Descripción del servicio

Squid es un servidor de proxy-caché de alta capacidad para clientes de Web. Soporta objetos HTTP, FTP y Gopher, y está diseñado para trabajar sobre cualquier sistema UNIX.

Un servicio de caché Web permite a sus usuarios configurar sus navegadores para dirigir sus accesos a la Web a un servidor proxy-caché en vez de ir directamente al origen del documento. El servidor caché se encarga de proporcionarle la página pedida, bien obteniéndola de su propia caché o accediendo al documento original.

Cada vez que un usuario accede a un documento a través de la caché, este documento queda almacenado. De esta manera, si otro usuario accede al mismo documento más tarde, se beneficia de un acceso más rápido desde la caché en vez de tener que ir al origen por líneas de comunicaciones, en muchos casos sobrecargadas y por lo tanto lentas. Al dar servicio a muchos usuarios la caché consigue un porcentaje de aciertos muy elevado; cerca del 50% de las peticiones son resueltas por la propia caché o haciendo una petición al origen de validación de que el documento no ha cambiado, sin tener que traérselo en caso afirmativo.

Al acceder a la Web a través de una caché se obtiene un acceso más rápido a los documentos en un gran porcentaje de los accesos. Una caché bien dimensionada llega a dar porcentajes de acierto cercanos e incluso superiores al 50%. El porcentaje de aciertos viene dado por el número total de documentos que la caché puede servir sin realizar consultas al original frente al número total de accesos.

Como efecto lateral también se logra que los accesos por las líneas internacionales a un mismo documento se reduzcan a un sólo acceso cada cierto tiempo logrando así un uso más eficiente de las líneas con la consecuente mejora de rendimiento de las comunicaciones.

La caché sigue un sofisticado mecanismo de Refresco de Documentos, mediante el cual antes de servirle un documento comprueba su validez en base a parámetros como la última modificación del documento, el tiempo que lleva en la caché y otros. Muchas veces antes de servirle un documento lanza al sitio original una petición para que coteje si el documento no ha sido modificado desde que la caché lo obtuvo, y en caso contrario procede a acceder directamente al original.

La mayoría de los visualizadores Web permiten entre sus opciones de configuración el especificar un proxy-caché a través del cual acceder a la WWW. Lo único que tiene que averiguar es el nombre de la máquina caché de su organización y el puerto a través del cual se accede al servicio. Una vez configurado el acceso a Web a través de un proxy-caché todos los accesos se realizarán a través de la caché sin que el usuario note diferencia alguna respecto a los accesos normales.

Al pasar todos los accesos a la Web por el proxy-caché, si esta máquina falla pueden fallar todos los intentos de acceso. Esto es fácilmente evitable en aquellos visualizadores que permiten especificar varias cachés alternativas o incluso que empiecen a realizar accesos directos automáticamente cuando falle la caché.

Paquetes de Squid

Squid está compuesto por diferentes programas:

- El programa principal *squid*, que lleva el peso de la aplicación;
- El programa *dnsserver*, que se encarga de la búsqueda de nombres de dominio;
- Otras herramientas clientes y de mantenimiento.

Cuando Squid se ejecuta, produce un número configurable de procesos *dnsserver*, y cada uno de ellos realiza su propio *DNS lookup*. De este modo, se reduce la cantidad de tiempo que la caché debe esperar a estas búsquedas DNS.

Requisitos e instalación de Squid

La instalación de Squid no requiere un sistema de enormes prestaciones; el recurso más importante a considerar es la memoria física. Una buena máquina para caché podría ser la siguiente:

- Pentium II a 300 MHz
- 512 MB RAM
- 5 discos SCSI de 9 GB

Para estudiar esta herramienta, hemos bajado de la red la última versión disponible, recogida en el fichero "*squid-2.3.STABLE2-src.tar.gz*". Para descomprimir e instalar este paquete en nuestro ordenador, y así estudiarlo con más profundidad, ejecutamos los siguientes comandos:

```
tar xzf squid-2.3.STABLE2-src.tar.gz
./configure --prefix=/usr/local/squid
make install
```

Con estos comandos conseguimos una instalación válida, aunque tanto `configure` como `make` son susceptibles de diversas opciones. La aquí empleada para `configure`, `--prefix`, nos permite elegir el directorio de instalación, y es la más utilizada. El resto de opciones, de carácter más particular, pueden verse con `configure --help`. También podemos seleccionar otras posibilidades, antes de la compilación, en el fichero `src/Makefile`.

Una vez realizada la instalación, tenemos a nuestro alcance dos ficheros interesantes:

- *QUICKSTART* – una guía sencilla y rápida de configuración, muy útil si nuestro objetivo no incluye características muy complejas.

- *Squid.conf* – define la configuración de Squid. Este es el fichero del que vamos a tratar en la mayor parte del trabajo restante. Se trata de un fichero de tamaño considerable (más de 30 páginas), lo que nos hace pensar que Squid ofrece incontables posibilidades de uso en nuestra máquina. Por ello, he decidido dedicarme a las que me han parecido más útiles o interesantes.

Ejecución de Squid

Una vez editado el fichero de configuración, podemos proceder a activar Squid. Para la versión que estamos tratando (2.2), primero hay que crear los directorios swap, con la orden:

```
/usr/local/squid/bin/squid -z
```

Posiblemente la mejor forma de lanzar Squid por primera vez es desde el terminal, lo que nos permite observar ciertos parámetros en pantalla y planificar una posible depuración de la configuración:

```
/usr/local/squid/bin/squid -NCd1
```

Si todo va bien, obtendremos la siguiente respuesta:

```
Ready to serve requests.
```

Si queremos lanzar squid en segundo plano, como un "*daemon process*", lo único que tenemos que hacer es retirar las opciones del caso anterior.

Squid 2.2 tiene la propiedad de ejecutarse automáticamente cuando el sistema se inicia. Una vez lanzado la primera vez, se observa en el fichero de mensajes *syslog* el texto siguiente:

```
May 08 23:44:11 kitty squid[14616]: Squid parent: child
process 14617 started
```

El proceso hijo que vemos en la sentencia anterior es el que hace todo el trabajo, mientras que el padre espera a que el hijo termine. Si el hijo terminase inesperadamente, el padre lanza otro hijo automáticamente; y si por alguna causa no fuese capaz, terminaría después de un cierto periodo de tiempo.

Podemos lanzar Squid con multitud de opciones. Algunas de las más interesantes son:

- **-a :** Permite especificar un puerto alternativo para las peticiones de entrada HTTP. Resulta útil para realizar pruebas en un puerto no estándar.
- **-d :** "Debugging". Esta opción nos permite leer los mensajes que se generan en la propia ejecución del programa Squid, tal y como hemos comentado anteriormente.
- **-f :** Permite especificar un fichero *squid.conf* alternativo.
- **-h :** Escribe el fichero de ayuda.
- **-k reconfigure :** Envía señal de HUP, que provoca que Squid relea sus ficheros de configuración.
- **-k shutdown :** Envía señal de TERM, que provoca que Squid espere un breve espacio de tiempo a que terminen sus conexiones, y acabe.
- **-k interrupt :** Envía señal de INT, que provoca la parada inmediata del squid, sin esperar a que terminen sus conexiones.

- **-k kill:** Envía señal de KILL, que provoca la salida inmediata del squid, sin cerrar conexiones ni archivos log.
- **-k check :** Comprueba si el Squid está ejecutándose o no.
- **-u :** Permite especificar un puerto alternativo para los mensajes ICP. Resulta útil para realizar pruebas en un puerto no estándar.
- **-v :** Imprime la versión de Squid.
- **-z :** Crea directorios swap en disco. Esta opción debe usarse cuando se instala Squid por primera vez, o tras la modificación de la opción CACHE_DIR del fichero de configuración.
- **-N :** Desactiva la opción de ejecución en segundo plano.

Configuración de Squid

Ya he mencionado anteriormente que la configuración de Squid puede ser larga y compleja, debido a las enormes posibilidades que nos ofrece. En los siguientes apartados, he querido destacar algunas de estas opciones.

Configuración del puerto

Una de las posibilidades más sencillas que se nos ofrecen es la elección del puerto por el que Squid recibirá las peticiones HTTP. El puerto por defecto es el 3128 (si usamos *httpd-accel*, el 80). Para modificarlo, simplemente nos dirigimos a la siguiente línea:

```
http_port 3128
```

Se pueden especificar varias líneas HTTP_PORT en el fichero de configuración (en versiones de Squid modernas), de modo que se soluciona el problema de que un puerto no pueda usarse, bien porque esté ocupado o bien debido a problemas de permisos.

También podemos mover el puerto ICP, que por defecto se encuentra en 3130. Mediante este puerto nuestra caché se comunica con las cachés vecinas, que podemos configurar tal y como se cuenta en el siguiente apartado.

Configuración de una jerarquía

Para introducir nuestra caché en una jerarquía, sencillamente tenemos que especificar en la directiva CACHE_HOST de *squid.conf* la caché padre y los posibles "cachés hermanos" (*sibling nodes*). Por ejemplo, el siguiente código configura, en *childcache.example.com* un padre y dos hermanos:

```
cache_peer   parentcache.example.com   parent  3128  3130
cache_peer   childcache2.example.com   sibling 3128  3130
cache_peer   childcache2.example.com   sibling 3128  3130
```

También podemos delimitar los dominios para los cuales se solicitará el servicio de los cachés vecinos, mediante la directiva CACHE_PEER_DOMAIN.

Tamaño de la caché

En CACHE_MEM podemos introducir la cantidad de memoria que queremos usar para la ejecución de Squid. En realidad, este parámetro sólo se refiere a determinados parámetros de la memoria, por lo que debemos calcular que el tamaño real del proceso será aproximadamente el triple del que marquemos aquí:

```
cache_mem 8 MB
```

Podemos también configurar el tamaño máximo de objetos que guardaremos en nuestra caché, de forma que los objetos de mayor tamaño serán rechazados.

```
maximum_object_size 4096 KB
```

Tamaño del directorio caché

La mayoría de la gente tiene reservada una partición de disco para la caché que configuramos con Squid. El tamaño a elegir depende de las necesidades de cada cliente; aunque para un caso "típico", con 9 GB de disco, una buena elección sería dedicar unos 7500 MB.

Podemos cambiar también el nombre de este directorio, y configurar el número máximo de subdirectorios de primer nivel y de segundo nivel que podemos soportar. En el ejemplo siguiente se incluyen los valores por defecto de estos parámetros:

```
cache_dir  /var/spool/squid  7500 16 256
```

Número de dnsserver

Squid usa los procesos *dnsserver* porque las rutinas de la librería *gethostbyname*, que son las que se usan normalmente para convertir nombres de Web en direcciones de Internet, se bloquean hasta que se obtiene una respuesta. Puesto que sólo hay un proceso *squid*, cada usuario tendría que esperar a que esta rutina obtuviera respuesta; por esta razón, se usan procesos independientes *dnsserver*.

Es importante tener *dnsserver* suficientes para cubrir todas las peticiones que pudieran llegar; una regla típica es comprobar cuál es el máximo de *dnsserver* que nuestro sistema suele utilizar y tener disponibles al menos dos más. De otro modo, el sistema irá muy lento e incluso podría llegar a bloquearse. Para añadir más *dnsserver*, sólo tenemos que incrementar el número de DNS_CHILDREN en el fichero de configuración.

Actualización de la información

Cada cierto tiempo, es de esperar que los objetos de nuestra caché se actualicen. Para ello, Squid calcula una serie de valores (referentes al tiempo que lleva el objeto en la caché, cuántas veces se ha accedido a él desde que está allí, sus modificaciones...) y las compara con las establecidas en la directiva REFRESH_PATTERN.

Pertenencia a NLANR

Se puede configurar la caché para hacerla pertenecer a la denominada jerarquía NLANR, que proporciona un gran número de padres y hermanos a los que conectarse, mejorando así la eficiencia de nuestro servicio caché. Para ello, sencillamente tenemos que habilitar en *squid.conf* las siguientes líneas:

```
cache_announce 24
announce_to sd.cache.nlanr.net : 3131
```

De esta forma, quedaremos registrados en la jerarquía NLANR. Podemos visitar desde la Web su base de datos, y descubrir qué otras cachés están disponibles para conectarnos con ellas. Eso sí, para realizar esta conexión es necesario obtener un permiso de sus administradores.

Control de acceso

Squid nos permite configurar opciones referidas al control de acceso, lo que nos permite múltiples posibilidades. Entre ellas se encuentra una de especial interés: la de trabajar detrás de un firewall. En ese caso, nosotros no podremos realizar conexiones directas con el mundo exterior, por lo que necesitamos usar un *parent cache*.

Debemos usar la directiva NEVER_DIRECT de *squid.conf* para indicar que las peticiones deben ser reenviadas a nuestro *parent cache* si pertenecen al exterior del firewall. Por ejemplo, si nuestro Squid puede conectarse a todos los servidores que terminan con *mydomain.com*, pero debe usar el *parent cache* para el resto, debemos escribir:

```
acl INSIDE dstdomain mydomain.com
never_direct deny INSIDE
```

Si tenemos varios *parent cache*, es posible que Squid no sepa cuál usar tras leer las sentencias anteriores; podemos marcar uno de ellos como “solución por defecto”, así:

```
cache_peer xyz.mydomain.com parent 3128 0 default
```

También podemos configurar Squid para que reenvíe todas las peticiones a otro proxy. Para ello, sencillamente tenemos que seguir los siguientes pasos:

- Proporcionar un *parent cache.*
- Indicar a Squid que no puede conectarse de forma directa con los servidores.

Las instrucciones a ejecutar, por lo tanto, serán las siguientes:

```
cache_peer parentcache.foo.com parent 3128 0 no-query
default
acl all src 0.0.0.0/0.0.0.0
never_direct allow all
```

X-Forwarders

Cuando se usa una proxy-caché, un servidor no ve la conexión procedente del cliente original. Hay determinados servidores que implementan sus accesos dependiendo del tipo de clientes; para acomodarnos a estos, Squid tiene su propia petición:

```
X-Forwarded For: 128.138.243.250, unknown, 192.52.106.30
```

Especificación de usuarios y grupos

Podemos limitar el acceso a nuestra caché a determinados usuarios y/o grupos. Para ello necesitamos escribir algo como esto:

```
ident_lookup on
acl_friends user kim joe mary
http_access allow friends
http_acess deny all
```

httpd-accelerator

Una posibilidad que suele usarse es la configuración de Squid como acelerador de httpd. Para ello, tenemos que configurar las siguientes líneas con el nombre y puerto del servidor HTTP:

```
httpd_accel_host hostname
httpd_accel_port port
```

Al habilitar esta opción, se deshabilita automáticamente el servicio de proxy-caché y el protocolo ICP. Si queremos que nuestro Squid funcione también con estas opciones en paralelo, habrá que habilitar el siguiente *flag*:

```
httpd_accel_with_proxy on
```

Existen muchas otras opciones de configuración: para conocerlas todas, sólo tenemos que leer con calma el fichero de configuración y modificar o "descomentar" las opciones que nos resulten pertinentes. Squid no relee periódicamente su fichero de configuración, por lo que si realizamos alguna modificación, lo más sencillo para asegurarnos que la recibe es utilizar el siguiente comando:

```
squid -k reconfigure
```

Uso de Squid : Comunicación con browsers

La mayoría de Web Browsers soportan una proxy-caché, y pueden configurarse fácilmente para usar a Squid como servidor. A modo de ejemplo, comentaremos algunos detalles sobre los navegadores más conocidos:

- Netscape: nos permite realizar una configuración manual muy sencilla. Todo lo que tenemos que hacer es:

    - Seleccionar *Options – Network Preferences – Proxies – Manual Proxy Configuration – View*
    - Para cada protocolo soportado por Squid (HTTP, FTP y Gopher, por defecto), introducir el número de puerto de su servidor en la columna *Port.*

    Se puede también realizar una configuración automática más avanzada, seleccionando *Automatic Proxy Configuration* y escribiendo en la ventana de texto correspondiente un JavaScript de configuración; aunque esta opción se reserva a administradores más expertos.

- Lynx: Lo más sencillo es introducir las siguientes líneas en el fichero *lynx.cfg*:

```
http_proxy:http://mycache.example.com:8080
ftp_proxy:http://mycache.example.com:8080
gopher_proxy:http://mycache.example.com:8080
```

- Internet Explorer: su configuración es muy similar a la que usamos con Netscape:

    - Seleccionar *View - Options – Connect Through Proxy Server – Proxy Settings*
    - Para cada protocolo soportado por Squid (HTTP, FTP y Gopher, por defecto), introducir el número de puerto de su servidor en la columna *Port*.

Mantenimiento de Squid

En realidad, una vez configurado correctamente, el Squid no precisa de una atención especial. No obstante, si queremos seguir con cierta regularidad su funcionamiento, podemos ayudarnos de los ficheros *log*. Los *log*'s son ficheros de información sobre Squid que nos permiten observar su comportamiento frente a los eventos externos. Algunas de sus funciones son:

- Recoger datos sobre los accesos a la información requerida
- Registrar los errores de configuración y funcionamiento
- Guardar información sobre el consumo de recursos

etc.

Además de la visita periódica a la información de estos ficheros, se pueden realizar algunas otras operaciones de forma periódica. Algunas de ellas son las siguientes:

- Ver estadísticas del sistema Squid – la propia distribución incluye un CGI denominado *cachemgr.cgi* (Cache Manager) que permite observar estas estadísticas a través de un Web Browser. El *cache manager* es un programa que ayuda a la buena administración de la caché y que además permite observar estadísticas sin acceder directamente al servidor.

- Relanzar Squid con un directorio caché limpio – las instrucciones a seguir son las siguientes:

- `squid –k shutdown` – detenemos el proceso *squid* en curso
- `echo " " > cache1/swap.state` – reescribimos los ficheros swap.state en cada directorio caché, vaciándolos. Podemos tratar de borrarlos, pero el sistema no nos permite hacerlo directamente y habría que emplear técnicas más sofisticadas.
- `squid` – una vez realizada la operación anterior, lanzamos de nuevo el proceso.

- Borrar un objeto de la caché - no nos estará permitido a menos que esté debidamente configurado en *squid.conf*, mediante, por ejemplo, las siguientes líneas:

```
Acl PURGE method purge
Acl localhost src 127.0.0.1
http_access allow purge localhost
http_access deny purge
```

De este modo, sólo el usuario del *localhost* puede eliminar objetos. Para hacerlo, podemos usar el programa *client*:

```
Client –m PURGE http://www.miscreant.com
```