex min percent max [options]`

Description
'Min' is the time (in minutes) an object without an explicit expiry time should be considered fresh. The recommended value is 0; any higher values may cause dynamic applications to be erraneously cached unless the application designer has taken the appropriate actions.

'Percent' is a percentage of the objects age (time since last modification age) an object without explicit expiry time will be considered fresh.

'Max' is an upper limit on how long objects without an explicit expiry time will be considered fresh.

Options:

override-expire
override-lastmod
reload-into-ims
ignore-reload

override-expire enforces min age even if the server sent a Expires: header. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems, which it causes.

override-lastmod enforces min age even on objects that was modified recently.

reload-into-ims changes client no-cache or ``reload'' to If-Modified-Since requests.

Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems, which it causes.

ignore-reload ignores a client no-cache or ``reload'' header. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems, which it causes.

Basically a cached object is: (the order is changed from 1.1.X)

FRESH if expiresbr>STALE if age >max
FRESH if lm-factorbr>STALE FRESH if age
else STALE

The refresh_pattern lines are checked in the order listed here. The first entry which matches is used. If none of the entries match, then the default will be used.

Default
```
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern . 0 20% 4320
```

Tag Name **reference_age**
Usage `reference_age time-units`

Description
As a part of normal operation, Squid performs Least Recently Used removal of cached objects. The LRU age for removal is computed dynamically, based on the amount of disk space in use. The dynamic value can be seen in the Cache Manager 'info' output.

The 'reference_age' parameter defines the maximum LRU age

Default `reference_age 1 year`

Example
For example, setting reference_age to '1 week' will cause objects to be removed, if they have not been accessed for a week or more. The default value is one year.

Specify a number here, followed by units of time. For example:

1 week
3.5 days
4 months
2.2 hours

Caution
This parameter is not used when using the enhanced replacement policies, GDSH or LFUDA.

Tag Name **quick_abort_min**
quick_abort_max
quick_abort_pct

Usage
```
quick_abort_min (KB)
quick_abort_max (KB)
quick_abort_pct (percent)
```

Description
The cache can be configured to continue downloading aborted requests. This may be undesirable on slow (e.g. SLIP) links and/or very busy caches. Impatient users may tie up file descriptors and bandwidth by repeatedly requesting and immediately aborting downloads.

When the user aborts a request, Squid will check the quick_abort values to the amount of data transferred until then.

If the transfer has less than 'quick_abort_min' KB remaining, it will finish the retrieval. Setting 'quick_abort_min' to –1 will disable the quick_abort feature.

If the transfer has more than 'quick_abort_max' KB remaining, it will abort the retrieval.

If more than 'quick_abort_pct' of the transfer has completed, it will finish the retrieval.

Default
```
quick_abort_min 16 KB
quick_abort_max 16 KB
quick_abort_pct 95
```

Caution
This may be undesirable on slow (e.g. SLIP) links and/or very busy caches.

Tag Name **negative_ttl**
Usage
```
negative_ttl time-units
```

Description
Time-to-Live (TTL) for failed requests. Certain types of failures (such as "connection refused" and "404 Not Found") are negatively-cached for a configurable amount of time. Note that, this is different from negative caching of DNS lookups.

Default
```
The default is 5 minutes.

negative_ttl 5 minutes
```

Tag Name **positive_dns_ttl**
Usage `positive_dns_ttl time-units`

Description
Time-to-Live (TTL) for positive caching of successful DNS lookups. If you want to minimize the use of Squid's ipcache, set this to 1, not 0.

Default `Default is 6 hours (360 minutes).`

```
positive_dns_ttl 6 hours
```

Tag Name **negative_dns_ttl**
Usage `negative_dns_ttl time-units`

Description
Time-to-Live (TTL) for negative caching of failed DNS lookups

Default `negative_dns_ttl 5 minutes`

Tag Name **range_offset_limit**
Usage `range_offset_limit (bytes)`

Description
Sets an upper limit on how far into the file a Range request may be to cause Squid to prefetch the whole file. If beyond this limit, then Squid forwards the Range request as it is and the result is NOT cached.

This is to stop a far ahead range request (lets say start at 17MB) from making Squid fetch the whole object up to that point before sending anything to the client.

A value of -1 causes Squid to always fetch the object from the beginning so that it may cache the result. (2.0 style).

A value of 0 causes Squid to never fetch more than the client requested. (default) .

Default `range_offset_limit 0 KB`

TIMEOUTS

Tag Name connect_timeout
Usage `connect_timeout seconds`

Description
The time duration until which squid waits for the reply from the origin server. If it exceeds this squid will respond with the error message "Connection timed out" to the client

Default `connect_timeout 120 seconds`

Example
connect_timeout 180 seconds

Caution
Increasing the time here will lead to annoying of browser user.

Tag Name peer_connect_timeout
Usage `peer_connect_timeout time-units`

Description
This parameter specifies how long to wait for a pending TCP connection to a peer cache. The default is 30 seconds. You may also set different timeout values for individual neighbors with the 'connect-timeout' option on a 'cache_peer' line

Default `peer_connect_timeout 30 seconds`

Example
peer_connect_timeout 45 seconds

Caution
Setting of peer_connect_timeout to more than 30 seconds will be a performance issue.

Tag Name siteselect_timeout
Usage `siteselect_timeout time-units`

Description
Site select timeout is the timeout for URN to the multiple URLs selection. URN is a protocol designed for location-independent name resolution, specified in RFC 2169. This option configures the siteselect_timeout directive and defaults to 4 seconds. You do not need to change this.

Default `siteselect_timeout 4 seconds`

Example
siteselect_timeout 6 seconds

Tag Name **read_timeout**
Usage `read_timeout time-units`

Description
The read_timeout is applied on server-side connections. After each successful read(), the timeout will be extended by this amount. If no data is read again after this amount of time, the request is aborted and logged with ERR_READ_TIMEOUT. The default is 15 minutes.

Default `read_timeout 15 minutes`

Example
read_timeout 10 minutes

Tag Name **request_timeout**
Usage `request_timeout seconds`

Description
This tag specifies Squid the time in seconds to wait for an HTTP request after connection establishment. For persistent connections, Squid will wait this long after the previous request completes

Default `request_timeout 30 seconds`

Example
request_timeout 20 seconds

Tag Name **client_lifetime**
Usage `client_lifetime time-units`

Description
The maximum amount of time that a client (browser) is allowed to remain connected to the cache process. This protects the Cache from having a lot of sockets (and hence file descriptors) tied up in a CLOSE_WAIT state from remote clients that go away without properly shutting down (either because of a network failure or because of a poor client implementation). The default is one day, 1440 minutes

Default `client_lifetime 1 day`

Example

client_lifetime 1000 minutes

Caution
The default value is intended to be much larger than any client would ever need to be connected to your cache. You should probably change client_lifetime only as a last resort. If you seem to have many client connections tying up filedescriptors, we recommend first tuning the read_timeout, request_timeout, pconn_timeout and quick_abort values. If the more file descriptors are in use then the memory in use will also increase, which is also a performance issue.

Tag Name **half_closed_clients**
Usage `half_closed_clients on|off`

Description
Some clients may shutdown the sending side of their TCP connections, while leaving their receiving sides open. Sometimes, Squid cannot tell the difference between a half-closed and a fully-closed TCP connection. By default, half-closed client connections are kept open until a read(2) or write(2) on the socket returns an error. Change this option to 'off' and Squid will immediately close client connections when read (2) returns "no more data to read

Default `half_closed_clients on`

Example
half_closed_clients off

Tag Name **pconn_timeout**
Usage `pconn_timeout seconds`

Description
Persistent timeout is the timeout value for persistent connections. Squid closes persistent connections if they are idle for this amount of time. Persistent connections are disabled entirely if this option is set to a value less than 10 seconds. The default is 120 seconds and likely does not need to be changed.

Default `pconn_timeout 120 seconds`

Example
pconn_timeout 60 seconds

Tag Name **ident_timeout**
Usage `ident_timeout seconds`

Description
Maximum time to wait for IDENT requests. If this is too high, and you enabled 'ident_lookup', then you might be susceptible to denial-of-service by having many ident requests going at once. Only src type ACL checks are fully supported. An

src_domainACL might work at times, but it will not always provide the correct result.
This option may be disabled by using --disable-ident with the configure script

Default `ident_timeout 10 seconds`

Example
ident_timeout 5 seconds

Tag Name **shutdown_lifetime**
Usage `shutdown_lifetime time-units`

Description
When SIGTERM or SIGHUP is received, the cache is put into "shutdown pending" mode until all active sockets are closed. This value is the lifetime to set for all open descriptors during shutdown mode. Any active clients after this many seconds will receive a 'timeout' message

Default `shutdown_lifetime 30 seconds`

Example
shutdown_lifetime 45 seconds

Caution
If this time is set to be too low then some file descriptors may remain open which will be a performance issue in memory usage.

ACCESS CONTROLS

Tag Name **acl**
Usage `acl aclname acltype string1 ... | "file"`

Description
This tag is used for defining an access List. When using "file" the file should contain one item per line By default, regular expressions are CASE-SENSITIVE. To make them case-insensitive, use the -i option.

Acl Type: **src**

Description
This will look client IP Address.

Usage `acl aclname src ip-address/netmask.`

Example
1.This refers to the whole Network with address 172.16.1.0 - acl aclname src 172.16.1.0/24
2.This refers specific single IP Address - acl aclname src 172.16.1.25/32
3.This refers range of IP Addresses from 172.16.1.25-172.16.1.35 - acl aclname src 172.16.1.25-172.16.1.35/32

Note
While giving Netmask caution must be exerted in what value is given

Acl Type: `dst`

Description
This is same as src with only difference refers Server IPaddress. First Squid will dns-lookup for IPAddress from the domain-name, which is in request header. Then this acl is interpreted.

Usage `acl aclname dst ip-address/netmask.`

Acl Type: `srcdomain`

Description
Since squid needs to reverse dns lookup (from client ip-address to client domain-name) before this acl is interpreted, it can cause processing delays. This lookup adds some delay to the request.

Usage `acl aclname srcdomain domain-name`

Example
acl aclname srcdomain .kovaiteam.com

Note
Here "." is more important.

Acl Type: `dstdomain`

Description
This is the effective method to control specific domain

Usage `acl aclname dstdomain domain-name`

Example
acl aclname dstdomain .kovaiteam.com
Hence this looks for *.kovaiteam.com from URL
Hence this looks for *.kovaiteam.com from URL

Note
Here "." is more important.

Acl Type: **srcdom_regex**

Description
Since squid needs to reverse dns lookup (from client ip-address to client domain-name) before this acl is interpreted, it can cause processing delays. This lookup adds some delay to the request

Usage `acl aclname srcdom_regex pattern`

Example
acl aclname srcdom_regex kovai
Hence this looks for the word kovai from the client domain name

Note
Better avoid using this acl type to be away from latency.

Acl Type: **dstdom_regex**

Description
This is also an effective method as dstdomain

Usage `acl aclname dstdom_regex pattern`

Example
acl aclname dstdom_regex kovai
Hence this looks for the word kovai from the client domain name

Acl Type: **time**

Description
Time of day, and day of week

Usage
```
acl aclname time [day-abbreviations] [h1:m1-h2:m2]
day-abbreviations:
```

```
S - Sunday
M - Monday
T - Tuesday
W - Wednesday
H - Thursday
F - Friday
A - Saturday
h1:m1 must be less than h2:m2
```

Example
acl ACLTIME time M 9:00-17:00
ACLTIME refers day of Monday from 9:00 to 17:00.

Acl Type: `url_regex`

Description
The url_regex means to search the entire URL for the regular expression you specify. Note that these regular expressions are case-sensitive. To make them case-insensitive, use the -i option.

Usage `acl aclname url_regex pattern`

Example
acl ACLREG url_regex cooking
ACLREG refers to the url containing "cooking" not "Cooking"

Acl Type: `urpath_regex`

Description
The urpath_regex regular expression pattern matching from URL but without protocol and hostname. Note that these regular expressions are case-sensitive

Usage `acl aclname urlpath_regex pattern`

Example
`acl ACLPATHREG urlpath_regex cooking`
ACLPATHREG refers only containing "cooking'' not "Cooking"; and without referring protocol and hostname.
If URL is http://www.visolve.com/folder/subdir/cooking/first.html then this acltype only looks after http://www.visolve.com/ .

Acl Type: `port`

Description
Access can be controlled by destination (server) port address

Usage `acl aclname port port-no`

Usage acl aclname port port ..

Example
This example allows http_access only to the destination 172.16.1.115:80 from network 172.16.1.0

```
acl acceleratedhost dst 172.16.1.115/255.255.255.255
acl acceleratedport port 80
acl mynet src 172.16.1.0/255.255.255.0
http_access allow acceleratedhost acceleratedport mynet
http_access deny all
```

Acl Type: `proto`

Description
This specifies the transfer protocol

Usage `acl aclname proto protocol`

Example
acl aclname proto HTTP FTP
This refers protocols HTTP and FTP

Acl Type: `method`

Description
This specifies the type of the method of the request

Usage `acl aclname method method-type`

Example
acl aclname method GET POST
This refers get and post methods only

Acl Type: `browser`

Description
Regular expression pattern matching on the request's user-agent header

Usage `acl aclname browser pattern`

Example
acl aclname browser MOZILLA
This refers to the requests, which are coming from the browsers who have "MOZILLA" keyword in the user-agent header.

Acl Type: `ident`

Description
String matching on the user's name

Usage `acl aclname ident username ...`

Example
You can use ident to allow specific users access to your cache. This requires that an ident server process runs on the user's machine(s). In your squid.conf configuration file you would write something like this:

ident_lookup on
acl friends ident kim lisa frank joe
http_access allow friends
http_access deny all

Acl Type: `ident_regex`

Description
Regular expression pattern matching on the user's name. String match on ident output. Use REQUIRED to accept any non-null ident

Usage `acl aclname ident_regex pattern`

Example
You can use ident to allow specific users access to your cache. This requires that an ident server process run on the user's machine(s). In your squid.conf configuration file you would write something like this:

ident_lookup on
acl friends ident_regex joe
This looks for the pattern "joe" in username

Acl Type: `src_as`

Description
source (client) Autonomous System number

Acl Type: `dst_as`

Description
destination (server) Autonomous System number

Acl Type: **proxy_auth**

Description
User authentication via external processes. proxy_auth requires an EXTERNAL authentication program to check username/password combinations (see authenticate_program).

Usage `acl aclname proxy_auth username...`

use REQUIRED instead of username to accept any valid username

Example
acl ACLAUTH proxy_auth usha venkatesh balu deepa

This acl is for authenticating users usha, venkatesh, balu and deepa by external programs.

Warning
proxy_auth can't be used in a transparent proxy. It collides with any authentication done by origin servers. It may seem like it works at first, but it doesn't. When a Proxy-Authentication header is sent but it is not needed during ACL checking the username is NOT logged in access.log.

Acl Type: **proxy_auth_regex**

Description
This is same as proxy_auth with a difference. That is it matches the pattern with usernames, which are given in authenticate_program

Usage `acl aclname proxy_auth_regex [-i] pattern...`

Acl Type: **snmp_community**

Description
SNMP community string matching

Example
acl aclname snmp_community public
snmp_access aclname

Acl Type: **maxconn**

Description

Description
A limit on the maximum number of connections from a single client IP address. It is an ACL that will be true if the user has more than maxconn connections open. It is used in http_access to allow/deny the request just like all the other acl types.

Example
acl someuser src 1.2.3.4
acl twoconn maxconn 5
http_access deny someuser twoconn
http_access allow !twoconn

Note
maxconn acl requires client_db feature, so if you disabled that (client_db off) maxconn won't work.

Acl Type: **req_mime_type**
Usage `acl aclname req_mime_type pattern`

Description
Regular expression pattern matching on the request content-type header

Example
acl aclname req_mime_type text

This acl looks for the pattern "text" in request mime header

Acl Type: **arp**
Usage `acl aclname arp ARP-ADDRESS`

Description
Ethernet (MAC) address matching This acl is supported on Linux, Solaris, and probably BSD variants.

To use ARP (MAC) access controls, you first need to compile in the optional code. Do this with the --enable-arp-acl configure option:

```
% ./configure --enable-arp-acl ...
% make clean
% make
```

If everything compiles, then you can add some ARP ACL lines to your squid.conf

Default

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
```

```
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
```

Example
acl ACLARP arp 11:12:13:14:15:16
ACLARP refers MACADDRESS of the ethernet 11:12:13:14:15:16

Note
Squid can only determine the MAC address for clients that are on the same subnet. If the client is on a different subnet, then Squid cannot find out its MAC address.

Tag Name **http_access**
Usage `http_access allow|deny [!]aclname ...`

Description
Allowing or denying http access based on defined access lists

If none of the "access" lines cause a match, the default is the opposite of the last line in the list. If the last line was deny, then the default is allow. Conversely, if the last line is allow, the default will be deny. For these reasons, it is a good idea to have a "deny all" or "allow all" entry at the end of your access lists to avoid potential confusion

Default

```
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
```

If there are no "access" lines present, the default is to allow the request

Example
1. To allow http_access for only one machine with MAC Address 00:08:c7:9f:34:41
2. To restrict access to work hours (9am - 5pm, Monday to Friday) from IP 192.168.2/24
3. Can i use multitime access control list for different users for different timing
4. Rules are read from top to bottom

Caution
The deny all line is very important. After all the http_access rules, if access isn't denied, it's ALLOWED !! So, specifying a LOT of http_access allow rules, and forget the deny all after them, is the same of NOTHING. If access isn't allowed by one of your rules, the default action (ALLOW) will be triggered. So, don't forget the deny all rule AFTER all the rules.

And, finally, don't forget rules are read from top to bottom. The first rule matched will be used. Other rules won't be applied. Click here to See examples.

Tag Name **icp_access**
Usage `icp_access allow|deny [!]aclname`

Usage icp_access allow|deny [!]aclname ...

Description
icp_access allow|deny [!]aclname ...

Default icp_access deny all

Example
icp_access allow all - Allow ICP queries from everyone

Tag Name **miss_access**
Usage miss_access allow|deny [!]aclname...

Description
Used to force your neighbors to use you as a sibling instead of a parent. For example:

```
acl localclients src 172.16.0.0/16
miss_access allow localclients
miss_access deny !localclients
```

This means that only your local clients are allowed to fetch MISSES and all other clients can only fetch HITS.

Default By default, allow all clients who passed the http_access rules to fetch MISSES from us.
miss_access allow all

Tag Name **cache_peer_access**
Usage cache_peer_access cache-host allow|deny [!]aclname ...

Description
Similar to 'cache_peer_domain ' but provides more flexibility by using ACL elements.

The syntax is identical to 'http_access' and the other lists of ACL elements. See 'http_access ' for further reference.

Default none

Example
The following example could be used, if we want all requests from a specific IP address range to go to a specific cache server (for accounting purposes, for example). Here, all the requests from the 10.0.1.* range are passed to proxy.visolve.com, but all other requests are handled directly.

Using acls to select peers,

```
acl myNet src 10.0.0.0/255.255.255.0
```

```
acl myNet src 10.0.0.0/255.255.255.0
acl cusNet src 10.0.1.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
cache_peer proxy.visolve.com parent 3128 3130
cache_peer_access proxy.visolve.com allow custNet
cache_peer_access proxy.visolve.com deny all
```

Tag Name **proxy_auth_realm**
Usage `proxy_auth_realm string`

Description
Specifies the realm name, which is to be reported to the client for proxy authentication (part of the text the user will see when prompted for the username and password).

Default `proxy_auth_realm Squid proxy-caching web server`

Example
proxy_auth_realm My Caching Server

Tag Name **ident_lookup_access**
Usage `ident_lookup_access allow|deny aclname...`

Description
A list of ACL elements, which, if matched, cause an ident (RFC 931) lookup to be performed for this request. For example, you might choose to always perform ident lookups for your main multi-user Unix boxes, but not for your Macs and PCs

Default By default, ident lookups are not performed for any requests
ident_lookup_access deny all

Example
To enable ident lookups for specific client addresses, you can follow this example:

```
acl ident_aware_hosts src 198.168.1.0/255.255.255.0
ident_lookup_access allow ident_aware_hosts
ident_lookup_access deny all
```

Caution
This option may be disabled by using --disable-ident with the configure script.

Examples:

(1) To allow http_access for only one machine with MAC Address 00:08:c7:9f:34:41

To use MAC address in ACL rules. Configure with option -enable-arp-acl.

```
acl all src 0.0.0.0/0.0.0.0
acl pl800_arp arp 00:08:c7:9f:34:41
http_access allow pl800_arp
http_access deny all
```

(2) To restrict access to work hours (9am - 5pm, Monday to Friday) from IP 192.168.2/24

```
acl ip_acl src 192.168.2.0/24
acl time_acl time M T W H F 9:00-17:00
http_access allow ip_acl time_acl
http_access deny all
```

(3) Can i use multitime access control list for different users for different timing.

AclDefnitions

```
acl abc src 172.161.163.85
acl xyz src 172.161.163.86
acl asd src 172.161.163.87
acl morning time 06:00-11:00
acl lunch time 14:00-14:30
acl evening time 16:25-23:59
```

Access Controls
http_access allow abc morning
http_access allow xyz morning lunch
http_access allow asd lunch

This is wrong. The description follows:
Here access line "http_access allow xyz morning lunch" will not work. So ACLs are interpreted like this ...

```
http_access RULE statement1 AND statement2 AND statement3 OR
http_access ACTION statement1 AND statement2 AND statement3 OR
```

……..

So, the ACL "http_access allow xyz morning lunch" will never work, as pointed, because at any given time, morning AND lunch will ALWAYS be false, because both morning and lunch will NEVER be true at the same time. As one of them is false, and acl uses AND logical statement, 0/1 AND 0 will always be 0 (false).

That's because this line is in two. If now read:

```
http_access allow xyz AND morning OR
http_access allow xyz lunch
```

If request comes from xyz, and we're in one of the allowed time, one of the rules will match TRUE. The other will obviously match FALSE. TRUE OR FALSE will be TRUE, and access will be permitted.

Finally Access Control looks...

```
http_access allow abc morning
http_access allow xyz morning
http_access allow xyz lunch
http_access allow asd lunch
http_access deny all
```

(4) Rules are read from top to bottom. The first rule matched will be used. Other rules won't be applied.

Example:

http_access allow xyz morning
http_access deny xyz
http_access allow xyz lunch

If xyz tries to access something in the morning, access will be granted. But if he tries to access something at lunchtime, access will be denied. It will be denied by the deny xyz rule, that was matched before the 'xyz lunch' rule.

ADMINISTRATIVE PARAMETERS

Tag Name **cache_mgr**
Usage `cache_mgr Administrator mailid`

Description
Using this tag, we can specify the email-address of the local cache manager who will receive mail, if the cache dies. The default is "webmaster." In case squid dies, the mail will be sent to the given mailid.

Default `cache_mgr webmaster`

Example
cache_mgr balu

Tag Name **cache_effective_user**
Usage `cache_effective_user userid`

Description
If the cache is run as root, it will change its effective/real UID to the UID specified below. The default is to change the UID to nobody. If Squid is not started as root, the default is to keep the current UID. Note that, if Squid is not started as root, then you cannot set http_port to a value lower than 1024.

If Squid is started with the userid squid, then the cache_effective_user will be given as squid. If it is started as root then the default value will be taken

Default `cache_effective_user nobody`

Example
cache_effective_user squid

Caution
If the above directive is not configured properly then Squid will have problems in starting.

Tag Name **cache_effective_group**
Usage `cache_effective_group groupid`

Description
If the cache is run as root, it will change its effective/real GID to the GID specified below. The default is to change the GID to nogroup. If Squid is not started as root, the default is to keep the current GID. Note that if Squid is not started as root then you cannot set http_port to a value lower than 1024.

If the squid started with the groupid squid then the cache_effective_group will be given as squid. If it is started as root, then the default value will be taken. For further info on the above two tags, see Effective user and group

Default `cache_effective_group nogroup`

Example
cache_effective_group squid

Caution
If it is not configured properly, then squid may run in to problem.

Tag Name **visible_hostname**
Usage `visible_hostname anyhostname`

Description
If you want to present a special hostname in error messages, etc then define this. Otherwise, the return value of gethostname () will be used. If you have multiple caches in a cluster and get errors about IP-forwarding, you must set them to have individual names with this setting. When the error message appears in the browser, it will show as it is generated from the visible_hostname.

Default none

Example
visible_hostname www.visolve.com

Tag Name **unique_hostname**
Usage unique_hostname hostname

Description
If you want to have multiple machines with the same 'visible_hostname' then you must give each machine a different unique_hostname' so that forwarding loops can be detected. In brief, Just set visible_hostname to the address the clients connects to, and unique_hostname to the externally visible address of each proxy. (address == registered domain name)

Default none

Example
unique_hostname www.kovaiteam.com

Tag Name **hostname_aliases**
Usage hostname_aliases

Description
A list of other DNS names that your cache has. This option is used to detect internal requests (Cache Digests), when a cache has more than one hostname in use.

Default none

Example
hostname_aliases

OPTIONS FOR THE CACHE REGISTRATION SERVICE

This section contains parameters for the (optional) cache announcement service. This service is provided to help cache administrators locate one another in order to join or create cache hierarchies.

An 'announcement' message is sent (via UDP) to the registration service by Squid. By default, the announcement message is NOT SENT unless you enable it with

'announce_period' below.

The announcement message includes your hostname, plus the following information from this configuration file:

- http_port
- icp_port
- cache_mgr

All current information is processed regularly and made available on the Web at http://ircache.nlanr.net/Cache/Tracker/

Tag Name **announce_period**
Usage `announce_period time units`

Description
This is how frequently to send cache announcements. To announce your cache, enable this tag.

Default The default is `0' which disables sending the announcement messages.

```
announce_period 0
```

Example
To send cache announcements every day, use this line

announce_period 1 day

Tag Name **announce_host**
announce_port
Usage `announce_host hostname`

`announce_port port`

Description
announce_host and announce_port set the hostname and port number where the registration message will be sent.

Default Hostname will default to 'tracker.ircache.net' and port will default to 3131.

```
announce_host tracker.ircache.net
announce_port 3131
```

Example
announce_host cache.kovaiteam.com

announce_port 3131

Tag Name **announce_file**
Usage announce_file filename

Description
If the 'filename' argument is given, the contents of that file will be included in the announce message. See also announce_host and announce_port.

Default none

HTTPD-ACCELERATOR OPTIONS

Tag Name **httpd_accel_host**
Usage httpd_accel_host hostname(IP)|virtual

Description
This tag is used to set the hostname of the accelerated server. It is possible to have only one destination server and hence there can be only one occurrence of this line. If you are going to accelerate more than one server, or transparently cache traffic, you will have to use the word virtual instead of a hostname here

Default none

Example
```
httpd_accel_host 172.16.1.115
httpd_accel_host virtual
httpd_accel_host CACHE
```

Caution
Enabling httpd_accel_host disables proxy-caching and ICP. If you want these features enabled also, then set the httpd_accel_with_proxy' option.

Tag Name **httpd_accel_port**
Usage httpd_accel_port port

Description
Accelerated requests can only be forwarded to one port: There is no table that associates accelerated hosts and a destination port. Squid will connect to the port that

you set the httpd_accel_port value to. When acting as a front-end for a web server on the local machine, you will set up the web server to listen for connections on a different port (8000, for example), and set this squid.conf option to match the same value. If, on the other hand, you are forwarding requests to a set of slow backend servers, they will almost certainly be listening to port 80 (the default web-server port), and this option will need to be set to 80. If you want virtual port support then specify the port as "0"

Default `none`

Example
httpd_accel_port 80
httpd_accel_port 8000

Tag Name **httpd_accel_single_host**
Usage `httpd_accel_single_host on|off`

Description
If you are running Squid as an accelerator and have a single backend server then set this to on. This causes Squid to forward the request to this server regardless of what any redirectors or Host headers says.

Leave this at off if you have multiple backend servers, and use a redirector (or host table or private DNS) to map the requests to the appropriate backend servers. See also redirect_rewrites_host_header

Default `httpd_accel_single_host off`

Caution
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.

Tag Name **httpd_accel_with_proxy**
Usage `httpd_accel_with_proxy on|off`

Description
If you use the httpd_accel_host option, Squid will stop recognizing cache requests. So that your cache can function both as an accelerator and as a web cache, you will need to set the httpd_accel_with_proxy option to on.

Default `httpd_accel_with_proxy off`

Tag Name **httpd_accel_uses_host_header**
Usage `httpd_accel_uses_host_header on|off`

Description
HTTP/1.1 requests include a Host: header, which is basically the hostname from the URL. Squid can be an accelerator for different HTTP servers by looking at this header.

URL. Squid can be an accelerator for different HTTP servers by looking at this header.
However, Squid does NOT check the value of the Host header, so it opens a big security hole. It is recommended that this option remain disabled unless having good understanding.

However, It is needed to enable this option if Squid run as a transparent proxy. Otherwise, virtual servers, which require the Host: header will not be properly cached. For detailed information, Click here

Default `httpd_accel_uses_host_header off`

Caution
If Squid runs as a transparent proxy, It is needed to enable this option.

MISCELLANEOUS

Tag Name **dns_testnames**
Usage `dns_testnames URL`

Description
The DNS tests exit as soon as the first site is successfully looked up This test can be disabled with the -D command line option.

Default `dns_testnames netscape.com internic.net nlanr.net microsoft.com`

Example
dns_testnames visolve.com

Tag Name **append_domain**
Usage `append_domain domainname`

Description
Appends local domain name to hostnames without any dots in them. append_domain must begin with a period

Default `none`

Example
append_domain .domain.com

Tag Name **logfile_rotate**
Usage `logfile_rotate NUMBER`

Description
Specifies the number of logfile rotations to make when you type 'squid -k rotate'. The default is 10, which will rotate with extensions 0 through 9. Setting logfile_rotate to 0 will disable the rotation, but the logfiles are still closed and re-opened. This will enable you to rename the logfiles yourself just before sending the rotate signal.

Note, the 'squid -k rotate' command normally sends a USR1 signal to the running squid process. In certain situations (e.g. on Linux with Async I/O), USR1 is used for other purposes, so -k rotate uses another signal. It is best to get in the habit of using 'squid -k rotate' instead of 'kill -USR1'.

Default `logfile_rotate 10`

Example
logfile_rotate 5

Caution
Note, the 'squid -k rotate' command normally sends a USR1 signal to the running squid process. In certain situations (e.g. on Linux with Async I/O), USR1 is used for other purposes; so -k rotate uses another signal. It is best to get in the habit of using 'squid -k rotate' instead of 'kill -USR1 '

Tag Name **dns_testnames**
Usage `dns_testnames URL`

Description
The DNS tests exit as soon as the first site is successfully looked up This test can be disabled with the -D command line option.

Default
```
dns_testnames netscape.com internic.net nlanr.net
microsoft.com
```

Example
dns_testnames visolve.com

Tag Name **logfile_rotate**
Usage `logfile_rotate NUMBER`

Description
Specifies the number of logfile rotations to make when you type 'squid -k rotate'. The default is 10, which will rotate with extensions 0 through 9. Setting logfile_rotate to 0 will disable the rotation, but the logfiles are still closed and re-opened. This will enable you to rename the logfiles yourself just before sending the rotate signal.

Specifies the number of logfile rotations to make when you type 'squid -k rotate'. The default is 10, which will rotate with extensions 0 through 9. Setting logfile_rotate to 0 will disable the rotation, but the logfiles are still closed and re-opened. This will enable you to rename the logfiles yourself just before sending the rotate signal.

Default `logfile_rotate 10`

Example
logfile_rotate 5

Caution
Note, the 'squid -k rotate' command normally sends a USR1 signal to the running squid process. In certain situations (e.g. on Linux with Async I/O), USR1 is used for other purposes; so -k rotate uses another signal. It is best to get in the habit of using 'squid -k rotate' instead of 'kill -USR1 '

Tag Name **tcp_recv_bufsize**
Usage `tcp_recv_bufsize (bytes)`

Description
Size of receive buffer to set for TCP sockets. Probably just as easy to change your kernel's default.

Default
```
Set to zero to use the default buffer size. By default,
if this is set to zero, then it means it is using
kernel's default.

tcp_recv_bufsize 0 bytes
```

Tag Name **err_html_text**
Usage `err_html_text text`

Description
This is used to specirfy the HTML text, which is to be included in error messages. Make this a "mailto" URL to your administrator address, or may be just a link to your organizations Web page.

To include this in your error messages, you must rewrite the error template files (found in the "$prefix/etc/errors" directory). Wherever you want the 'err_html_text' line to appear, insert a %L tag in the error template file

Default `none`

Example
err_html_text venkatesh@visolve.com

Consider you want to display this mail Id when access denied error occurs, then edit the corresponding file (ERR_ACCESS_DENIED in '$prefix/etc/errors' directory) with %L where this mail Id should be displayed.

Tag Name **deny_info**

Tag Name deny_info
Usage deny_info err_page_name acl

Description
This can be used to return an ERR_ page for requests, which do not pass the 'http_access' rules. A single ACL will cause the http_access check to fail. If a 'deny_info' line exists for that ACL then Squid returns a corresponding error page.

You may use ERR_ pages that come with Squid or create your own pages and put them into the configured errors/ directory

Default none

Example
If you want to deny domain 'deny.com' and want to display access denied message specifically, add these lines in conf. And add the file called ERR_CUSTOM_ACCESS_DENIED in $prefix/etc/errors/ directory with your own format.

acl DSTDOMAIN dstdomain .deny.com

http_access deny DSTDOMAIN

http_access allow all

deny_info ERR_CUSTOM_ACCESS_DENIED DSTDOMAIN

So now if users try to browse 'deny.com' they will get your defined error message

Tag Name memory_pools
Usage memory_pools on|off

Description
If set, Squid will keep pools of allocated (but unused) memory available for future use. If memory is a premium on your system and you believe your malloc library outperforms Squid routines, disable this.

Default memory_pools on

Tag Name memory_pools_limit
Usage memory_pools_limit (bytes)

Description
If set to a non-zero value, Squid will keep at most the specified limit of allocated (but unused) memory in memory pools. All free() requests that exceed this limit will be handled by your malloc library. Squid does not pre-allocate any memory, just safe-keeps objects that otherwise would be free()d. Thus, it is safe to set

keeps objects that otherwise would be free()d. Thus, it is safe to set memory_pools_limit to a reasonably high value even if your configuration will use less memory.

If not set (default) or set to zero, Squid will keep all memory it can. That is, there will be no limit on the total amount of memory used for safe-keeping.

aa

Default `none`

```
By default, memory_pools is not set. So there is no
default value for memory_pools_limit
```

Caution
Used only with memory_pools on: To disable memory allocation optimization, do not set memory_pools_limit to 0. Set memory_pools to "off" instead. An overhead for maintaining memory pools is not taken into account when the limit is checked. This overhead is close to four bytes per object kept. However, pools may actually _save_ memory because of reduced memory thrashing in your malloc library.

Tag Name **forwarded_for**
Usage `forwarded_for on|off`

Description
Current HTTP/1.1 does not provide any standard way of indicating the client address in the request. Since a number of people missed having the originating client address in the request, Squid now adds its own request header called "X-Forwarded-For" which looks like this: X-Forwarded-For: 192.1.2.3|unknown

If set, Squid will include your system's IP address or name in theHTTP requests it forwards. By default it looks like this: X-Forwarded-For: 192.1.2.3

If you disable this, it will appear as X-Forwarded-For: unknown

Default `forwarded_for on`

Tag Name **log_icp_queries**
Usage `log_icp_queries on|off`

Description
If set, ICP queries are logged to access.log. You may wish to disable this if your ICP load is very high to speed things up or to simplify log analysis

Default `log_icp_queries on`

Tag Name **icp_hit_stale**
Usage `icp_hit_stale on|off`

Description
If you want to return ICP_HIT for stale cache objects, set this option to 'on'. If you have sibling relationships with caches in other administrative domains, this should be 'off'. If you only have sibling relationships with caches under your control, then it is probably okay to set this to 'on'

Default `icp_hit_stale off`

Tag Name **minimum_direct_hops**
Usage `minimum_direct_hops NUMBER`

Description
If using the ICMP pinging stuff, do direct fetches for sites which are no more than this many hops away. This parameter plays a role in deciding latency

Default `minimum_direct_hops 4`

Tag Name **minimum_direct_rtt**
Usage `minimum_direct_rtt time-units`

Description
If using the ICMP pinging stuff, do direct fetches for sites which are no more than this many rtt milliseconds away.

Default `minimum_direct_rtt 400`

Tag Name **cachemgr_passwd**
Usage `cachemgr_passwd password action action ...`

Description

Description:

This tag is used to specify passwords for cachemgr operations. Some valid actions are (see cache manager menu for a full list):

5min

60min

asndb

authenticator

cbdata

client_list

comm_incoming

config *

counters delay

digest_stats

dns

events

filedescriptors

fqdncache

histograms

http_headers

info

io

ipcache

mem menu

netdb

non_peers

objects

pconn

peer_select

peer_select

redirector

refresh

server_list

shutdown *

store_digest

storedir

utilization

via_headers

vm_objects

* Indicates actions which will not be performed without a valid password, others can be performed if not listed here.

To disable an action, set the password to "disable".

To allow performing an action without a password, set the password to "none".

Use the keyword "all" to set the same password for all actions.

cachemgr_passwd secret shutdown

cachemgr_passwd lesssssssecret info stats/objects

cachemgr_passwd disable all

Default none

Tag Name **store_avg_object_size**
Usage store_avg_object_size (kbytes)

Description
Average object size, used to estimate number of objects your cache can hold. To Estimate the number of objects your cache can hold: NUM_OBJ = cache_swap / store_avg_object_size Cache_swap is the size of the cache

Default The default is 13 KB.

```
store_avg_object_size 13 KB
```

Tag Name **store_objects_per_bucket**
Usage `store_objects_per_bucket (kbytes)`

Description
Target number of objects per bucket in the store hash table. Lowering this value increases the total number of buckets and also the storage maintenance rate. Then we estimate the number of hash buckets needed: NUM_BUCKETS = NUM_OBJ / store_objects_per_bucket NUM_OBJ is the number of objects your cache can hold, estimated by store_avg_object_size.

Default `store_objects_per_bucket 20`

Example
store_objects_per_bucket 50

Tag Name **client_db**
Usage `client_db on|off`

Description
If you want to disable collecting per-client statistics, then turn off client_db here

Default `client_db on`

Tag Name **netdb_low**
netdb_high
Usage `netdb_low entries`

`netdb_high entries`

Description
The low and high water marks for the ICMP measurement database. These are counts, not percents. The defaults are 900 and1000. When the high water mark is reached,

not percents. The defaults are 900 and1000. When the high water mark is reached, database entries will be deleted until the low mark is reached

Default `netdb_low 900`

`netdb_high 1000`

Tag Name **netdb_ping_period**
Usage `netdb_ping_period time-units`

Description
The minimum period for measuring a site. There will be at least this much delay between successive pings to the same network

Default `netdb_ping_period 5 minutes`

Tag Name **query_icmp**
Usage `query_icmp on|off`

Description
If you want to ask your peers to include ICMP data in their ICP replies, enable this option. If your peer has configured Squid (during compilation) with '--enable-icmp' then that peer will send ICMP pings to origin server sites of the URLs it receives. If you enable this option then the ICP replies from that peer will include the ICMP data (if available). Then, when choosing a parent cache, Squid will choose the parent with the minimal RTT to the origin server.When this happens, the hierarchy field of the access.log will be "CLOSEST_PARENT_MISS ".

Default `query_icmp off`

Tag Name **test_reachability**
Usage `test_reachability on|off`

Description
When this is 'on', ICP MISS replies will be ICP_MISS_NOFETCH instead of ICP_MISS if the target host is NOT in the ICMP database, or has a zero RTT

Default `test_reachability off`

Tag Name **buffered_logs**
Usage `buffered_logs on|off`

Description
Some log files (cache.log useragent.log) are written with stdio functions, and as such they can be buffered or unbuffered. By default they will be unbuffered. Buffering them can speed up the writing slightly (though you are unlikely to need to worry).

Default `buffered_logs off`

Tag Name **reload_into_ims**
Usage `reload_into_ims on|off`

Description
When you enable this option, client no-cache or "reload'' requests will be changed to If-Modified-Since requests. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems, which it causes.

See also refresh_pattern for a more selective approach.

This option may be disabled by using --disable-http-violations with the configure script.
reload_into_ims off

Default `reload_into_ims off`

Tag Name **always_direct**
Usage `always_direct allow|deny [!]aclname ...`

Description
Here you can use ACL elements to specify requests, which should ALWAYS be forwarded directly to origin servers. This is mostly used while using cache_peer. See also never_direct . For Further reference on always_direct, please click here.

Default `always_direct is by default deny.`

Example
For example, to always directly forward requests for local servers use something like:

acl local-servers dstdomain .my.domain.net

always_direct allow local-servers

To always forward FTP requests directly, use

acl FTP proto FTP

always_direct allow FTP

Example for denying specific domain

acl local-external dstdomain .external.foo.net

acl local-servers dstdomain .foo.net

always_direct deny local-external

always_direct allow local-servers

Caution
There is a similar, but opposite option named ' never_direct'. You need to be aware that "always_direct deny foo" is NOT the same thing as "never_direct allow foo". You may need to use a deny rule to exclude a more-specific case of some other rule

Tag Name **never_direct**
Usage `never_direct allow|deny [!]aclname ...`

Description
never_direct is the opposite of always_direct. Please read the description for always_direct if you have not already.

With 'never_direct' you can use ACL elements to specify requests, which should NEVER be forwarded directly to origin servers

When always_direct and never_direct are deny (By default), Squid selects based on the request type and a number of other factors if a parent should be used or not, and if a parent could not be reached it will always fallback on direct.

If always_direct is allow then Squid will always go direct to the source without considering any peers.

If never_direct is allow then Squid will never attempt to go direct to the source. Instead it tries very hard to find a parent to send the request to. If no parent can be found then an error is returned. For Further reference on never_direct, please clickhere.

Default never_direct is by default deny.

Default never_direct is by default deny.

Example
For example, to force the use of a proxy for all requests, except those in your local domain use something like:

acl local-servers dstdomain foo.net

acl all src 0.0.0.0/0.0.0.0

never_direct

deny local-servers

never_direct allow all

or if squid is inside a firewall and there are local intranet servers inside the firewall then use something like:

acl local-intranet dstdomain .foo.net

acl local-external dstdomain .external.foo.net

always_direct deny local-external

always_direct allow local-intranet

never_direct allow all

Caution
It will be better to understand always_direct before enabling this tag

Tag Name **anonymize_headers**
Usage `anonymize_headers allow|deny header_name ...`

Description
This option replaces the old 'http_anonymizer' option with something that is much more configurable. You may now specify exactly which headers are to be allowed, or which are to be removed from outgoing requests.

There are two methods of using this option. You may either allow specific headers (thus denying all others), or you may deny specificheaders (thus allowing all others).

For example, to achieve the same behavior as the old 'http_anonymizer standard' option, you

should use:

anonymize_headers deny From Referer Server

anonymize_headers deny User-Agent WWW-Authenticate Link

Or, to reproduce the old 'http_anonymizer paranoid' feature you should use:

anonymize_headers allow Allow Authorization Cache-Control

anonymize_headers allow Content-Encoding Content-Length

anonymize_headers allow Content-Type Date Expires Host

anonymize_headers allow If-Modified-Since Last-Modified

anonymize_headers allow Location Pragma Accept

anonymize_headers allow Accept-Encoding Accept-Language

anonymize_headers allow Content-Language Mime-Version

anonymize_headers allow Retry-After Title Connection

anonymize_headers allow Proxy-Connection

Default `By default, all headers are allowed (no anonymizing is performed).`

Example
anonymize_headers deny Proxy-Connection

Caution
You cannot mix "allow" and "deny". All 'anonymize_headers' lines must have the same second argument.

Tag Name **fake_user_agent**
Usage `fake_user_agent String`

Description
If you filter the User-Agent header with 'anonymize_headers' it may cause some Web servers to refuse your request. Use this to fake one up.

Default If you filter the User-Agent header with ' anonymize_headers' it may cause some Web servers to refuse your request. Use this to fake one up.

Example
fake_user_agent Nutscrape/1.0 (CP/M; 8-bit)vv

Tag Name **icon_directory**
Usage icon_directory directorypath/directoryname

Description
This tag is to specify the location where the icons are stored

Default These are normally kept in /usr/local/squid/etc/icons

Example
icon_directory /etc/icons

Tag Name **error_directory**
Usage error_directory directorypath/directoryname

Description
If you wish to create your own versions of the default (English) error files, either to customize them to suit your language or company, copy the template English files to another directory and point this tag at them

Default These are normally kept in /usr/local/squid/etc/errors

Example
icon_directory /etc/errors

Tag Name **minimum_retry_timeout**
Usage minimum_retry_timeout (seconds)

Description
This specifies the minimum connect timeout, when the connect timeout is reduced to compensate for the availability of multiple IP addresses. When a connection to a host is initiated, and that host has several IP addresses, the default connection timeout is reduced by dividing it by the number of addresses. So, a site with15 addresses would

reduced by dividing it by the number of addresses. So, a site with15 addresses would then have a timeout of 8 seconds for each address attempted. To avoid having the timeout reduced to the point where even a working host would not have a chance to respond, this setting is provided.

Default The default, and the minimum value, is five seconds, and the maximum value is sixty seconds, or half of connect_timeout, which ever is greater and less than connect_timeout.

```
minimum_retry_timeout 5 seconds
```

Tag Name **maximum_single_addr_tries**
Usage `maximum_single_addr_tries NUMBER`

Description
This sets the maximum number of connection attempts for a host that only has one address (for multiple-address hosts, each address is tried once)

Default The default value is three tries, the (not recommended) maximum is 255 tries. `maximum_single_addr_tries 3`

Caution
A warning message will be generated if it is set to a value greater than ten

Tag Name **snmp_port**
Usage `snmp_port port`

Description
Squid can now serve statistics and status information via SNMP. If you don't wish to use SNMP, set this to "0".

The snmpd daemon is a server that supports both the Simple Network Management Protocol v2 and v1. It receives and responds to SNMP messages sent to the SNMP port on the local machine. snmpd.conf is the configuration file which defines how the ucd-smnp SNMP agent operates.

Default By default it listens to port 3401 on the machine.
`snmp_port 3401`

Caution
SNMP support requires use of the --enable-snmp configure command line option

Tag Name **snmp_access**
Usage `snmp_access allow|deny [!]aclname ...`

Description
Allowing or denying access to the SNMP port. This option is only available if Squid is rebuilt with the --enable-snmp option

Default `All access to the agent is denied by default.`

Example
snmp_access allow snmppublic localhost

snmp_access deny all

Tag Name **snmp_incoming_address**
snmp_outgoing_address
Usage
```
snmp_incoming_address IPAddress

snmp_outgoing_address IPAddress
```

Description
Just like 'udp_incoming_address' above, but for the SNMP port. This option is only available if Squid is rebuilt with the--enable-snmp option

snmp_incoming_address is used for the SNMP socket receiving messages from SNMP agents. snmp_outgoing_address is used for SNMP packets returned to SNMP agents. See also snmp_port

Default
```
The default behavior is to not bind to any specific
address.

snmp_incoming_address 0.0.0.0

snmp_outgoing_address 255.255.255.255
```

Example
snmp_incoming_address 172.16.1.115

snmp_outgoing_address 172.16.1.114

Caution
snmp_incoming_address and snmp_outgoing_address cannot have the same value since they both use port 3130.

Tag Name **as_whois_server**
Usage as_whois_server Server-Name

Description
WHOIS server to query for AS numbers. NOTE: AS numbers are queried only when Squid starts up, not for every request.

Default as_whois_server whois.ra.net

Tag Name **wccp_router**
Usage wccp_router Router-IPAddress

Description
This option is used to define the WCCP ``home'' router for Squid. Setting the 'wccp_router' to 0.0.0.0 (the default) disables WCCP

Default wccp_router 0.0.0.0

Tag Name **wccp_version**
Usage wccp_version Version

Description
According to some users, Cisco IOS 11.2 only supportsWCCP version 3. If you're using that version of IOS, change this valueto 3

Default wccp_version 4

Tag Name **wccp_incoming_address**
Usage wccp_incoming_address IPAddress

Description
Use this option if you require WCCP messages to be received on only one interface. Do NOT use this option if you're unsure how many interfaces you have, or if you know you have only one interface

Default The default behavior is to not bind to any specific address

```
wccp_incoming_address 0.0.0.0
```

Caution
wccp_incoming_address and wccp_outgoing_address cannot have the same value since they both use port 2048.

Tag Name **wccp_outgoing_address**
Usage `wccp_outgoing_address IPAddress`

Description
Use this option if you require WCCP messages to be sent out on only one interface. Do NOT use this option if you're unsure how many interfaces you have, or if you know you have only one interface

Default The default behavior is to not bind to any specific address `wccp_outgoing_address 255.255.255.255`

Caution
wccp_incoming_address and wccp_outgoing_address cannot have the same value since they both use port 2048.

DELAYPOOL PARAMETERS

Tag Name **delay_pools**
Usage `delay_pools numbers`

Description
This represents the number of delay pools to be used. For example, if you have one-class 2 delay pool and one- class 3 delay pool, you have a total of 2 delay pools. Delay pools allow you to limit traffic for clients or client groups, with various features. Objects retrieved from the cache will not be delayed. Only the object from the server will be

delayed.

Default `delay_pools 0`

Example
delay_pools 2 # 2 Delay pools

Caution
To enable this option, you must use --enable-delay-pools with the # configure script.

Tag Name **delay_class**
Usage `delay_class number (delay-pool number), number (delay class)`

Description
This defines the class of each delay pool. There must be exactly one delay_class line for each delay pool. For example, to define two delay pools, one of class 2 and one of class 3, the settings will be like as given in the example. For details on the delay pool classes see Glossary.

Default `none`

Example
delay_pools 2 # 2 delay pools
delay_class 1 2 # pool 1 is a class 2 pool
delay_class 2 3 # pool 2 is a class 3 pool

Caution
To enable this option, you must use --enable-delay-pools with the # configure script.

Tag Name **delay_access**
Usage `delay_access allow acl name|deny acl name`

Description
This is used to determine which delay pool a request falls into. The first matched delay pool is always used, i.e., if a request falls into delay pool number one, no more delay are checked, otherwise the rest are checked in order of their delay pool number until they have all been checked. For example, if you want pool_1_acl in delay pool 1 and pool_2_acl in delay pool 2, then look at the example below.

Default `none`

Example
To specify which pool a client falls into, create ACLs which specifies the ip ranges for each pool, and use the following:

delay_access 1 allow pool_1_acl
delay_access 1 deny all
delay_access 2 allow pool_2_acl

delay_access 2 deny all

Caution
To enable this option, you must use --enable-delay-pools with the # configure script.

Tag Name **delay_parameters**
Usage
```
delay_parameters pool aggregate (for delay_class 1
networks)
delay_parameters pool aggregate individual (for
delay_class 2 networks)
delay_parameters pool aggregate network individual (for
delay_class 3 networks)
```

Description
This defines the parameters for a delay pool. Each delay pool has number of "buckets" associated with it, as explained in the description of delay_class. For a class 1,2 and 3 delay pool, the syntax is given in the usage. For a Glossary of term related to delay_pool see Glossary .

Default none

Example 1:
```
acl tech src 192.168.0.1-192.168.0.20/32
```
acl no_hotmail url_regex -i hotmail
acl all src 0.0.0.0/0.0.0.0
delay_pools 1 #Number of delay_pool 1
delay_class 1 1 #pool 1 is a delay_class 1
delay_parameters 1 100/100
delay_access 1 allow no_hotmail !tech

In the above example, hotmail users are limited to the speed specified in the delay_class. IP's in the ACL tech are allowed in the normal bandwidth. You can see the usage of bandwidth through cachemgr.cgi.

Example 2:
```
acl all src 0.0.0.0/0.0.0.0 # might already be defined
```
delay_pools 1
delay_class 1 1
delay_access 1 allow all
delay_parameters 1 64000/64000 # 512 kbits == 64 kbytes per second

The above example tells that the squid is limited to the bandwidth of 512k. For ACL you can go Here .

Caution
To enable this option, you must use --enable-delay-pools with the # configure script.

Tag Name **delay_initial_bucket_level(percent, 0-100)**
Usage
```
delay_initial_bucket_level bytes
```

Description
The initial bucket percentage is used to determine how much is put in each bucket when squid starts, is reconfigured, or first notices a host accessing it (in class 2 and class 3, individual hosts and networks only have buckets associated with them once they have been "seen" by squid).

Default `delay_initial_bucket_level 50(bytes)`

Caution
This option is only available if Squid is rebuilt with the --enable-delaypools option.

Tag Name
incoming_icp_average
incoming_http_average
incoming_dns_average
min_icp_poll_cnt
min_dns_poll_cnt
min_http_poll_cnt

Usage `TagName Number`

Description
This describes the algorithms used for the above tags.

INCOMING sockets are the ICP and HTTP ports. We need to check these fairly regularly, but how often? When the load increases, we want to check the incoming sockets more often. If we have a lot of incoming ICP, then we need to check these sockets more than if we just have HTTP. The variables 'incoming_icp_interval' and 'incoming_http_interval'determine how many normal I/O events to process before checking incoming sockets again. Note we store the incoming_interval multiplied by a factor of (2^INCOMING_FACTOR) to have some pseudo-floating point precision.

The variable 'icp_io_events' and 'http_io_events' counts how many normal I/O events have been processed since the last check on the incoming sockets. When io_events >incoming_interval, its time to check incoming sockets.

Every time we check incoming sockets, we count how many new messages or connections were processed. This is used to adjust the incoming_interval for the next iteration. The new incoming_interval is calculated as the current incoming_interval plus what we would like to see as an average number of events minus the number of events just processed.

incoming_interval = incoming_interval + target_average - number_of_events_processed.

There are separate incoming_interval counters for both HTTP and ICP events. You can see the current values of the incoming_interval, as well as a histogram of 'incoming_events' by asking the cache manager for 'comm_incoming', e.g.:

% ./client mgr:comm_incoming

Default `incoming_icp_average 6`

```
incoming_http_average
4 incoming_dns_average
4 min_icp_poll_cnt 8
min_dns_poll_cnt 8
min_http_poll_cnt 8
```

Caution
-We have MAX_INCOMING_INTEGER as a magic upper limit on incoming_interval for both types of sockets. At the largest value the cache will effectively be idling.

-The higher the INCOMING_FACTOR, the slower the algorithm will respond to load spikes/increases/decreases in demand. A value between 3 and 8 is recommended.

Tag Name **max_open_disk_fds**
Usage `max_open_disk_fds number`

Description
This specifies the maximum file descriptor squid can use to open files. To avoid having disk as the I/O bottleneck, Squid can optionally bypass the on-disk cache if more than this amount of disk file descriptors are open.

A value of 0 indicates no limit

Default `max_open_disk_fds 0`

Tag Name **offline_mode**
Usage `offline_mode on|off`

Description
Enable this option and Squid will never try to validate cached objects. offline_mode gives access to more cached information than the proposed feature would allow (stale cached versions, where the origin server should have been contacted).

Default `offline_mode off`

Tag Name **uri_whitespace**
Usage `uri_whitespace options`

Description
The action to be done on the requests that have whitespace characters in the URI is decided with this tag. Available options:

strip:
The whitespace characters are stripped out of the URL. This is the behavior recommended by RFC2616.

deny:
The request is denied. The user receives an "Invalid Request" message.

allow:
The request is allowed and the URI is not changed. The whitespace characters remain in the URI. Note the whitespace is passed to redirector processes if they are in use.

Encode:
The request is allowed and the whitespace characters are encoded according to RFC1738. This could be considered a violation of the HTTP/1.1 RFC because proxies are not allowed to rewrite URI's.

chop:
The request is allowed and the URI is chopped at the first whitespace. This might also be considered as a violation.

Default `uri_whitespace strip`

Example
```
uri_whitespace chop
```

Tag Name **broken_posts**
Usage `broken_posts allow|deny acl name`

Description
A list of ACL elements which, if matched, causes Squid to send a extra CRLF pair after the body of a PUT/POST request. Some HTTP servers have broken implementations of PUT/POST, and rely on an extra CRLF pair sent by some WWW clients.

Default `none`

Example
acl buggy_server url_regex ^http://....
broken_posts allow buggy_server

Tag Name **mcast_miss_addr**
Usage `mcast_miss_addr enable|disable`

Description
If you enable this option, every "cache miss" URL will be sent out on the specified multicast address. This option is only available if Squid is rebuilt with the -DMULTICAST_MISS_STREAM option.

Default `mcast_miss_addr 255.255.255.255`

Caution
This option should be enabled only after a careful understanding. See multicast

Tag Name **mcast_miss_ttl**

Usage `mcast_miss_ttl time-units`

Description
This is the time-to-live value for packets multicasted when multicasting off cache miss URLs is enabled. This option is only available if Squid is rebuilt with the -DMULTICAST_MISS_TTL option.

Default `mcast_miss_ttl 16`

Tag Name **mcast_miss_port**
Usage `mcast_miss_port port no`

Description
This is the port number to be used in conjunction with 'mcast_miss_addr'. This option is only available if Squid is rebuilt with the -DMULTICAST_MISS_TTL option.

Default `mcast_miss_port 3135`

Caution
This tag is used only when you enable mcast_miss_addr

Tag Name **mcast_miss_encode_key**
Usage `mcast_miss_encode_key enable|disable`

Description
The URLs that are sent in the multicast miss stream are encrypted. This is the encryption key. This option is only available if Squid is rebuilt with the -DMULTICAST_MISS_STREAM option.

Default `mcast_miss_encode_key XXXXXXXXXXXXXXX`

Tag Name **nonhierarchical_direct**
Usage `nonhierarchical_direct on|off`

Description
By default, Squid will send any non-hierarchical requests (matching hierarchy_stoplist or not cacheable request type) direct to origin servers. If you set this to off, then Squid will prefer to send these requests to parents. Note that in most configurations, by turning this off you will only add latency to this request without any improvement in global hit ratio. If you are inside a firewall then see never_direct instead of this directive.

Default `nonhierarchical_direct on`

Tag Name **prefer_direct**
Usage `prefer_direct on|off`

Description
Normally Squid tries to use parents for most requests. If you by some reason like it to first try going direct and only use a parent if going direct fails then set this to off. By combining non hierarchical_direct off and prefer_direct on you can set up Squid to use a parent as a backup path if going direct fails.

Default `prefer_direct off`

Tag Name **strip_query_terms**
Usage `strip_query_terms on|off`

Description
Squid by default does not log query parameters. These parameters are however forwarded to the server verbatim. If we want to enable logging of query parameters, the strip_query_terms directive can be used .

By default, Squid strips query terms from requested URLs before logging. This protects your user's privacy

Default `strip_query_terms on`

Tag Name **coredump_dir**
Usage `coredump_dir directory`

Description
By default Squid leaves core files in the first cache_dir directory. If you set 'coredump_dir' to a directory that exists,Squid will chdir() to that directory at startup and coredump files will be left there.

Default `none`

Example
```
coredump_dir /usr/local
```

Tag Name **redirector_bypass**
Usage `redirector_bypass on|off`

Description
When this is 'on', a request will not go through the redirector if all redirectors are busy. If this is 'off' and the redirector queue grows too large, Squid will exit with a FATAL error and ask you to increase the number of redirectors. You should only enable this if the redirectors are not critical to your caching system.If you use redirectors for access control, and you enable this option,then users may have access to pages that they should not be allowed to request.

Default redirector_bypass off

Tag Name **digest_generation**
Usage digest_generation on|off

Description
This controls whether the server will generate a Cache Digest of its contents. By default, Cache Digest generation is enabled if Squid is compiled with USE_CACHE_DIGESTS defined. This option is only available if Squid is rebuilt with the --enable-cache-digests option.

Default digest_generation on

Tag Name **ignore_unknown_nameservers**
Usage ignore_unknown_nameservers on|off

Description
By default Squid checks that DNS responses are received from the same IP addresses that they are sent to. If they don't match, Squid ignores the response and writes a warning message to cache.log. You can allow responses from unknown nameservers by setting this option to 'off'.

Default ignore_unknown_nameservers on

Tag Name **digest_bits_per_entry**
Usage digest_bits_per_entry number

Description
This is the number of bits of the server's Cache Digest, which will be associated with the Digest entry for a given HTTP Method and URL (public key) combination. The default is 5. This option is only available if Squid is rebuilt with the --enable-cache-digests option.

Default digest_bits_per_entry 5

Tag Name **digest_rebuild_period**
Usage digest_rebuild_period time-units

Description
This is the number of seconds between Cache Digest rebuilds. By default the server's Digest is rebuilt every hour. This option is only available if Squid is rebuilt with the --enable-cache-digests option.

Default digest_rebuild_period 1 hour

Tag Name **digest_rewrite_period**
Usage `digest_rewrite_period time-units`

Description
This is the number of seconds between Cache Digest writes to disk. By default the server's Digest is written to disk every hour. This option is only available if Squid is rebuilt with the--enable-cache-digests option.

Default `digest_rewrite_period 1 hour`

Tag Name **digest_swapout_chunk_size**
Usage `digest_swapout_chunk_size bytes`

Description
This is the number of bytes of the Cache Digest to write to disk at a time. It defaults to 4096 bytes (4KB), the Squid default swap page. This option is only available if Squid is rebuilt with the --enable-cache-digests option.

Default `digest_swapout_chunk_size 4096 bytes`

Tag Name **digest_rebuild_chunk_percentage**
Usage `digest_rebuild_chunk_percentage %(0 to 100)`

Description
This is the percentage of the Cache Digest to be scanned at a time. By default it is set tc 10% of the Cache Digest. This option is only available if Squid is rebuilt with the --enable-cache-digests option.

Default `digest_rebuild_chunk_percentage 10`

Tag Name **chroot**
Usage `chroot enable|disable`

Description
Squid by default does not fully drop root privileges because it may be required during reconfigure.So use this directive to have Squid do a chroot() while initializing. This also causes Squid to fully drop root privileges after initializing . Squid only drops all root privilegies when chroot_dir is used. Without chroot_dir it runs as root with effective user nobody. This means, for example, that if you use a HTTP port less than 1024 and try to reconfigure, you will get an error .

Default `none`

Default none

Tag Name **server_persistent_connections**
Usage server_persistent_connections on|off

Description
Persistent connection support for clients and servers. By default, Squid uses persistent connections (when allowed) with its clients and servers. You can use these options to disable persistent connections with clients and/or servers.

Default server_persistent_connections on

Tag Name **client_persistent_connections**
Usage client_persistent_connections on|off

Description
Persistent connection support for clients and servers. By default, Squid uses persistent connections (when allowed) with its clients and servers. You can use these options to disable persistent connections with clients and/or server.

Related information :

If the browser is talking to web server directly, socket can be closed after it is done using keep-alive directive in apache configuration file. The same thing can be done in Squid using these directives client_persistent_connections and server_persistent_connections.

Default client_persistent_connections on

Tag Name **pipeline_prefetch**
Usage pipeline_prefetch on|off

Description
To boost the performance of pipelined requests to closer match that of a non-proxied environment Squid tries to fetch up to two requests in parallell from a pipeline.

Default pipeline_prefetch on

Tag Name **extension_methods**
Usage extension_methods request_method

Usage extension_methods request method

Description
Squid only knows about standard HTTP request methods. Unknown methods are denied, unless you add them to this list. You can add up to 20 additional "extension" methods here.

Default none

Tag Name **high_response_time_warning**
Usage high_response_time_warning msec

Description
If the one-minute median response time exceeds this value, Squid prints a WARNING with debug level 0 to get the administrators attention. The value is in milliseconds.

Default high_response_time_warning 0

Tag Name **high_page_fault_warning**
Usage high_page_fault_warning time-units

Description
If the one-minute average page fault rate exceeds this value, Squid prints a WARNING with debug level 0 to get the administrators attention. The value is in page faults per second.

Default high_page_fault_warning 0

Tag Name **high_memory_warning**
Usage high_memory_warning number

Description
If the memory usage (as determined by mallinfo) exceeds value, Squid prints a WARNING with debug level 0 to get the administrators attention.

Default high_memory_warning 0

Tag Name **store_dir_select_algorithm**
Usage store_dir_select_algorithm algorithm type

Description
Squid currently supports two algorithms for selecting cache directories for new objects: least-load and round-robin. Set this to 'round-robin' as an alternative.

Default store_dir_select_algorithm least_load

Default [illegible]

Tag Name **ie_refresh**
Usage `ie_refresh on|off`

Description
Microsoft Internet Explorer up until version 5.5 Service Pack 1 has an issue with transparent proxies, where in it is impossible to force a refresh. Turning this on provides a partial fix to the problem, by causing all IMS-REFRESH requests from older IE versions to check the origin server for fresh content. This reduces hit ratio by some amount (~10%), but allows users to actually get fresh content when they want it. Note that because Squid cannot tell if the user is using 5.5 or 5.5SP1, the behavior of 5.5 is unchanged from old versions of Squid (i.e. a forced refresh is impossible).Newer versions of IE will, hopefully, continue to have the new behavior and will be handled based on that assumption. This option defaults to the old Squid behavior, which is better for hit ratios but worse for clients using IE, if they need to be able to force fresh content.

Default `ie_refresh off`

GLOSSARY

parent

In a parent relationship, the child cache will forward requests to its parent cache. If the parent does not hold a requested object, it will forward the request on behalf of the child. A cache hierarchy should closely follow the underlying network topology. Parent caches should be located along the network paths towards the greater Internet. For example, if your Internet Service Provider (ISP) operates a cache, it should probably be a parent to yours, since your Web traffic will have to travel along your ISP's infra structure anyway.

sibling

In a sibling relationship, a peer may only request objects already held in the cache; a sibling can not forward cache misses on behalf of the peer. The sibling relationship should be used for caches ``nearby'' but not in the direction of your route to the

Internet. For example, it may make sense for a number of department-specific caches within an organization to have sibling relationships among them. This approach is even more compelling when there is no parent cache available for the organization as a whole.

Multicast and Unicast

A unicast packet is the complete opposite: one machine is talking to only one other machine. All TCP connections are unicast, since they can only have one destination host for each source host. UDP packets are almost always unicast too, though they can be sent to the broadcast address so that they reach every single machine in some cases.

A multicast packet is from one machine to one or more. The difference between a multicast packet and a broadcast packet is that hosts receiving multicast packets can be on different lans, and that each multicast data-stream is only transmitted between networks once, not once per machine on the remote network. Rather than each machine connecting to a video server, the multicast data is streamed per-network, and multiple machines just listen-in on the multicast data once it's on the network.

Netmask

An IP address has two components, the network address and the host address. For example, consider the IP address 172.16.1.25. Assuming this is part of a Class B network, the first two numbers (172.16) represent the Class B network address, and the second two numbers (1.25) identify a particular host on this network.

Subnetting enables the network administrator to further divide the host part of the address into two or more subnets. In this case, a part of the host address is reserved to identify the particular subnet. This is easier to see if we show the IP address in binary format. The full address is:

11111000.00001000.00000001.00011001 The Class B network part is:
11111000.00001000
and the host address is
00000001.00011001
If the subnetmask for this IP Address is
255.255.255.0,11111111.11111111.11111111.00000000 (binary).
The resultant Subnet Address is got by bitwise AND operations.
If this network is divided into 255 subnets, then the first 8 bits of the host address (00000000) are reserved for identifying the subnet.
11111000.00001000.00000001.00000000 Hence, resultant is 172.16.1.0. It refers IPAddress from 172.16.1.1 to172.16.1.255.

FileSystems in Squid

The cache_dir type in Squid has nothing to do with the underlying filesystem type, it defines the storage method / implementation.

Currently Squid has 4 different implementations:
ufs :- On top of a normal filesystem supporting directories and files.
aufs :- As "ufs", but using threads to implement non-blocking disk I/O
diskd :- As "ufs", but using a separate process to implement non-blocking disk I/O
coss :- An experimental "raw" filesystem, where all objects are stored in one big file.
Other storage methods are being worked upon

Kind of diskd is designed to work around the problem of blocking IO in a unix process. async ufs gets around this by using threads to complete disk IO. diskd uses external processes to complete disk IO.

Asyncufs works just that little bit faster, but only works on systems where threads can do async disk IO without blocking the main process. Systems with user-threads (eg FreeBSD) can not use this effectively. Diskd, being implemented as an external process, gets around this. If cache is slightly active, then the difference cannot be noticed. diskd/aufs are only useful when the cache is under high load.

In case it was not clear, asyncronous I/O (diskd/aufs) is beneficial for single drive configurations with "higher" request loads, in many cases allowing you to push about 100% more I/O thru the drive before latency creeps up too high.

For multiple drive configurations, it is almost a requirement to be able to use the I/O capacity of the extra drives. Without it, a multiple disk configuration is effectively limited to almost the speed of a single disk configuration. With asyncronous I/O, the disk I/O scales quite well (at least for the first few drives, other limits gets very apparent when you have more than ~3 drives).

Cache_peer Options

proxy-only

Data retrieved from this remote cache will not be stored locally, but retrieved again on any subsequent request. By default, Squid will store objects it retrieves from other caches: by having the object available locally it can return the object fast, if it is requested again. This feature is often useful in a cluster of sibling caches to prevent each cache from holding every object. When the caches are close to each other (e.g. on the same ethernet segment), then it costs relatively little to transfer an object from one to the other. While this is good for latency, it can be a waste of bandwidth, especially if the other cache is on the same piece of ethernet. In the examples section of this chapter,we use this option when load-balancing between two cache servers.

Weight

If more than one cache server has an object (based on the result of an ICP query), Squid decides which cache to get the data from the cache that responded fastest. If you want to prefer one cache over another, you can add a weight value to the preferred cache's config line. Larger values are preferred. Squid times how long each ICP request takes (in milliseconds), and divides the time by the weight value, using the cache with

the smallest result. A higher weight will artificially lower the calculated RTT between peers, thereby favoring it in the selection algorithm. Your weight value should thus not be an unreasonable value.

ttl

An outgoing multicast packet has a ttl (Time To Live) value, which is used to ensure that loops are not created. Each time a packet passes through a router, the router decrements this ttl value, and the value is then checked. Once the value reaches zero, the packet is dropped. If you want multicast packets to stay on your local network, you would set the ttl value to 1. The first router to see the packet would decrement the packet, discover the ttl was zero and discard it. This value gives you a level of control on how many multicast routers will see the packet. You should set this value carefully, so that you limit packets to your local network or immediate multicast peers (larger multicast groups are seldom of any use: they generate too many responses, and when geographically dispersed, may simply add latency. You also don't want crackers picking up all your ICP requests by joining the appropriate multicast group.)

no- query

Squid will send ICP requests to all configured caches. The response time is measured, and used to decide which parent to send the HTTP request to. There is another function of these requests: if there is no response to a request, the cache is marked down. If you are communicating with a cache that does not support ICP, you must use the no-query option: if you don't, Squid will consider that cache down, and attempt to go directly to the destination server. (If you want, you can set the ICP port on the config line to point to the echo port, port 7. Squid will then use this port to check if the machine is available. Note that you will have to configure inetd.conf to support the UDP echo port.) This option is normally used in conjunction with the default option and round-robin option.

cache_peer proxy.visolve.com1 parent 3128 3130 no-query default

Default

This sets the host to be the proxy of last resort. If no other cache matches a rule (due to acl or domain filtering), this cache is used. If you have only one way of reaching the outside world, and it doesn't support ICP, you can use the default and no-query options to ensure that, all queries are passed through it. If this cache is then down, the client will see an error message (without these options, Squid would attempt to route around the problem.)

round-robin
This option must be used on more than one cache_peer line to be useful. Connections to caches configured with this options are spread evenly (round-robined) among the caches. This can be used by client caches to communicate with a group of loaded parents, so that load is spread evenly. If you have multiple Internet connections, with a parent cache on each side, you can use this option to do some basic load-balancing of the connections.

In other words,the round-robin option is similar to default, except that Squid forwards the request to the parent with the lowest use count. The cache_peer_domain restrictions still apply, of course. A typical configuration might look like:

restrictions still apply, of course. A typical configuration might look like:

```
cache_peer proxy.visolve.com1 parent 3128 3130 round-robin no-query
cache_peer proxy.visolve.com2 parent 3128 3130 round-robin no-query
cache_peer proxy.visolve.com3 parent 3128 3130 round-robin no-query
```

Squid treats all round-robin parents equally. It is not currently possible to, e.g., forward 25% of the requests to one parent and 75% to another.

no-net-dbexchange

If your cache was configured to keep ICMP (ping) timing information with the --enable-icmp configure option, your cache will attempt to retrieve the remote machine's ICMP timing information from any peers. If you don't want this to happen (or the remote cache doesn't support it), you can use the no-netdb-exchange option to stop Squid from requesting this information from the cache

no-delay

Hits from other caches will normally be included into a client's delay-pool information. If you have two caches load-balancing, you don't want the hits from the other cache to be limited. You may also want hits from caches in a nearby hierarchy to come down at full speed, not to be limited as if they were misses. Use the no-delay option to ensure that requests come down at their full speed.

login
Caches can be configured to use usernames and passwords on accesses. To authenticate with a parent cache, you can enter a username and password using this tag. Note that, the HTTP protocol makes authenticating to multiple cache servers impossible: you cannot chain together a string of proxies, each one requiring authentication. You should only use this option if this is a personal proxy.

Round Trip time

It is the time interval between the sending of the first byte of an HTTP request for the request, until the last bytes of the server response has arrived at the requesting web client.

Probe

Squid will wait for up to dead_peer_timeout seconds after sending out an ICP request before deciding to ignore a peer. With a multicast group, peers can leave and join at will, and it should make no difference to a client. This presents a problem for Squid: it can't wait for a number of seconds each time (whatif thecaches are on the same network, and responses come back in milliseconds: the waiting just adds latency.) Squid gets around this problem by sending ICP probes to the multicast address occasionally. Each host in the group responds to the probe, and Squid will know how many machines are currently in the group. When sending a real request, Squid will wait until it gets atleast as many responses as were returned in the last probe: if more arrive, great. If less arrive, though, Squid will wait until the dead_peer_timeout value is reached. If there is still no reply, Squid marks that peer as down, so that all connections

reached, if there is still no reply, Squid marks that peer as down, so that all connections are not held up by one peer.

What is the httpd-accelerator mode

An accelerator caches incoming requests for outgoing data (i.e., that which you publish to the world). It takes load away from your HTTP server and internal network. You move the server away from port 80 (or whatever your published port is), and substitute the accelerator, which then pulls the HTTP data from the ``real" HTTP server (only the accelerator needs to know where the real server is). The outside world sees no difference (apart from an increase in speed, with luck).

The httpd_accel_uses_host_header Option

The httpd_accel_uses_host_header option. A normal HTTP request consists of three values: the type of transfer (normally a GET, which is used for downloads); the path and filename to be retrieved (or executed, in the case of a cgi program); and the HTTP version.

This layout is fine if you only have one web site on a machine. On systems where you have more than one site, though, it makes life difficult: the request does not contain enough information, since it doesn't include information about the destination domain. Most operating systems allow you to have IP aliases, where you have more than one IP address per network card. By allocating one IP per hosted site, you could run one web server per IP address. Once the programs were made more efficient, one running program could act as a server for many sites: the only requirement was that you had one IP address per domain. Server programs would find out which of the IP addresses clients were connected to, and would serve data from different directories for each IP.

There are a limited number of IP addresses, and they are fast running out. Some systems also have a limited number of IP aliases, which means that you cannot host more than a (fairly arbitrary) number of web sites on machine. If the client were to pass the destination host name along with the path and filename, the web server could listen to only one IP address, and would find the right destination directores by looking in a simple hostname table.

From version 1.1 on, the HTTP standard supports a special Host header, which is passed along with every outgoing request. This header also makes transparent caching and acceleration easier: by pulling the host value out of the headers, Squid can translate a standard HTTP request to a cache-specific HTTP request, which can then be handled by the standard Squid code. Turning on the httpd_accel_uses_host_header option enables this translation. You will need to use this option when doing transparent caching.

It's important to note that acls are checked before this translation. You must combine this option with strict source-address checks, so you cannot use this option to accelerate multiple backend servers (this is certain to change in a later version of Squid).

The always_direct and never_direct tag

Squid checks all always_direct tags before it checks any never_direct tags. If a matching 'always_direct tag' is found, Squid will not check the never_direct tags, but decides which cache to talk to immediately. This behavior is demonstrated by the following example here, Squid will attempt to go the machine intranet, even though the same host is also matched by all acl.

Bypassing a parent for a local machine

```
cache_peer proxy.visolve.com parent 3128 3130
acl all src 0.0.0.0/0.0.0
acl localmachines dstdomain intranet.mydomain.example
never_direct allow all
always_direct allow localmachines
```

i.e.,
Let's consider a request destined for the web server intranet.mydomain.example. Squid first works through all the always_direct lines; the request is matched by the first (and only) line. The never_direct and always_direct tags are acl-operators, which means that the first match is considered. In this illustration, the matching line instructs Squid to go directly when the acl matches, so all neighboring peers are ignored for this request. If the line used the deny keyword instead of allow, Squid would have simply skipped on to checking of the former never_direct lines.

Now, suppose, a request arrives for an external host. Squid works through the always_direct lines, and finds that none of them match. The never_direct lines are then checked. The all acl matches the connection, so Squid marks the connection as never to be forwarded directly to the origin server.

Access.log details

The native access.log has ten (10) fields.There is one entry here for each HTTP (client) request and each ICP Query. HTTP requests are logged when the client socket is closed. A singledash (-) indicates unavailable data.

1. Timestamp
The time when the client socket is closed. The format is 'Unix time' (seconds since Jan 1, 1970) with millisecond resolution. This can be modified to visible format by 'cat access.log | perl -nwe 's/^(\d+)/localtime($1)/e; print';.

2. Elapsed Time
The elapsed time of the request, in milliseconds. This is time between the accept() and close() of the client socket.

3. Client Address
The IP address of the connecting client, or the FQDN if the 'log_fqdn' option is enabled in the config file.

4. Log Tag / HTTP Code
The Log Tag describes how the request was treated locally (hit, miss, etc). All the tags are described below. The HTTP code is the reply code taken from the first line of the

are described below. The HTTP code is the reply code taken from the first line of the HTTP reply header. Non-HTTP requests may have zero reply codes.

5. Size
The number of bytes written to the client.

6. Request Method
The HTTP request method, or ICP_QUERY for ICP requests.

7. URL
The requested URL.

8. Ident
If ident_lookup is on, this field may contain the username associated with the client connection as derived from the ident service.

9. Hierarchy Data / Hostname
A description of how and where the requested object was fetched.

10. Content Type
The Content-type field from the HTTP reply.

Access Log Tag / HTTP Code

"TCP_" refers to requests on the HTTP port.

TCP_ HIT
A valid copy of the requested object was in the cache.

TCP_MISS
The requested object was not in the cache.

TCP_REFRESH_HIT
The object was in the cache, but STALE. An If-Modified-Since request was made and a '304 Not Modified' reply was received.

TCP_REF_FAIL_HIT
The object was in the cache, but STALE. The request to validate the object failed, so the old (stale) object was returned.

TCP_REFRESH_MISS
The object was in the cache, but STALE. An If-Modified-Since request was made and the reply contained new content.

TCP_CLIENT_REFRESH
The client issued a request with the 'no-cache' pragma.

TCP_CLIENT_REFRESH_MISS
The client issued a "no-cache" pragma, or some analogous cache control command along with the request. Thus, the cache has to refetch the object from origin server. It is users pushing that reload-button forcing the proxy to check for a new copy (also triggered by selecting a bookmark in some browser versions). In short, the browser

triggered by selecting a bookmark in some browser versions). In short, the browser forced the proxy to check for a new version

TCP_IMS_HIT
The client issued an If-Modified-Since request and the object was in the cache and still fresh. TCP_HIT and TCP_IMS_HIT are hits, the only difference is that in the TCP_IMS_HIT case, the browser already had an up to date version, so there was no need to send the Squid cached copy to the requestor.

TCP_IMS_MISS
The client issued an If-Modified-Since request for a stale object.

TCP_SWAPFAIL
The object was believed to be in the cache, but could not be accessed.

TCP_DENIED
Access was denied for this request

"UDP_" refers to requests on the ICP port

UDP_HIT
A valid copy of the requested object was in the cache.

UDP_HIT_OBJ
Same as UDP_HIT, but the object data was small enough to be sent in the UDP reply packet. Saves the following TCP request.

UDP_MISS
The requested object was not in the cache.

UDP_DENIED
Access was denied for this request.

UDP_INVALID
An invalid request was received.

UDP_RELOADING
The ICP request was "refused" because the cache is busy, reloading its metadata.

SIBLING_HIT
The object was fetched from a sibling cache which replied with UDP_HIT.

PARENT_HIT
The object was requested from a parent cache which replied with UDP_HIT.

DEFAULT_PARENT
No ICP queries were sent. This parent was chosen because it was marked ``default'' in the config file.

FIRST_UP_PARENT
The object was fetched from the first parent in the list of parents.

NO_PARENT_DIRECT
The object was fetched from the origin server, because no parents existed for the given URL

URL.

FIRST_PARENT_MISS
The object was fetched from the parent with the fastest (possibly weighted) round trip time.

CLOSEST_PARENT_MISS
This parent was chosen, because it included the the lowest RTT measurement to the origin server. See also the closests-only peer configuration option.

CLOSEST_PARENT
The Parent selection was based on our own RTT measurements.

Refresh Pattern

Squid switched from a Time-To-Live based expiration model to a Refresh-Rate model. Objects are no longer purged from the cache when they expire. Instead of assigning TTL's when the object enters the cache, we now check freshness requirements when objects are requested. If an object is 'fresh' it is given directly to the client. If it is 'stale' then we make an If-Modified-Since request for it.

Terms in delay pool

Pool:
A collection of bucket groups as appropriate to a given class.

bucket Pool:
a group of buckets within a pool, such as the per-host bucket group, the per-network bucket group or the aggregate bucket group (the aggregate bucket group is actually a single bucket).

bucket:
an individual delay bucket represents a traffic allocation, which is replenished at a given rate (up to a given limit) and causes traffic to be delayed when empty.

Classes:
There are 3 classes of delay pools - class 1 is a single aggregate bucket, class 2 is an aggregate bucket with an individual bucket for each host in the class C, and class 3 is an aggregate bucket, with a network bucket (for each class B) and an individual bucket for each host.

class:-
Class of a delay pool determines how the delay is applied, ie, whether the different client IPs are treated separately or as a group (or both).

class1:-
Class 1 delay pool contains a single unified bucket, which is used for all requests from hosts subject to the pool.

class2:-

class2:-
Class 2 delay pool contains one unified bucket and 255 buckets,one for each host on an 8-bit network

class3:-
It contains 255 buckets for the subnets in a 16-bit network, and individual buckets for every host on these networks (IPv4 class B)

Setting the parameters for each pool is done by :

delay_parameters pool aggregate network individual.

The variables here are:

where pool is a pool number , i.e., a number between 1 and the number specified in delay_pools as used in delay_class lines, aggregate is the parameter for the aggregate bucket, network for the network bucket, and individual for the individual bucket. Aggregate is only useful for classes 1, 2 and 3, individual for classes 2 and 3, and network for class 3. Each of these parameters is specified as restore / maximum - restore being the bytes per second restored to the bucket, and maximum being the amount of bytes that can be in the bucket at any time. It is important to remember that they are in bytes per second, not bits. To specify that a parameter is unlimited, use a -1.

If we wish to limit any parameter in bits per second, divide this amount by 8, and use the value for both the restore and the maximum. For example, to restrict the entire proxy to 64kbps, use:

delay_parameters 1 8000/8000

Ftp Login Information

Squid can act as a proxy server for various Internet protocols. The most commonly used protocol is HTTP, but the File Transfer Protocol (FTP) is still alive and well.

FTP was written for authenticated file transfer (it requires a username and password). To provide public access, a special account is created: the anonymous user. When you log into an FTP server you use this as your username. As a password, you generally use your email address. Most browsers these days automatically enter a useless email address.

It's polite to give an address that works, though. If one of your users abuses a site, it allows the site admin get hold of you easily.

Squid allows you to set the email address that is used with the ftp_user tag. You should probably create a squid@yourdomain.example email address specifically for people to contact you on.

There is another reason to enter a proper address here: some servers require a real email address. For your proxy to log into these ftp servers, you will have to enter a real email address here.

Effective User and Group ID

Squid can only bind to low numbered ports (such as port 80) if it is started as root. Squid is normally started by your system's rc scripts when the machine boots. Since these scripts run as root, Squid is started as root at bootup time.

Once Squid has been started, however, there is no need to run it as root. Good security practice is to run programs as root only when it's absolutely necessary, and for this reason Squid changes user and group ID's once it has bound to the incoming network port.

The cache_effective_user and cache_effective_group tags tell Squid what ID's to change to. The Unix security system would be useless if it allowed all users to change their ID's will, so Squid only attempts to change ID's if the main program is started as root.

If you do not have root access to the machine, and are thus not starting Squid as root, you can simply leave this option commented out. Squid will then run with whatever user ID starts the actual Squid binary.

Now let us assume that, you have created both a squid user and a squid group on your cache machine. The above tags should thus both be set to 'squid' .

Timeouts

Half closed clients:
The clients that shutsdown the sending side of their TCP connections, while leaving their receiving sides open, we term it as halfclosed clients. ie., the clients closes while the handshaking is in progress.

Fully closed clients:
The clients and servers have shared their acknowledgements(request and responses) before closing.

Persistent Connection:
Persistent Connection (keep alive) feature allows the same Connection to remain open for multiple requests. Obviously the drawback is that, the next request processing cannot start before the previous response has been sent by the server.

IDENT:
Squid will make an RFC931/ident request for client connections if 'ident_lookup' is enabled in the config file. Currently, the ident value is only logged with the request in the access.log. It is not currently possible to use the ident return value for access control purposes.

URN:
The URI architecture requires that a resource be named by a URN (Uniform Resource Name) and be retrieved by a URL (Uniform Resource Locator). A URC (Uniform Resource Characterstitic) binds the URN of a resource to one or more URLs. Once this system is activated, URNs will be used to "reference" information resources World Wide

system is activated, URNs will be used to "reference" information resources. World Wide Web clients will then send the URN for a desired resource to an international network of URN to URL resolvers (the URC service) that will return to the client one or more URLs that can be used to access the resource.

SIGHUP or SIGTERM:
The system signal sent to processes running in linux OS to shutdown.

External Programs

Htpasswd:
It is apache type passwd ,You can use this to create passwd for squid also.

The Syntax is:

htpasswd [-c] passwdfile username .

Redirector:
Squid now has the ability to rewrite requested URLs. Implemented as an external process (similar to a dns server), Squid can be configured to pass every incoming URL through a 'redirector' process that returns either a new URL, or a blank line to indicate no change.

The redirector program is NOT a standard part of the Squid package. However there are a couple of user-contributed redirectors in the "contrib/" directory. Since everyone has different needs, it is up to the individual administrators to write their own implementation. For testing, and a place to start, this very simple Perl script can be used:

```
#!/usr/local/bin/perl
$|=1;
print while (<>);
```

The redirector program must read URLs (one per line) on standard input, and write rewritten URLs or blank lines on standard output. Note that the redirector program can not use buffered I/O.

Ftp Passive Connections:
Ftp uses two data streams, one for passing commands around, the other for moving data. The command channel is handled by the ftpd listening on port 21.

The data channel varies depending on whether you ask for passive ftp or not. When you request data in a non-passive environment, you client tells the server ``I am listening on .'' The server then connects FROM port 20 to the ip address and port specified by your client. This requires your "security device" to permit any host outside from port 20 to any host inside on any port >1023. Somewhat of a hole.

In passive mode, when you request a data transfer, the server tells the client ``I am listening on .'' Your client then connects to the server on that IP and port and data flows.

Unlinkd Program:

Unlinkd is an external process used for unlinking old files in the cache to make room for newer object.

Pinger Process:
Squid ping program is an external program that provides Squid with icmp RTT information so that, it can more effectively choose between multiple remote parent caches for request fulfillment. There are special cases when this option is required, and your Squid must have been compiled with the --enable-icmp configure option for it to work. This option should only be used on caches with multiple parent caches on different networks that it must choose between. The default program to use for this task is called pinger. This option configures the pinger_program directive.

BYTES-hit ratio

The byte-hit ratio measures the ratio of total bytes from cached objects over the total bytes of objects requested.

Created By: squid@visolve.com Date: May 26,2002
Revision No:0.0
Modified By Date