# 6. Файлы журналов Squid

Журналы - ценный источник информации о нагрузке на ваш Squid и его производительности. Журналируются записи не только с информацией о доступе, но также и о системных ошибках и потребляемых ресурсах (память, дисковое пространство). Squid поддерживает несколько типов файлов журналов. Некоторые из них требуют активации на этапе процесса компиляции, другие могут быть безопасно отключены прямо во время работы.

Есть несколько основополагающих компонет для всех лог-файлов. Временные метки обычно записываются в файлы журналов в формате UTC в секундах, если не указано другого. Временная метка обычно содержит расширение в милисекундах.

Частые запросы временной метки на загруженных кешах могут сказываться на производительности системы. Если при компиляции указать ключ *--enable-time-hack*, то Squid будет запрашивать новое значение времени раз в секунду. При этом используются функциональные возможности Unix-вого *alarm()*. Имейте ввиду, что журналируемое время в этом случае не будет столь точным и этого может оказаться недостаточно для некоторых программ, анализирующих содержимое логов. Обычно нет никакой необходимости играть в форматом временной метки.

## 6.1 *squid.out*

Если вы запускаете ваш Squid из скрипта *RunCache*, то файл *squid.out* содержит время старта Squid, а также все сообщения о фатальных ошибках, генерируемых неудачным вызовом *assert()*. Если вы не используете *RunCache*, вы не увидите подобного файла.

## 6.2 *cache.log*

Файл *cache.log* содержит отладочную информацию и сообщения о ошибках, которые генерирует Squid. Если для запуска Squid вы используете скрипт *RunCache* или запускаете его с ключем *-s*, то копия некоторых сообщений попадет в ваш syslog. Использлвание отдельного файла для хранения лога Squid - вопрос личных предпочтений.

Для автоматических анализаторов логов файлу *cache.log* особо предложить нечего. Вы обычно будете обращаться к этому файлу для отчета о ошибках при программировании Squid, тестировании новых возможностей или поиске причин непонятного поведения прокси и т.п..

## 6.3 *useragent.log*

Этот лог-файл будет вестить, если:

1. вы указали при сборке опцию *--enable-useragent-log* и
2. вы указали в какой файл это попадет при помощи опции *useragent_log*.

В лог-файле user agent вы можете найти информацию о броузерах ваших клиентов. Использование этой опции вкупе с загруженным сквидом - не самая лучшая из идей.

## 6.4 *store.log*

В файле *store.log*отмечаются объекты, который в данный момент сохзранены на диск или удалены с диска. Подобнае журналирование операций обычно используется в отладочных целях. Точное подверждение того, находиться ли объект на вашем диске возможно только после *полного* анализа файла журнала. Удаление объекта может быть занечено в лог позже, чем сохранение на диск.

Файл *store.log* может быть интересен для анализа данных по объектам, хранящимся на вашем диске, времени их хранения или информации о том, сколько раз обращались к тому или иному "горячему" объекту. The latter may be covered by another log file, too. With knowledge of the *cache_dir* configuration option, this log file allows for a URL to filename mapping without recursing your cache disks. Однако, разработчики Squid рекомендуют все же рассматривать *store.log* как файл с отладочной инормацией и вам желательно этому следовать. пока вы не будете точно уверены в том, что вы делаете.

Формат строки, которая заносится в store.log состоит из одиннадцати полей разделенных пробелами в соответствии с функцией *storeLog()* файла *src/store_log.c*:

```
"%9d.%03d %-7s %08X %4d %9d %9d %9d %s %d/%d %s %s\n"
```

**time**

время, когда запись попала в лог, в UTC с миллисекундами.

**action**

Действие, произведенное над объектом, в соответствии с *src/store_log.c*:

- **CREATE** не используется.
- **RELEASE** объект был удален из кеша (см. также номер файла).
- **SWAPOUT** объект был сохранен на диск.
- **SWAPIN** объект есть на диске и прочитан в память.

**file number**

Номер файла для размещенного в нем объекта. Обратите внимание, что путь к этому файлу определяется в соотвестствии со значением указанной вами *cache_dir*.

Номер файла *FFFFFFFF* указывает на объект, находящийся в памяти. Любой результирующий код для подобного номера файла относиться к объекту, существующему только в памяти, а не на диске. К примеру, если код *RELEASE* был записан в лог с номером файла *FFFFFFFF*, то объект существовал только в памяти и был из нее удален.

**status**

код статуса HTTP-ответа.

**datehdr**

значение заголовка HTTP "Date: " ответа.

**lastmod**

значение заголовка HTTP "Last-Modified: " ответа.

**expires**

значение заголовка HTTP "Expires: " ответа.

**type**

значение HTTP "Content-Type" либо "unknown", если не может быть определено.

**sizes**

Этот столбец состоит из двух полей разделенных слешем:

1. Заявленная длина содержимого, взятая из заголовка HTTP "Content-Length: " ответа.
2. Фактически прочитаный размер.

Если заявленная (или ожидаемая) длина неопределена, то она будет установлена в нуль. Если заявлена ненулевая длина, но она не равна реальному размеру, то объект будет удален из кеша.

**method**

Метод запроса объекта, к примеру *GET*.

**key**

Ключ объекта, обычно это URL.

Формат временной метки для стобцов Date и Expires - секунды с начала UTC. Фактические значения выбираются из HTTP-заголовка ответа. заголовки, которые не удается отпарсить помечаются значением -1, а отсутствующие заголовки - значением -2.

Столбец key - обычно просто URL объекта. Some objects though will never become public. Таким образом ключ указывает включить уникальное целое число и метод запроса в дополнение к URL.

## 6.5 *hierarchy.log*

Этот лог-файл используется только в версии Squid-1.0. Его формат

```
[date] URL peerstatus peerhost
```

## 6.6 *access.log*

Работа большинства программ, анализирующих файл журналов, основана на сожержимом *access.log*. В настоящее время существуют два возможных формата этого файла журнала в зависимости от значения

конфигурационной опции *emulate_httpd_log*. По умолчанию, Squid журналирует данные в собственном формате. Если указанная выше опция включена, то Squid будет использовать общий формат файла журнала, который использует web-демон CERN.

Общий формат файла журнала содержит меньше информации, чем собственный формат и она немного различна. Собственный формат содержит больше интересной для администратора информации о работе кеша.

### *Общий формат файла журанала*

Общий формат файла журнала используется большинством HTTP-серверов. Этот формат включает такие семь полей:

```
remotehost rfc931 authuser [date] "method URL" status bytes
```

Обрабатывается большим кол-вом утилит. Общий формат содержит информацию, отличную от собственного формата файла журнала. Версия HTTP записывается в журнал, чего не происходит при собственном формате журнала.

### *Собственный формат файла журнала*

Собственный формат различен для разных версий Squid. Для Squid-1.0 это:

```
time elapsed remotehost code/status/peerstatus bytes method URL
```

А в For Squid-1.1 информация, которая содержалась в *hierarchy.log* была перенесена в *access.log*. Его формат:

```
time elapsed remotehost code/status bytes method URL rfc931 peerstatus/peerhost type
```

Для Squid-2 столбцы остались теже, но их содержимое немного отличается.

Собственный формат файл сожержит больше информации, которая отличается от об общего формата файла журнала: продолжительность запроса, некоторая информация о таймаутах, адрес следующего сервера верхнего уровня и тип содержимого.

Существуют инструменты, преобразующие один формат файла в другой. Имейте ввиду, что хотя и оба формата содержат общую информацию, но также оба из них содержат такую информацию, которая не является частью другого и эта часть ифномрации будет утеряна в процессе конвертации. Особенно стоит отметить, что конвертирование из одного форматат в другой и обрратно вообще не возможно без подобных потерь.

Утилита *squid2common.pl* преобразовывает лобой из форматов файлов журнала squid к старому стилю запими для прокси CERN. Также существуют инструменты для анализа, оценки и отображения результатов анализа этого формата.

### *Детальное описание собственного формата access.log*

Рекомендуется использовать собственный формата файла журнала Squid изза большего количества предоставляемой информации, которая может быть проанализирована впоследствии. Формат строки вывода для собственного *access.log* выглядит примерно так:

```
"%9d.%03d %6d %s %s/%03d %d %s %s %s %s%s/%s %s"
```

Поэтому запись в *access.log* обычно состоит по крайней мере из 10 полей, разделенныцм одним или более пробелами:

**time**

Временная метка Unix в секунда с начала UTC с миллисекундым разрешением. Вы можете конверитровать временную метку Unix во что-нибудь более читабельное, использовав такой простой скрипт на Perl:

```
#! /usr/bin/perl -p
s/^\d+\.\d+/localtime $&/e;
```

**duration**

Общее время, которое показываетс сколько миллисекунд у кеша заняла обработка транзакции. Есть различия в интерпритации между TCP и UDP:

- Для HTTP/1.0 это в общем случае время между *accept()* и *close()*.
- For persistent connections, this ought to be the time between scheduling the reply and finishing sending it.
- Для ICP - этовремя между планирумым , this is the time between scheduling a reply и действительной его отправкой.

Имейте ввиду, что запись заносится в журнал *после* того, как ответ был полностью отправлен получателю, а *не* во время обработтки транзакции.

**client address**

IP-адрес of the requesting instance, клиентский IP-адрecs. Опция конфигурации *client_netmask* может позволять скрыть ифнормацию о клиенте, по причине защиты данных, однако это сильно затруднит анализ. Often it is better to use one of the log file anonymizers.

Also, the *log_fqdn* configuration option may log the fully qualified domain name of the client instead of the dotted quad. The use of that option is discouraged due to its performance impact.

**result codes**

Эта колонка состоит из двух значений, которые разделены слешем. This column encodes the transaction result:

1. The cache result of the request contains information on the kind of request, how it was satisfied, or in what way it failed. Обратитесь к разделу результирующие коды Squid, для пояснений по результирующим кодам.

   Several codes from older versions are no longer available, were renamed, or split. Especially the *ERR_* codes do not seem to appear in the log file any more. Also refer to section Squid result codes for details on the codes no longer available in Squid-2.

   The NOVM versions and Squid-2 also rely on the Unix buffer cache, thus you will see less *TCP_MEM_HIT*s than with a Squid-1. Basically, the NOVM feature relies on *read()* to obtain an

object, but due to the kernel buffer cache, no disk activity is needed. Only small objects (below 8KByte) are kept in Squid's part of main memory.

2. The status part contains the HTTP result codes with some Squid specific extensions. Squid uses a subset of the RFC defined error codes for HTTP. Refer to section status codes for details of the status codes recognized by a Squid-2.

**bytes**

Размер кол-ва доставленных клиенту данных. Имейте ввиду, что это не "чистый" размер объекта, заголовки тоже учитываются. Также в результате неудачного запроса возвращается страница с сообщением о ошибке, размер которой записывается в данной колонке.

**request method**

Метод запроса на получение объекта. Обратитесь к разделу методы запроса, чтобы получить информацию о доступных методах. Если вы установите в off директиву *log_icp_queries* в вашем файле конфигурации, то вы не увидите (а значит и не сможете проанализировать) данных по ICP-обмену. Метод *PURGE* доступен только в случае, если у вас разрешен ACL для ``method purge'' в вашем конфигурационном файле.

**URL**

Это поле содержит запрошенный URL. Please note that the log file may contain whitespaces for the URI. The default configuration for *uri_whitespace* denies whitespaces, though.

**rfc931**

В восьмой колонке может содержаться результат запроса ident для клиента, пославшего звпрос. Т.к. ident-запросы влияют на производительность, то по умолчанию *ident_loookups* выключены. В этом случае или если данная ифнормация недоступна, то в лог будет записан символ ``-''.

**hierarchy code**

The hierarchy information consists of three items:

1. Any hierarchy tag may be prefixed with *TIMEOUT_*, if the timeout occurs waiting for all ICP replies to return from the neighbours. The timeout is either dynamic, if the *icp_query_timeout* was not set, or the time configured there has run up.
2. A code that explains how the request was handled, e.g. by forwarding it to a peer, or going straight to the source. Refer to section hier-codes for details on hierarchy codes and removed hierarchy codes.
3. The name of the host the object was requested from. This host may be the origin site, a parent or any other peer. Also note that the hostname may be numerical.

**type**

Тип содержимого (content type) объекта, извлеченный из заголовка HTTP-ответа. Имейте ввиду, что для ICP обмена обычно никакого типа содержимого не указывается, поэтому в журнал в этом случае подадет ``-''. Кроме того в некоторых тип сожержимого указывается как ``:'' или не указывается вовсе.

There may be two more columns in the *access.log*, if the (debug) option *log_mime_headers* is enabled In this case, the HTTP request headers are logged between a ``[" and a ``]", and the HTTP reply headers are also logged between ``[" and ``]". All control characters like CR and LF are URL-escaped, but spaces are *not* escaped! Parsers should watch out for this.

## 6.7 Результирующие коды Squid

Коды **TCP_** соотвествуют запросам на HTTP-порту (обычно 3128). Коды **UDP_** соотвествуют запросам на ICP-порту (обычно 3130). Если журналирование ICP было выключено при помощи опции *log_icp_queries*, то ICP-отыетф попадать в журнал не будут.

Нижеследующие результирующие коды были взяты из Squid-2, в соответствии со структурой *log_tags* в файле *src/access_log.c*:

**TCP_HIT**

Верная копия запрошенного объекта была в кеше.

**TCP_MISS**

Запрошенного объекта не было в кеше.

**TCP_REFRESH_HIT**

Запрошенный объект был закеширован, но *УСТАРЕЛ*. IMS-запрос для этого объекта вернул "304 not modified".

**TCP_REF_FAIL_HIT**

Запрошенный объект был закеширован, но *УСТАРЕЛ*. IMS-запрос завершен неудачно и устаревший объект был доставлен.

**TCP_REFRESH_MISS**

Запрошенный объект был закеширован, но *УСТАРЕЛ*. IMS-запрос вернул новое содержимое.

**TCP_CLIENT_REFRESH_MISS**

Клиент послал прагму "no-cache" или другу аналогичную команду контроля кеширования в запросе. Поэтому кеш должен повторно получить объект.

**TCP_IMS_HIT**

Клиент использовал IMS-запрос для объекта, который был найден в кеше свежим.

**TCP_SWAPFAIL_MISS**

Объект скорее всего был в кеше, но доступа к нему нет.

**TCP_NEGATIVE_HIT**

Запрос для негативно кешированных объектов типа "404 not found", о которых кеш знает, что они

недоступны. См. пояснения по *negative_ttl* в вашем файле *squid.conf*.

**TCP_MEM_HIT**

Верная копия запрошенного объекта была в кеше *и* в памяти, доступа к диску не производилось.

**TCP_DENIED**

Доступ запрещен для этого запроса.

**TCP_OFFLINE_HIT**

Запрошенный объект был извлечен из кеша в режиме offline. В режиме offline никогда не проверяются, см. *offline_mode* в файле *squid.conf*.

**UDP_HIT**

Верная копия запрошенного объекта была в кеше.

**UDP_MISS**

Запрошенный объект отсутствует в этом кеше.

**UDP_DENIED**

Доступ запрещен для этого запроса.

**UDP_INVALID**

Был получен неверный запрос.

**UDP_MISS_NOFETCH**

Из-за опции запуска "-Y" или частых отказов, кеш при хите будет возвращать либо UDP_HIT или этот код. Соседи таким образом получат только хиты.

**NONE**

Указывается с ошибками и запросами cachemgr.

Следующие коды больше недоступны в Squid-2:

**ERR_***

Ошибки теперь указываются в статусе кода.

**TCP_CLIENT_REFRESH**

См. TCP_CLIENT_REFRESH_MISS.

**TCP_SWAPFAIL**

См. TCP_SWAPFAIL_MISS.

**TCP_IMS_MISS**

Удалено, вместо этого используется TCP_IMS_HIT.

**UDP_HIT_OBJ**

Совпавший объект больше недоступен.

**UDP_RELOADING**

См. UDP_MISS_NOFETCH.

# 6.8 HTTP status codes

These are taken from RFC 2616 and verified for Squid. Squid-2 uses almost all codes except 307 (Temporary Redirect), 416 (Request Range Not Satisfiable), and 417 (Expectation Failed). Extra codes include 0 for a result code being unavailable, and 600 to signal an invalid header, a proxy error. Also, some definitions were added as for RFC 2518 (WebDAV). Yes, there are really two entries for status code 424, compare with *http_status* in *src/enums.h*:

```
 000 Используется в основном для UDP-трафика.

 100 Continue
 101 Переключение протоколов
*102 Processing

 200 OK
 201 Создано
 202 Принято
 203 Non-Authoritative Information
 204 Нет содержимого
 205 Содержимое отвергнуто
 206 Partial Content
*207 Мультистатус

 300 Множественный выбор
 301 Moved Permanently
 302 Удаленно временно
 303 See Other
 304 Не изменено
 305 Используется прокси
[307 Временное перенарпавление]

 400 Неверный запрос
 401 Unauthorized
 402 Payment Required
 403 Запрещено
 404 Не найдено
 405 Метод не разрешен
 406 Not Acceptable
 407 Прокси требует аутентификации
 408 Таймаут запроса
 409 Конфликт
 410 Gone
 411 Требуется длина
 412 Precondition Failed
```

```
 413 Request Entity Too Large
 414 Запрошенный URI слишком велик
 415 Неподдерживаемый тип Media
[416 Request Range Not Satisfiable]
[417 Expectation Failed]
*424 Заблокировано
*424 Failed Dependency
*433 Unprocessable Entity

 500 Внутренняя ошибка сервера
 501 Not Implemented
 502 Неверный шлюз
 503 Сервис недоступен
 504 Таймаут шлюза
 505 Версия HTTP не поддерживается
*507 Insufficient Storage

 600 ошибка разбора заголовка Squid
```

# 6.9 Методы запроса

Squid распознает несколько методов запросов как описано в RFC 2616. Новые версии Squid (2.2.STABLE5 и выше) также распознают расширения RFC 2518 ``HTTP Extensions for Distributed Authoring -- WEBDAV".

```
method    defined    cachabil.  meaning
--------- ---------- ---------- -----------------------------------------
GET       HTTP/0.9   possibly   object retrieval and simple searches.
HEAD      HTTP/1.0   possibly   metadata retrieval.
POST      HTTP/1.0   CC or Exp. submit data (to a program).
PUT       HTTP/1.1   never      upload data (e.g. to a file).
DELETE    HTTP/1.1   never      remove resource (e.g. file).
TRACE     HTTP/1.1   never      appl. layer trace of request route.
OPTIONS   HTTP/1.1   never      request available comm. options.
CONNECT   HTTP/1.1r3 never      tunnel SSL connection.

ICP_QUERY Squid      never      used for ICP based exchanges.
PURGE     Squid      never      remove object from cache.

PROPFIND  rfc2518    ?          retrieve properties of an object.
PROPATCH  rfc2518    ?          change properties of an object.
MKCOL     rfc2518    never      create a new collection.
MOVE      rfc2518    never      create a duplicate of src in dst.
COPY      rfc2518    never      atomically move src to dst.
LOCK      rfc2518    never      lock an object against modifications.
UNLOCK    rfc2518    never      unlock an object.
```

# 6.10 Коды иерархий

В Squid-2 используются следующие коды иерархий:

**NONE**

> Для TCP HIT, неудачных TCP, запросов cachemgr и всех UDP-запросов - нет иерархической информации.

**DIRECT**

Объект был получен напрямую с сервера.

**SIBLING_HIT**

Объект был получен с кеша sibling, который ответил UDP_HIT.

**PARENT_HIT**

Объект был запрошен из кеша parent, который ответил UDP_HIT.

**DEFAULT_PARENT**

ICP-запросы не посылались. Парент был выбран, потому-что для него указано ``default'' в конфигурационном файле.

**SINGLE_PARENT**

Объект был запрошен с того парента, который соответствует данному URL.

**FIRST_UP_PARENT**

Объект был получен с первого парента в списке.

**NO_PARENT_DIRECT**

Объект был получен напрямую с сервера, т.к. нет парента для данного URL.

**FIRST_PARENT_MISS**

Объект был получен с самого быстрого парента (возможно из-за приоритета) исходя из RTT.

**CLOSEST_PARENT_MISS**

Этот парент был выбран, т.к. он имеет меньшее значение RTT к запрашиваемому серверу. См. также конфигурационную *closests-only* для соседа.

**CLOSEST_PARENT**

Выбор парента был основан на нашем собственном измерении RTT.

**CLOSEST_DIRECT**

Наше собственное измерение RTT вернуло меньшее время, чем любой парент.

**NO_DIRECT_FAIL**

Объект не может быть запрошен из-за настроек файервола (см. также *never_direct* и сопутствующие материалы), нет доступных парентов.

**SOURCE_FASTEST**

Был выбран оригинальный сервер, т.к. ping достигает его быстрей всего.

**ROUNDROBIN_PARENT**

Не было получено ICP-ответов ни от одного из парентов. Парент был выбран т.к. он помечен как round robin в конфиге и имеет меньшее число использования.

**CACHE_DIGEST_HIT**

Сосед был выбран, потому-что cache digest сообщил о хите. Эта опция впоследствии была заменена, чтобы различать parent-ов и sibling-ов.

**CD_PARENT_HIT**

Парент был выбран, потому-что cache digest предсказал хит.

**CD_SIBLING_HIT**

Сиблинг был выбран, птому-что cache digest предсказал хит.

**NO_CACHE_DIGEST_DIRECT**

похоже не используется?

**CARP**

Сосед был выбран по CARP.

**ANY_PARENT**

часть *src/peer_select.c:hier_strings[]*.

**INVALID CODE**

часть *src/peer_select.c:hier_strings[]*.

Почти каждому из них может предшествовать 'TIMEOUT_', если 2-х секундное (по умолчанию) ожидание ICP-ответов от всех соседей оканчивается таймаутом. См. также конфигурационную опцию *icp_query_timeout*.

Следующие коды иерархии удалены в Squid-2:

```
code                  meaning
--------------------  -------------------------------------------------
PARENT_UDP_HIT_OBJ    hit objects are not longer available.
SIBLING_UDP_HIT_OBJ   hit objects are not longer available.
SSL_PARENT_MISS       SSL can now be handled by squid.
FIREWALL_IP_DIRECT    No special logging for hosts inside the firewall.
LOCAL_IP_DIRECT       No special logging for local networks.
```

# 6.11 *cache/log* (Squid-1.x)

Этот файл имеет очень неудачное название. Также часто это называется *swap log*. Это записи о каждом

объекте кеша записанном на диск. Этот файл читается, когда запускается Squid, чтобы ``перегрузить'' кеш. Если вы удалите этот файл когда Squid НЕ запущен, то вы потеряете содержимое вашего кеша. Если вы удалите этот файл когда Squid УЖЕ запущен, то вы можете достаточно просто воссоздать его. Самый безопастный путь - просто завершить запущенный процесс:

```
% squid -k shutdown
```

This will disrupt service, but at least you will have your swap log back. Другой способ - вы можете сказать squid провести ротацию логов. Это также приведет к созданию нового swap-лога.

```
% squid -k rotate
```

В Squid-1.1 он содержит шесть полей:

1. **fileno**: The swap file number holding the object data. Это связано с именем файла в вашей файловой системе.
2. **timestamp**: Это время последней проверки объекта на "свежесть". The time is a hexadecimal representation of Unix time.
3. **expires**: Значение заголовка Expires в HTTP-ответе. Если заголовок Expires отсутствуе, поле будет иметь значение -2 или fffffffe. Если заголовок Expires присутствовал, но был неверен, значение будет -1 или ffffffff.
4. **lastmod**: Значение заголовке Last-Modified в HTTP-ответе. Если отсутсвтует, то значение будет -2, если неверен, то -1.
5. **size**: Размер объекта включая заголовок.
6. **url**: URL для этого объекта.

## 6.12 *swap.state* (Squid-2.x)

В Squid-2 лог-файл свопа теперь называется *swap.state*. Это бинарный файл, содержащий контрольные суммы MD5 и поля *StoreEntry*. См. Programmers Guide для получения информации по данной теме, а также описания формата этого файла.

Если вы удалите *swap.state*, когда Squid запущен, просто укажите Squid сделать rotate его лог-файлов:

```
% squid -k rotate
```

Можно также остановить Squid и он перезапишет этот файл перед завершением работы.

Если вы удалите *swap.state*, когда Squid не запущен, вы не потеряете содержимое вашего кеша. В этом случае Squid просканирует все кеш-директории и прочитает каждый свап-файл, чтобы перестроить кеш. Это может занять много времени, поэтому вы должны быть осторожны.

По умлочанию файл *swap.state* располагается в корне каждой *cache_dir*. Вы можете переместит логи в другую директорию, используя опцию *cache_swap_log*.

## 6.13 Какие лог-файлы я могу безопасно удалить?

Вы не когда не должны удалять *access.log*, *store.log*, *cache.log* или *swap.state*, когда Squid запущен. В Unix вы можете удалить открытый процессом файл. Однако место в файловой системе не будет освобождено пока процесс этот файл не закроет.

Если вы случайно удалили *swap.state*, когда Squid был запущен, вы можете восстановить его, следуя инструкциям из предыдущего вопроса. Если вы удалили другой лог во время работы Squid, вы не сможете его восстановить.

Правильный путь управления лог-файлами - использование ключа ``rotate''. Вам следует делать ротацию ваших логов не менее раза в сутки. Текущий лог-файл закрывается и переименовывается с расширением в виде числа (.0, .1 и т.п.). Если у вас есть желание, можете написать свой собственный скрипт для архивации или удаления лог-файлов. Если нет, то Squid будет хранить только такое число копий каждого лог-файлов, какое указано в опции *logfile_rotate*. Процедура ротации лог-файлов также создает новый файл *swap.state*, но не оставляет пронумерованных версий старых файлов.

Чтобы сделать ротацию логов Squid, просто используйте коамнду:

```
squid -k rotate
```

К примеру следующий пример демонстрирует ротацию логов в полночь при помощи cron:

```
0 0 * * * /usr/local/squid/bin/squid -k rotate
```

## 6.14 Как мне отключить лог-файлы в Squid?

Выключение *access.log*:

```
cache_access_log /dev/null
```

Выключение *store.log*:

```
cache_store_log none
```

Выключать *cache.log* - плохая идея, т.к. этот файл содержит много важной отладочной информации и сообщений о статуте. Однако, если действительно этого хотите: Выключение *cache.log*:

```
cache_log /dev/null
```

## 6.15 Мой лог становиться очень большим!

Вам необходимо делать *rotate* для ваших лог-файлов при помощи cron. К примеру:

```
0 0 * * * /usr/local/squid/bin/squid -k rotate
```

## 6.16 Хочу использовать другую утилиту для управления файлами журналов.

Если установить *logfile_rotate* в 0, то Squid просто закрывает и открывает заново файлы журналов. Это позволяет другим системам управления файлами журналов типа *newsyslog* поддерживать обработку его логов.

## 6.17 Управление лог-файлами

Предпочитаемый для анализа файл журнала - *access.log* в родном формате. For long term evaluations, the

log file should be obtained at regular intervals. Squid предполагает простое использование API для ротации файлов журналов, для того, чтобы они могли быть удалены (или перемещеныd) без нарушая функциональногсти кеша в процессе работы. Эта процедура была описана выше.

В зависимости от количества дискового пространства, выделенного под размещение файлов журналов, рекомендуется запускать посредством cron задания по ротации логов каждые 24, 12 или 8 часов. Вам таже необходимо установить достаточно большое значение для *logfile_rotate*. Во время простое вы можете безопасно передавать файлы журналов на ваш анализирующий их хост за один раз.

Before transport, the log files can be compressed during off-peak time. On the analysis host, the log file are concatinated into one file, so one file for 24 hours is the yield. Обратите также внимание, что при включенном *log_icp_queries*, у вас может оказаться до 1Гб несжатых логов в день при нагруженном кеше. Обратитесь к странице info вашего кеш-менеджера, что бы объективно оценить размер ваших файлов журналов.

Европейский проект DESIRE разработал некоторые основные правила, которым необходимо следовать при хранении и обработке файлов журналов:

- Respect the privacy of your clients when publishing results.
- Keep logs unavailable unless anonymized. Most countries have laws on privacy protection, and some even on how long you are legally allowed to keep certain kinds of information.
- Rotate and process log files at least once a day. Even if you don't process the log files, they will grow quite large, see section log-large . If you rely on processing the log files, reserve a large enough partition solely for log files.
- Keep the size in mind when processing. It might take longer to process log files than to generate them!
- Limit yourself to the numbers you are interested in. There is data beyond your dreams available in your log file, some quite obvious, others by combination of different views. Here are some examples for figures to watch:
    - The hosts using your cache.
    - The elapsed time for HTTP requests - this is the latency the user sees. Usually, you will want to make a distinction for HITs and MISSes and overall times. Also, medians are preferred over averages.
    - The requests handled per interval (e.g. second, minute or hour).

# 6.18 Почему я так часто получаю сообщения получаю ERR_NO_CLIENTS_BIG_OBJ?

Это сообщение значит, что запрошенный объект попал в режим ``Delete Behind'' и пользователь разорвал передачу данных. Объект попадет в режим ``Delete Behind'' если:

- Если он больше, чем *maximum_object_size*
- Если он скачивается с соседа, для которого установлена опция *proxy-only*.

# 6.19 Что значит ERR_LIFETIME_EXP?

Это значит, что наступил таймаут во время предачи объекта. Скорее всего доставка этого объекта проходила слишком медленно (или остановилась перед самым завершением) и пользователь прекратил обработку запроса. Однако, в зависимости от ваших установок *quick_abort*, Squid может продолжить попытки получить объект. Squid устанавливат максимальное значение промежутка времени на все открытые сокеты, после истечения которого обрабатываемый запрос отвергается и в лог записывается

сообщение ERR_LIFETIME_EXP.

## 6.20 Восстановление ``потеряных'' файлов из кеша.

> *I've been asked to retrieve an object which was accidentally destroyed at the source for recovery. So, how do I figure out where the things are so I can copy them out and strip off the headers?*

Следующий метод применим только для версии Squid-1.1:

Используйте *grep*, чтобы найти имя объекта (Url) в файле cache/log. Первое поле в этом файле - *номер файла* (целое число).

Потом найдите файл *fileno-to-pathname.pl*, он расположен в директории ``scripts'' исходников Squid. При использовании

```
perl fileno-to-pathname.pl [-c squid.conf]
```

номер файла будет читаться из stdin, а путь/имя к нему будет выдаваться на stdout.

## 6.21 Могу ли я использовать *store.log*,чтобы определить был ли объект закеширован ?

Вообщем да. Вы можете использовать *store.log*. чтобы узнать был ли определенный ответ *закеширован*.

Закешированные ответы помечаются в логе тегом SWAPOUT. Незакешированные ответы помечаются в журнале тегом RELEASE.

Однако ваш анализ должен учитывать и тот факт, что когда закешированный ответ удаляется из кеша (for example due to cache replacement), то он тоже будет занесен в *store.log* с тегом RELEASE. Чтобы различить эти два варианты, вы можете обратить внимание на поле номера файла (3-е). Когда удаляются незакешированные ответа, то номер файла FFFFFFFF (-1). Любой другой нормер файла указывает на то, что удален закешированный ответ.

---