

Instalación de una caché WWW

En esta página se describen los pasos para instalar en una máquina Unix un servidor de proxy-caché de páginas Web.

Programas y máquinas para instalar una caché:

A día de hoy existen no sólo muchas aplicaciones software para realizar tareas de proxy-caché de páginas Web sino también muchas máquinas diseñadas exclusivamente para realizar tal función. Entre los vendedores de máquinas exclusivas para caché WWW se encuentran: [NetCache](#), [Cisco](#), [Inktomi](#), [Novell](#) y [DynaCache](#) entre muchos otros.

Respecto a las cachés implementadas como aplicaciones para correr en máquinas de propósito general existen numerosas implementaciones de cachés WWW para distintas plataformas. La mayor parte de la implementaciones de libre distribución están desarrolladas para correr sobre máquinas Unix. A continuación listamos algunos programas de proxy-caché disponibles en el mercado y/o la red:

- [Squid](#): evolución de libre distribución de Harvest.
- [NetApp](#): evolución comercial de Harvest, corre tanto en Unix como en Windows NT.
- [Netscape](#)
- [Microsoft](#)
- [Apache](#) es un servidor web que incluye un módulo de proxy-caché.

RedIRIS recomienda a sus instituciones el software [Squid](#) que está disponible en [nuestro FTP](#). Esta recomendación viene guiada por los siguientes motivos:

- Squid es software de libre distribución, esto no sólo implica que sea gratuito, sino que además al estar su código fuente libremente accesible y modificable, hace posible la modificación del programa en caso de querer mejorar sus prestaciones o características.
- Squid tiene un rendimiento superior a las demás implementaciones de proxy-cachés implementadas por software.
- Squid soporta el protocolo ICP para integrar las cachés en grupos colaborativos, muchas de las demás implementaciones no soportan este punto. ICP permite además de formar jerarquía hacer que el fallo de una caché sea superado por la colaboración de las otras.
- Squid soporta además otros mecanismos de comunicación entre cachés más eficientes como las Cache-Digest por el cual las cachés intercambian cada cierto tiempo (una hora) un mapa de bits de la tabla de digests MD5 del hash que contiene todas las URLs de cada caché.
- Squid se ha popularizado tanto que ya hay muchas aportaciones para tratamiento de estadísticas, administración por Web y muchas otras; además su grupo de desarrolladores es tan grande que se incorporan continuamente nuevas funcionalidades de acuerdo con las últimas tendencias en tecnología de Redes y sistemas de información (ej: multicasting)
- Squid es utilizado en las grandes agrupaciones de cachés mundiales, como la NLNR.
- Squid compila muy fácilmente en casi cualquier plataforma Unix, y su instalación es bastante simple.

Quizás uno de los pocos inconvenientes de Squid es que su desarrollo continuado da lugar a nuevas versiones cada muy poco tiempo con corrección de errores y nuevas características, no obstante, desde hace tiempo todas las versiones son bastante estables.

2.- Instalación de Squid en una máquina Unix:

El primer paso a dar es conseguir una distribución del código fuente de Squid, puede conseguir tal distribución en [Mirror oficial de Squid en Uniovi](#) o a partir de [la página oficial de Squid](#)

Actualmente trabajamos con la versión 2.X de squid que incorpora notables mejoras de rendimiento y flexibilidad sobre la anterior versión 1.1.X (utilizamos X para denotar la sub-versión del programa)

La distribución fuente de Squid-2 es un archivo con nombre squid-2....-src.tar.gz dónde los tres puntos vienen substituidos por información sobre la versión concreta. Al tener extensión .gz deberá descomprimir el archivo con gunzip (o 'gzip -d', si no lo tiene busque en [el FTP de RedIRIS](#)) La extensión .tar denota que es un paquete de ficheros, deberá pues una vez descomprido desempaquetar el fichero con la orden unix tar xf squid-2....-src.tar tras lo cual tendrá un directorio llamado squid-2.... en el directorio de trabajo dónde ha ejecutado las ordenes anteriores. La secuencia completa desde una sesión Unix podría ser:

```

> cd /directorio/de/trabajo
> ftp ftp.rediris.es
Name: ftp
Password: sudireccion@email
...
> cd software/infosystems/Squid/squid-2
...
> binary
...
> get squid-2.0.RELEASE-src.tar.gz

...
> bye
...
> gunzip squid-2.0.RELEASE-src.tar.gz
> tar xf squid-2.0.RELEASE-src.tar.gz
> cd squid-2.0.RELEASE

```

Generación del programa squid:

Una vez en el directorio con la distribución de Squid tendrá que compilar el software. La instalación de Squid al igual que su ejecución pueden realizarse sin necesidad de ser root. Los pasos para una instalación exitosa son:

- En primer lugar ejecutar el comando configure, al que deberá proporcionarle varias opciones sobre el programa a generar y el directorio dónde piensa instalar Squid salvo que lo vaya a instalar en /usr/local/squid. A continuación vea las órdenes necesarias con las opciones recomendadas por nosotros. Proceda:

```

> ./configure --enable-snmp --enable-cache-digests
--enable-err-language=Spanish --enable-icmp

```

ó

```

> ./configure --prefix=/directorio/de/instalacion
--enable-snmp --enable-cache-digests
--enable-err-language=Spanish --enable-icmp

```

El soporte de ICMP con --enable-icmp es opcional, interesa sobre todo en las cachés que actúen como padres de otras cachés.

- Una vez hecho el paso anterior bastará con ejecutar:

```

> make all

```

Lo cual realizará la compilación de todo el código del programa squid. Si la ejecución del comando no fuera exitosa intente obtener una versión actualizada del compilador gcc en su máquina, que esté accesible en el PATH y vuelva a empezar desde el configure. También puede probar a tocar a mano el fichero acconfig.h o el fichero src/config.h para cambiar después del configure parámetros sobre la instalación en la que se está generando el programa. El siguiente paso es la instalación de los programas en sus directorios con:

```

> make install

```

Si ha instalado el soporte de ICMP también deberá teclear:

```

> make install-pinger

```

que copiará los programas necesarios al directorio de instalación (/usr/local/squid por defecto) al igual que los archivos de configuración.

Nota: parece que las últimas versiones no llevan todos los mensajes de Error en Español necesarios, deberá por tanto copiarlos de <ftp://ftp.rediris.es/tmp/Spanish> a su directorio /usr/local/squid/etc/errors.

Fichero de configuración:

Una vez instalado el programa sólo falta editar el fichero de configuración y poner Squid a correr. El fichero de configuración se llama **squid.conf**. Después del make install queda creado un fichero de configuración de ejemplo en el directorio etc de la raíz de la instalación (o sea, normalmente en /usr/local/squid/etc). Será este fichero el que editemos para adecuarlo al entorno de nuestra caché.

Para editar el fichero de configuración basta con quitar el caracter de comentario a principio de línea (#) en las directivas más importantes y fijar los valores adecuados, las directivas que queden como comentarios (líneas empezando por #) o no aparezcan tomarán sus valores por defecto, que en la mayoría de los casos son adecuados.

A continuación comentamos las directivas imprescindibles para que Squid se ejecute con una configuración correcta y no de problemas:

cache_peer: con esta directiva configura otras cachés padres o hermanas (ver [coordinación de cachés](#)). Por ejemplo, un modo de encadenar su caché con la de RedIRIS (previa autorización de cachemgr@rediris.es) bastaría con las líneas:

```
cache_peer gate1.rediris.es parent 8080 3130 no-digest no-query no-netdb-exchange round-robin cache_peer gate2.rediris.es
parent 8080 3130 no-digest no-query no-netdb-exchange round-robin
```

Si no piensa encadenar su caché con ninguna otra deje esta línea tal y como aparece en el fichero de configuración (comentada)

cache_mem: la cantidad de memoria que va a utilizar Squid, es un valor orientativo, asegurese de dar un valor razonable y que en ningún caso exceda la tercera parte de la memoria física de la máquina, más aún si tiene otros servicios corriendo. Ej:

```
cache_mem 32 MB
```

cache_dir: es muy importante que indique en esta línea la ubicación, la capacidad y la distribución de directorios dónde se van a ubicar los archivos de la caché. Puede especificar varias líneas con varios directorios, los valores de 16 y 256 para la distribución de directorios parecen razonables, en la capacidad especifique un poco menos de la capacidad real del directorio seleccionado.

```
cache_dir /usr/local/squid/cache 200 16 256
```

En muchos sistemas operativos la caché de inodos tiene un límite de caracteres en sus entradas por lo que conviene que el camino de directorio de la directiva anterior sea corto (mejor "/cache" que "/home/volumen2/usuario2/local/squid...")

local_domain: especifica el dominio al cual accederá directamente sin consultar otras cachés en caso de estar coordinada con otras cachés. Típicamente pondrá aquí el dominio de su organización o directamente 'es' (para España). En Squid 2.0 la forma de especificar como local el dominio org.es es:

```
acl local-servers dstdomain org.es
always_direct allow local-servers
```

También conviene ir directo siempre a los puertos de SSL:

```
always_direct allow SSL_ports
```

Puede especificar varios dominios locales de su organización separados por espacios dónde esta escrito org.es

cache_mgr: la dirección de correo electrónico del responsable de la caché. Ej:

```
cache_mgr cachemgr@dominio.es
```

hierarchy_stoplist: URLs que no deben ser obtenidos de otra caché, normalmente los cgi-bin's:

```
hierarchy_stoplist cgi-bin ?
```

no_cache: patrones de URLs que no deben ser cacheados, normalmente los cgi-bin's:

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

cache_effective_user: el usuario y grupos con que se ejecutará el proceso squid. Por razones de seguridad no deben tener apenas permisos en el sistema (y por supuesto no tener shell, ej: nobody y nogroup), no obstante deben tener permisos plenos sobre los directorios de Squid (logs y caché) por ejemplo:

```
cache_effective_user nobody nogroup
```

visible_hostname: el nombre con el que se anuncia la caché. Normalmente:

visible_hostname proxy.dominio.es

prefer_direct: parámetro que define si se le da más preferencia a intentar resolver la consulta de manera directa "on" (por defecto), o por el contrario, a través de uno de sus padres "off".

prefer_direct off

Permisos de utilización de la caché: hay tres directivas que controlan los permisos que usted quiera fijar para utilización de la caché, tanto a nivel de usuarios finales como de otras cachés en un grupo de cachés.

Primeramente tendrá que definir varias listas de acceso, una de ellas con las sub-redes IP de los posibles clientes de su caché (o simplemente direcciones IP concretas si son pocos clientes) o directamente los dominios desde los que permite conexiones (o nombres de las máquinas concretas en el DNS); en caso de estar coordinándose con otras cachés tendrá así mismo que definir una lista de las cachés hermanas (si hay cachés padre no necesita configurarlas aquí, ver [coordinación de cachés](#)) bien por dirección IP o por nombre de máquina.

En el siguiente ejemplo las acl's distintas de 'clientes' y 'caches' se recomienda que se dejen intactas.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl all src 0.0.0.0/0.0.0.0
acl clientes srcdomain rediris.es
acl caches srcdomain proxy.org1.es proxy.org2.es
acl Safe_ports port 80 81 21 70 1025-65535
acl CONNECT method CONNECT

http_access deny manager !localhost
http_access deny !Safe_ports !SSL_ports
http_access deny CONNECT !SSL_ports

http_access allow clientes
http_access allow caches
http_access deny all

icp_access allow caches
icp_access deny all

miss_access allow clientes
miss_access deny all
```

Dónde rediris.es sería el dominio de su organización y cada proxy.org.es una caché hermana.

Puede consultar como ejemplo el [fichero de configuración de la caché de RedIRIS](#)

Consideraciones sobre seguridad:

Existen algunas consideraciones en cuanto a la seguridad que se deben tener en cuenta a la hora de instalar un proxy.

La primera se refiere al control del acceso o utilización del servidor. Como hemos visto en la sección anterior se soluciona mediante listas de acceso "acl" y parámetros: "http_access", "icp_access" y "miss_access". Con ello, se consigue evitar la utilización no autorizada de nuestra línea a través del proxy, y por tanto, el control de este recurso de la organización.

La segunda se refiere a las conexiones permitidas a través del proxy. El método CONNECT es una puerta para la conexión a otros servidores desde el proxy, y por tanto, debemos tener cuidado a la hora de definir a qué puertos se puede conectarse para atender a las peticiones. Para ello, "squid" incorpora (como se ve en la sección anterior) en su configuración por defecto las listas "Safe_ports" y "SSL_ports":

```
acl Safe_ports port 80 81 21 443 563 70 210 1025-65535
acl SSL_ports port 443 563
acl CONNECT method CONNECT

http_access deny !SSL_ports !Safe_ports
http_access deny CONNECT !SSL_ports
```

En este caso se permite la conexión, mediante CONNECT, a puertos típicos SSL: 443 y 563. Existe algún caso de servidores seguros que utilizan otros puertos no estandar, en cuyo caso, es posible que el administrador del proxy tenga que abrir algún otro puerto, añadiéndolo a la lista SSL_ports.

Consideraciones sobre el rendimiento de la aplicación:

El proceso Squid tiene un perfil de rendimiento en el que la mayor parte de su actividad es la entrada y salida con los discos duros. Si monitoriza el proceso verá una cierta desproporción entre el tiempo de CPU consumido por la parte de usuario y el consumido por la parte de sistema y de espera por entrada y salida.

El rendimiento de Squid se verá sensiblemente mejorado en la medida en que se disponga de más memoria física (y actualice en consecuencia el parámetro `cache_mem` visto más arriba) y se acelere lo máximo posible la velocidad de los accesos a discos. Como norma general intente separar en discos distintos la parte del sistema operativo, el swap, la caché y los logs de la caché. Intente que estos discos sean lo más rápidos posible y que no saturen la controladora que les conecta a la máquina (como norma general no tener más de cuatro discos SCSI por controladora)

Ejecutar Squid:

Una vez configurado el `squid.conf` necesita ejecutarlo una vez con el parámetro `-z` para que cree los directorios de la caché, si este paso no se ejecuta correctamente mire los permisos de todos los directorios de caché y de logs especificados en el fichero de configuración.

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

Una vez hecho esto bastará ejecutar `squid` para que la caché se ponga en funcionamiento. La mejor manera de hacerlo es con el script `RunCache` que queda tras la instalación en el directorio `bin`. Un ejemplo sería:

```
/usr/local/squid/bin/RunCache &
```

y a continuación observe el fichero `cache.log` en el directorio de logs y `squid.out` en el directorio base de Squid, pruebe un `tail -f /usr/local/squid/logs/cache.log` para ir viendo si todo va bien (no da mensajes de error).

Asegurese de invocar `RunCache` en alguno de los scripts de inicialización del sistema (`rc?.d/` ó `rc.local`) para que squid se ejecute cuando se re arranque la máquina.

3. ¿ Y ahora qué ?

En primer lugar deberá mandar un correo electrónico al [administrador de la caché de RedIRIS](#) dándole su impresión sobre la utilidad de esta página de ayuda a la instalación y si ha tenido algún problema que no esté descrito en esta página así como comunicarle cualquier gazapo contenido en ella.

Conviene igualmente que lea la página relativa al [mantenimiento de una caché](#)

Por otra parte es muy importante que tan pronto como considere que el servicio de caché WWW va a ser estable anuncie a todos los usuarios la existencia de la misma y promueva su uso. Lo ideal en el futuro es poner filtros para evitar cualquier acceso incontrolado a la WWW que no tenga lugar a través de la caché.

Para esto último es de bastante utilidad disponer en el servidor Web de la organización de una página explicativa del servicio de caché WWW con java-scripts de autoconfiguración de cachés (ver [las páginas de Netscape sobre este tema](#) compatibles con Netscape y Explorer versiones superiores a la 3.0).

A continuación tiene un ejemplo de un posible javascript. Debe poner el script en un fichero con extensión `.pac` y substituir las palabras `sudominio,suproxy,proxybackup,suotrodominio,puerto,etc` por los valores adecuados y `x.x.x.x` por la subred IP de sus clientes o bien quítelo):

```
function FindProxyForURL(url, host)
{
  if ((url.substring(0, 5) != "http:") &&
      (url.substring(0, 4) != "ftp:") &&
      (url.substring(0, 7) != "gopher:")) {
    return "DIRECT";
  }
  if (isPlainHostName(host) ||
      dnsDomainIs(host, ".sudominio.es") ||
      dnsDomainIs(host, ".suotrodominio.es") ||
      shExpMatch(host, "x.x.x.x") ||
      shExpMatch(host, "127.*"))
    return "DIRECT";
  else
    return "PROXY suproxy.sudominio.es:8080;
    PROXY proxybackup.otrodominio.es:puerto ; DIRECT";
}
```

También deberá instruir al servidor Web dónde aloje el script `.pac` a servir los documentos `.pac` con el tipo MIME adecuado. En Apache lo puede conseguir con la siguiente línea en el fichero de configuración de Apache:

```
AddType application/x-ns-proxy-autoconfig pac
```

Actualizado el 03/08/2000

RedIRIS © 1994-2002