

Table of Contents

Using a squid version other than the frontier-squid distribution.....	1
Enabling monitoring.....	3
Double checking.....	4

Using a squid version other than the frontier-squid distribution

Squid has a lot of parameters that are set in a file called squid.conf. These should be tuned according to the application. What we are describing below are changes to the default parameters that are needed for the Frontier application. The normal application for squid is to cache internet web pages, which is a bit different than Frontier. For web pages there are lots and lots of rather small objects. In the Frontier application there are relatively few objects, but some of them are much bigger.

NOTE: WE STRONGLY DISCOURAGE attempting to use any version of squid-3.X prior to squid-3.5.28 (including the default squid on EL6 and EL7) as it was missing crucial features (such as [collapsed_forwarding](#), at least prior to 3.5 and even then it had such significant problems that we had to hire a consultant to fix them, see below) and bug fixes (such as [squid bug #7](#)) essential to Frontier. If your squid is feeding other squids then definitely use our distribution because all upstream squid versions still have a problem with the default ufs cache related to [squid bug #7](#) and we have included a simple patch for it. We also discourage using 4.X versions prior to 4.4 because there has been no significant testing of it on the grid. Also, do not use rock cache because it does not work properly with If-Modified-Since.

If you want to use squid-2, all versions of squid before September 2009 have a [bug](#) that affects Frontier performance, so you should use squid version squid-2.7.STABLE7 or later. If your squid-2 might be feeding other squids then we especially recommend using our version as we have patches (for [squid bug #2831](#) and [squid bug #2833](#)) that were never released by the squid project.

Squid-3 & Squid-4 Gory Details: Squid-3 is a complete rewrite of squid in C++ instead of C so it took a very long time to catch up with all the features and stability of squid-2. Fortunately squid-4 is much more similar to squid-3, with only the addition of C++11 as a requirement. [Collapsed_forwarding](#) was added to squid-3.5, but it only worked for the initial loading of cached items, not for revalidating or reloading anything already cached but expired, and when not using rock cache it caused [deadlocks](#) with multiple workers. In any case Frontier cannot use rock cache in squid-3 because it is still susceptible to [squid bug #7](#) (the problem is more succinctly stated in [squid bug #4324](#)). That problem with rock cache was resolved in squid-4.2, but then an additional problem documented in [squid bug #4890](#) was discovered with rock cache + [collapsed forwarding](#) + If-Modified-Since and is not yet fixed, so we still cannot use rock cache with squid-4. A fix for [squid bug #2831](#) is in squid-3.5.16. squid-3.5.21 included a fix for [squid bug #4428](#) which Frontier needs. We hired a consultant to fix bugs [#2833](#), [#4311](#), and [#4471](#) which describe problems with collapsed forwarding. The patches for those bugs were included in squid-3.5.22, but squid-3.5.23 broke the fix for [bug #2833](#); it was fixed again in squid-3.5.27. Also, although [squid bug #7](#) was mostly fixed in squid-3.5, a small subset of the problem showed up again and we have a patch for that in our version which is not yet in any upstream squid version. Our version has a workaround for [bug #4575](#) which caused many wasted DNS lookups and which has not yet been fixed in any upstream squid version, although the workaround can be applied by configuration (instructions below). Frontier-squid-3 had patches for [#3952](#) and [#4616](#) that didn't get into any upstream squid-3.x version, but they did get into squid-4.3. A fix for [#4767](#) which was a problem with IPv6 on multiple workers was included in frontier-squid-3.5.27 but it was fixed in upstream 3.5.28 and 4.2.

So for the Frontier application:

1) If there's any chance that you will need to support Frontier clients that are older than the December 2010 release (version 2.8.0), essential changes for squid-2.7 or later are:

```
hierarchy stoplist cgi-bin
refresh_pattern -i /cgi-bin/ 0 0% 0
```

These two changes allow queries with ? in the URL to be cached. By default squid does not cache the dynamic web pages usually expected from a question mark, and the older Frontier clients used the question

MyOwnSquid < Frontier < TWiki

mark in all URLs. Without these two changes, squid won't cache anything for Frontier clients older than December 2010. (The default values of these two lines are:

```
hierarchy stoplist cgi-bin ?
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
```

so it's not much of a change.) squid-3 and later do not have the hierarchy stoplist, so you can skip that.

For squid-2.6 or earlier, the "hierarchy stoplist" line still needs to be changed, but instead of the

```
refresh_pattern -i /cgi-bin/ 0 0% 0
```

line there is a line:

```
acl QUERY urlpath_regex cgi-bin \?
```

that needs to be changed to:

```
acl QUERY urlpath_regex cgi-bin
```

2) Hardware dependent changes

```
cache_mem
cache_dir
```

You have to define a parameter `cache_dir` which tells the squid where to keep the information on disk and how large it should be. This should be at least 20000 megabytes but probably no more than 70% of the partition size (to allow room for log files and other uses). The other hardware parameter is `cache_mem`. The default `cache_mem` is only 8 MB and is probably a very old default and should be increased some, but we have found that squid performs better for large objects out of the disk cache than the memory cache. We recommend at most 1/8 of the physical RAM and no more than 128 MB, leaving a lot of memory for disk buffering.

3) Tuning changes

```
maximum_object_size 1048576 KB
maximum_object_size_in_memory 128 KB
```

The first item lets us cache objects up to 1 GB in size (which is a lot more than most web pages - the default is only 4 MB). To date, we have often cached tarballs up to 300 MB in size, and more might be needed. The second item lets us use the `cache_mem` we gave it above (the default is only 8 KB).

In order to avoid not caching any objects that squid first reads when they're less than 60 seconds from expiring (as described in squid bug #4531 [↗](#)) set this parameter:

```
minimum_expiry_time 0
```

An important option for squid-2.6 and later is:

```
collapsed_forwarding on
```

This option combines requests aggressively so that a file is retrieved only once from the origin server. This is a very good idea for computer farms so make sure it is on.

For version squid-3.2 and later, in order to enable frontier-client versions older than 2.8.21 (which is not yet released as of July 2018) to clear certain server errors from the cache, it is important to include the following

config parameters as described in squid bug #4809:

```
acl PragmaNoCache req_header Pragma no-cache
send_hit deny PragmaNoCache
```

To avoid many DNS lookups as described in squid bug #4575, set the following:

```
url_rewrite_extras XXX
store_id_extras XXX
```

Finally, if your squid is a squid-2 version and might possibly feed other squids then set this:

```
ignore_ims_on_miss on
```

The default for that option prevents caching when an upstream squid sends an If-Modified-Since request and the object isn't already cached. This is fixed in squid-3 and is not an option.

4) Recommended log file changes

```
strip_query_terms off
cache_store_log none
logformat awstats %>a %ui %un [%d/%b/%Y:%H:%M:%S]t1.%03tu %z] "%rm %ru HTTP/%rv" %Hs %st
```

and then use "awstats" as the second parameter to the "access_log" option, for example:

```
access_log /var/log/squid/access.log awstats
```

Since the log files can get very big very fast, we run a cron job 4 times an hour that checks to see if access.log is greater than a chosen size and if so does an extra rotation, in addition to nightly rotations.

Enabling monitoring

That's basically it, except for one important thing. For Frontier we monitor our squids remotely using SNMP. This is not turned on by default in squid, so to use it you have to turn it on at compilation time. If you want to use SNMP there are a few more settings needed in the squid.conf, especially the ACL access for whatever machines are allowed to read the SNMP information. Scientific Linux/Redhat Enterprise Linux squid rpms should already have SNMP enabled at compilation time. Therefore, it should be possible to enable monitoring by adding something like the following to your squid.conf which allow requests from the main CERN network and the CERN Hungary data center:

```
acl HOST_MONITOR src 127.0.0.1/32 128.142.0.0/16 188.184.128.0/17 188.185.128.0/17
acl snmppublic snmp_community public

snmp_access allow snmppublic HOST_MONITOR
snmp_access deny all
```

all in the appropriate places in squid.conf.

If you are using a version of squid-3 you will also need to set

```
snmp_port 3401
```

because it is not on by default in squid-3.

You may also need to open firewall and/or iptables holes for the addresses on the HOST_MONITOR line above. If your firewall administrators don't like opening the whole IP address ranges please look at the instructions in the frontier-squid install documentation for justification and an alternative. In any case leave

the squid.conf acl more open so only one place needs to change when the monitoring machine IP addresses change.

Double checking

One thing you can do is make a dummy installation of our tarball. It can be installed anywhere by any user. Then do a diff of your squid.conf with our squid.conf. For startup and shutdown procedures, you are on your own (unless of course you are using someone else's package).

Responsible: DaveDykstra

This topic: Frontier > MyOwnSquid

Topic revision: r45 - 2019-05-14 - DaveDykstra



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback