

逢甲大學

資訊工程學系專題報告

Proxy Server之應用

學生： 鄭永福(四丁)

指導教授：賴榮滄 教授

中華民國九十二年五月

目錄

圖表目錄.....	4
第一章 導論.....	5
1.1 動機.....	5
1.2 目的.....	6
第二章 Proxy Server 概述.....	7
2.1 起源.....	7
2.2 優點.....	7
2.3 運作原理.....	8
2.4 squid安裝.....	10
第三章 Squid 概述.....	13
3.1 簡介.....	13
3.2 各主要元件功能.....	13
3.3 程式流程.....	16
第四章 系統製作.....	17
4.1 重導機制.....	17
4.2 代理伺服器設定.....	17
4.3 使用工具介紹.....	18

4.4 系統製作.....	19
第五章 系統測試.....	21
第六章 心得與未來展望.....	26
參考資料	27
附錄	28

圖表目錄

圖2.1 代理伺服器順序圖	8
圖2.2 階層式快取架構圖	9
圖3.1 Squid各主要元件	15
圖3.2 Squid資料流程簡圖	16
圖4.1 Redirector與Proxy Server 架構圖	17
圖4.2 程式流程圖	20
圖5.1 測試網頁(http://kurt.serveftp.org)	22
圖5.2 掛上代理伺服器後的網頁	22
圖5.3 測試網頁(http://www.geocities.com/coffin_1967)	23
圖5.4 原始網頁右上角的廣告已經被檔掉	23
圖5.5 測試網頁	24
圖5.6 掛上代理伺服器後的網頁 (原圖5.5的廣告已經被檔掉)	24
圖5.7 測試網頁	25
圖5.8 無法存取的連結	25

第一章 導論

1.1 動機

隨著網際網路的普及，網際網路(Internet)的使用已經成為現代人資訊取得不可或缺的一環，目前在網際網路中使用率最高的服務當屬全球資訊網(WWW)了，然而因其融合了圖片、聲音、文字、影片、動畫等多媒體特效，以致在傳送時佔用了大量的網路頻寬。

為了減少資料重覆傳送所造成的網路頻寬浪費，所以聰明的人們便想到利用類似電腦架構中快取(cache)的原理，在使用者的近端設立一個代理伺服器(Proxy Server)，將使用者曾瀏覽過的網頁多留一份備份在代理伺服器中，如果之後有人要瀏覽相同的網頁時，代理伺服器就將先前所留下的備份網頁提供給使用者瀏覽，而不必再千里迢迢的到遠端的網頁伺服器上去抓取資料。

網際網路雖然加速了資訊的傳遞，然而人們卻也因龐大卻無用的資訊為苦。例如當我們在使用瀏覽器時，常會出現一些廣告視窗，廣告橫條，造成瀏覽上的不方便，若能自動阻擋這些令人討厭的垃圾資訊，將是使用者所樂見的。目前已經有許多這方面的軟體可用，但如果能和目前廣泛使用的代理伺服器做結合，則在此代理伺服器下的所有使用者皆能享有同樣的功能，不失為一種好的方法。

1.2 目的

代理伺服器可用來協助此代理伺服器後方連接到網際網路。例如一台在區網內的電腦要去瀏覽全球資訊網時，無法直接連結，必須在瀏覽器的設定中加上代理伺服器，藉由代理伺服器代為取得網頁。

如果在一個區域網路中，所有在此網路中的電腦均需掛上代理伺服器才能上網，則網路管理人員將可輕易設定某些禁止存取的網址。因此我們將代理伺服器中加入此類功能，凡是在此代理伺服器之下的

電腦所要存取的網頁，將被受到限制，依此達到阻決廣告的功能。本專題將結合 Squid 代理伺服器，利用Squid本身提供的重導機制 (Redirector)的功能來達到此目的。

目前關於阻擋一些不當網站的方式，多半是檔IP位址，然而有些網站本身不是色情網站，但卻有部分色情內容，若只是限制這個IP的存取，則整個網站皆無法瀏覽；我們解決的方法是只限制不法的網頁目錄，而原本正常的網頁內容依然可以正常瀏覽。

第二章 Proxy Server 概述

2.1 起源

九0年代是網際網路大放異彩的年代，尤其是全球資訊網(World Wild Web)的出現更使得人與人之間的距離越來越小，但是圖形、影像和聲音的劇增對原本速度就不快的網際網路來說無疑是雪上加霜，因此代理伺服器(Proxy Server)便因此產生，其目的是為減輕網路負載，也可加快使用者端網頁瀏覽的速度。

2.2 優點

使用代理伺服器的優點：

1. 節省頻寬

如果在區域網路的聯外節點上佈署代理伺服器，則區域網路裡面的要求能夠馬上透過近端代理伺服器取得資料，不必千里迢迢跑到原始網站去取得，如此可以減少連外所佔用的頻寬。

2. 改善回應時間

資料能夠馬上透過近端代理伺服器取時，網頁存取的回應時間自然大有改進。

3. 減少來源伺服器負載

透過近端代理伺服器取得資料，不必到原始網站去取得，如此可減少來源伺服器的負載。

2.3 運作原理

代理伺服器接受使用者的要求之後會先檢查本身有沒有一份使用者要的資料，如果沒有，則幫使用者到目的地去擷取一份傳給使用者，

並存一份在代理伺服器的硬碟中。而同樣的下一個使用者來要求相同的資料時，代理伺服器會一樣先檢查本身有沒有一份使用者要的資料，如果有的話，則再檢查與目的端的資料是否相符，若相符，則由代理伺服器直接給要求的使用者端即可。

底下的圖形表示client,proxy server,origin server之間互動的場景。

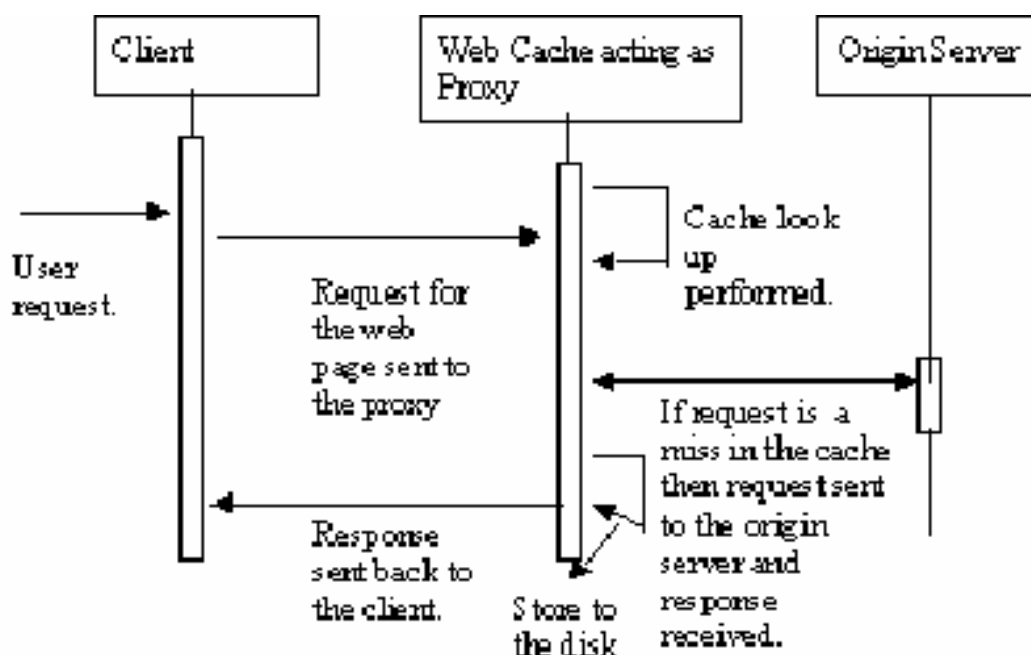


圖 2.1 代理伺服器順序圖。

代理伺服器和代理伺服器之間亦可互相溝通，以取得快取的訊，國際網路上已經定義 Internet Cache Protocol (ICP) 用來規範網頁快取的通訊 ICP 是一種訊息的格式，用於網頁快取的通訊，ICP 可以交換相鄰代理伺服器是否有某些 URL 的訊息，代理伺服器藉由交換 ICP 詢問 (Query) 和回覆 (Reply) 收集資訊，選擇最適合的位址來查詢快取的資料。

此外代理伺服器之間形成一個Cache Hierarchy，即為階層式快取，通常要求文件的流程是由階層架構由下而上。如果代理伺服器沒有所要的文件時，它會藉由 ICP 詢問相鄰的代理伺服器是否有，如果有任一部相鄰的伺服器有所要的文件，則這部伺服器會從相鄰的伺

服器取得文件;如果所有相鄰的伺服器都沒有,則快取伺服器則會往上會要求其上層代理伺服器,若上層代理伺服器沒有所要的文件時,則上層代理伺服器代為取得所要的文件,並傳回給底層代理伺服器。

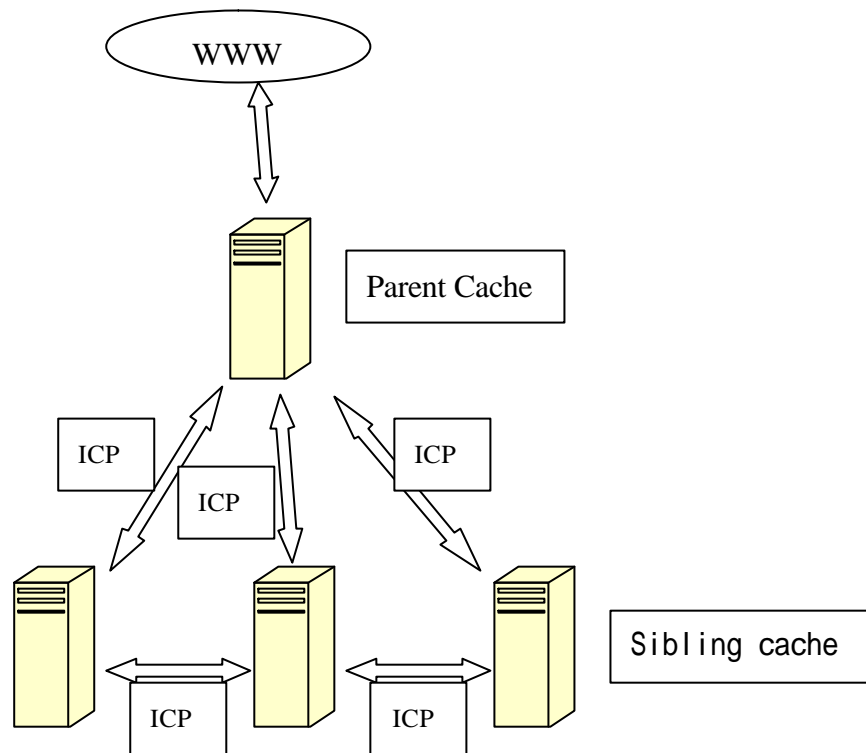


圖 2.2 階層式快取架構圖

2.4 squid 安裝

1. 新增使用者squid,其home directory為/usr/local/squid/
login as squid;
mkdir -p /usr/local/squid/src;
cd /usr/local/squid/src

2. 編譯程式
tar zxvf squid-1.1.21-src.tar.gz
cd /usr/local/squid/src/squid-1.1.21/
./configure --prefix=/some/other/directory
make; make install

3. squid.conf 的基本設定
其中比較重要的有下面幾項參數：

http_port:
瀏覽器連接proxy時,所使用的port.
#http_port 3128
http_port 8080

cache_swap:
/usr/local/squid/cache可以有多大,也就是指可以在硬碟中存放多少MB的東西.
#cache_swap 100
cache_swap 1024

cache_mem :
使用記憶體的上限,以 MBytes 為單位,請注意 伺服器 memory 的大小 EX: cache_mem 8 ps: 請注意這個數值不要設太大,部份 squid 使用的記憶體會隨使用的 cache_swap 硬碟空間而增加,一般說來約 1G 硬碟需要 7MB 的 memory 來建立 cache object 的 index table
而 cache_mem 是指 squid 用來將 cache object 存放在記憶體所用的空間 (將 cache object 存在 memory 是為了避免 open/close file 的 overhead).

整個 squid 所用的空間至少是 $\text{cache_mem} + \text{cache_swap}/\text{GB} * 7$
以 $\text{cache_mem}=16$, cache_dir 共佔 2GB 來看的話, squid 會用掉
 $16+2*7=30$ (MB), 所以你的系統會有 30MB 的記憶體要分給 squid
來用

cache_dir:

規劃CACHE目錄, 預設值是

```
cache_dir /var/spool/squid 100 16 256
```

也就是在 /var/spool/squid 中增加16x256個目錄, 而SQUID 就利用這4096個目錄來存放 100MB 的 CACHE 資料。而在實際的應用上, 100MB 的空間實在太少了! 所以我們可以依 SERVER 上的空間來調整 CACHE 的大小, 另外 16x256 這個目錄大小我們也可依 server 上的實體記憶體和 cache 目錄大小來調整, 原則上, 除非記憶體大於 256 MB, CACHE 的目錄也需要到 10 GB 以上, 否則 16x256 或 64x256 就可以了。

Cache_effective_user:

squid 執行的身份, 請勿使用 root (安全性的考量)

```
#                               UID      GID
#cache_effective_user nobody nogroup
cache_effective_user squid squid
```

acl:

設定Defining an Access List

```
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 allowed_domain ==>設定自己的網域
```

http_access:

```
http_access deny manager all
http_access allow allowed_domain
```

4. 執行squid前要先執行

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

接著執行 /usr/local/squid/bin/squid &
此時squid已經正式運作了

5. `/usr/local/squid/logs/cache.log` 看看有沒有錯誤訊息，
如果有請馬上更改 `/usr/local/squid/etc/squid.conf`，然後行
`kill -HUP `cat /usr/local/squid/logs/squid.pid`` 或者是
`squid -k reconfigure`來重讀 `squid.conf`.
若要關機時，便啟動，請在 `/etc/rc.local` 中加入

```
if [ -f /usr/local/squid/bin/squid ] ; then
    echo -n ' Squid...'
    (/usr/local/squid/bin/squid &)
fi
```

此外 Squid 所進行的每一項動作會記錄在 `/usr/local/squid/logs/` 內，其中

`access.log`: 記錄 client 端的 request

`store.log`: 記錄 squid 將哪些東西存入硬碟

`cache.log`: 記錄 squid 的執行狀況

如果修改了 `squid.conf`，又不想停掉正在執行的 squid，可以透過

`/usr/local/sbin/squid -k reconfigure`

讓 squid 重讀設定檔

第三章 Squid 概述

3.1 簡介

特點：

1. 目前最多人使用的代理伺服器。
2. 支援HTTP、FTP、gopher、WAIS通訊協定。
3. 對於使用者的要求採取 non-blocking 的處理方式。
4. 可支援網域名稱的快取用以加快網址及其IP位址的轉換。
5. 支援SSL網頁的代理功能。
6. 不同 Squid Proxy Server可互相交換 Cache 中的資料，在代理伺服器相互之間溝通的協定中，支援ICP、HTCP、CARP、WCCP。

Squid Internet Object Cache (Harvest Project的後續版本) 是美國政府大力助的一項研究計畫，其目的為解決網路頻寬不足的問題。它是一個高效能的Proxy Server，支援ftp(檔案傳輸協定)，gopher和http(超文件傳輸協定)。它是一個使用non-blocking I/O 的 single process server。squid 的目前版本是由網路上許多人一起努力的結果。能夠在任何 Unix作業系統工作，並且已經知道能在下列的平臺成功運作：Linux，FreeBSD，SunOS/Solaris, AIX 和 OS/2，是現在Unix系統上使用者最多功能也最完整的一套軟體，目前台灣三大網Tanet、Hinet和Seednet有超過95%以上的Proxy Server使用。

3.2 各主要元件功能

解開Squid的原始程式碼，將會發現為數不少的檔案，其中相關連的函式通常放在同一個.c檔之中，底下大致介紹Squid所包含的各項元件：

1. 伺服器端(Server Side)：主要接受新的連線要求，分析，和處理，在這裡決定是Cache HIT，REFRESH，MISS等。對於每個連線的狀態資訊儲存於一個ConnStateData資料。每個request狀態的資訊儲存於clientHttpRequest結構中。

2. 客戶端(Client Side)：根據使用的傳輸協定,這部分的常式負責把判定為Cache Miss的要求傳送到其他服務器。如果是要求於其它proxy則使用http超文件傳輸協定。Cache Miss的要求可能是對Original Server起源服務器或者對另一個proxy。
3. 儲存管理員(Storage Manager)：這部分是squid架構中最重要的。同時也是伺服器端和客戶端之間的中介人。每一個儲存於Cache之中的物件皆有一個StoreEntry結構。這個存儲管理員在記憶體中為cache object存放一份object map(hash table),如此squid才能夠快速找到cache object。此外這個存儲管理員也使用一個存儲策略來追蹤和管理這些快取物件,並決定保留或是移除快取物件來減少儲存空間。在squid中可能有多個client透過單一StoreEntry來接收資料。每個client皆對應一個StoreEntry,當新資料到達時,它才能夠被通知以接收資料。
4. 存取控制(Access Control)：存取控制常式負責控制權的檢查,依據一些不同參數來允許或者拒絕服務要求,這些參數可以可以是客戶端的IP address,目的網站的hostname等。
5. 網路通訊：管理sockets的open,close,read,write,並決定使用TCP或UDP等相關網路通訊工作。
6. 檔案/磁碟輸入輸出：這些routines管理磁碟檔案的讀寫。在squid架構中,網路I/O和磁碟I/O分離,。磁碟I/O中每依次寫入要求不一定發生write callback。
7. 快取管理員：這提供到cache資訊的管理。它只唯讀存取的方式。
8. 重導機制(Redirector)：算是一個外部處理程序,主要從stdin讀取資料並將處理後的結果寫到stdout。在檢查存取控制權以後,但是在檢查cache hit以前,client端的request可以被此redirect程式處理,即 redirector可用來改寫client端的request。本專題就是利用redirector這個外部處理程序以提供一些應用功能。
9. 記憶體管理員(Memory Manager)：它配置和管理記憶中經常被使用的資料架構。當設定memory_pools這個架構時,原本要被記憶體free的將被保留。在一個大的process時,記憶體將更有效地被使

用。

- 10 . 持續的伺服器連結(Persistent server connections): 保持和 origin servers and neighbour caches之間的連線。
- 11 . Filedescriptor 管理: 這紀錄目前使用中的數個file descriptor, 和已經從file descrir讀取或寫入的位元組數目。
- 12 . 除錯函式: 可設定層級除錯追蹤程式。
- 13 . 認證架構(Authentication Framework): 負責處理一些需要通過認證使用的網頁需求。
- 14 . 節點選擇(Peer Selection): 依照設定負責選擇使用最佳相鄰的快取伺服器。
- 15 . 網域名稱快取(IP/FQDN Cache): 網域名稱與IP位址的查詢快取。

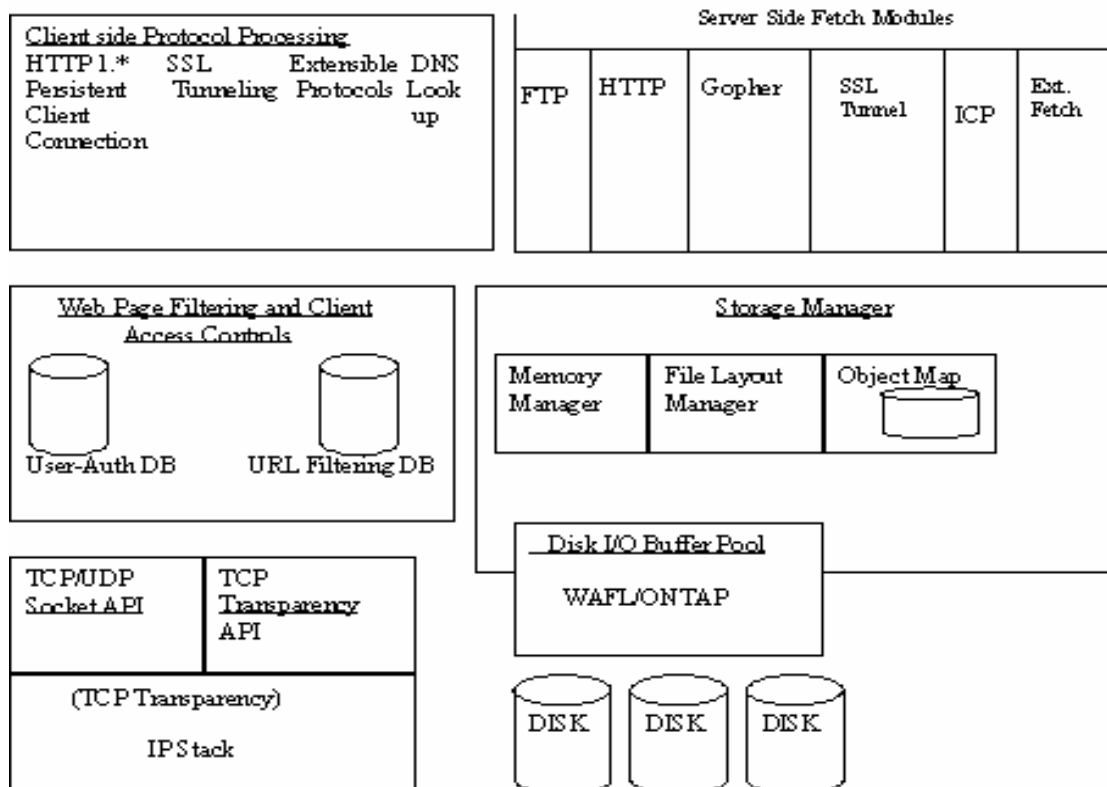


圖3.1 squid各主要元件

3.3 程式流程

Squid是一個single process, non-blocking、I/O driven的程式,所有的要求都是經由一個process來處理,整個程式重複執行一個迴圈,如下所示,經由迴圈內的select()或poll()函式及callback function來處理多個client request.callback function藉由指標在函式間傳遞,當某些function結束時,就呼叫執行這些callbacks。

```
main()
{
    while(1);
    {
        poll();
    }
}
```

程式一開始執行一連串的初始化動作:設定檔分析、配置、測試等動作,之後進入無窮迴圈,等待I/O事件發生(可使用poll()或select()方式),在程式執行期間還可直接接受更改設定(reconfigure),也可接受關閉(shutdown)指令終止程式。

每一個使用者要求皆對應一StoreEntry來接收資料(可能有多個使用者要求透過單一StoreEntry來接收資料) 每一筆快取資料主要是由StoreEntry這個資料結構負責記錄,當一開始使用時,在記憶體的资料由MemObject負責,但如果符合被取代條件,這個資料結構則會被釋放掉。當新資料到達時,Server-side的資料透過StoreEntry結構傳到Client-side。底下是squid資料流程簡圖。

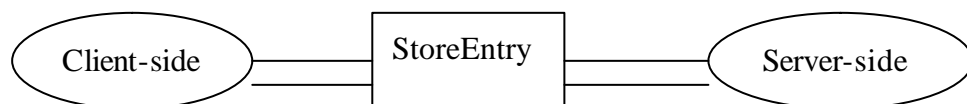


圖3.2 squid資料流程簡圖

第四章 系統製作

4.1 重導機制

現在利用重導機制 (Redirector) 這個外部功能模組來製作廣告阻擋的功能,採用對字串處理功能最快速好用的Perl,將要阻擋的banner, javascript的URL列於一個資料檔中,可使用Regular Expression,利用Perl來比對資料檔與Client端傳送給Squid的URL,假如Client端傳送給Squid的URL與存在於資料檔中的廣告網址相符,這個URL將被改寫成一個空的網頁並傳回給Squid,如此這些廣告網址就不被Squid所下載而達到過濾的功能。

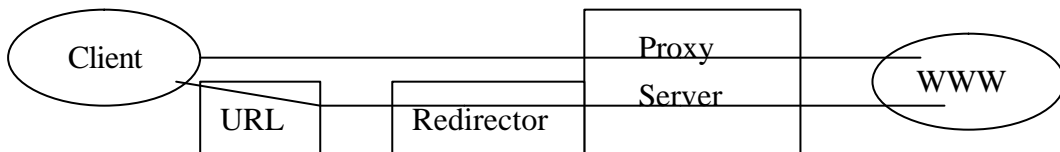


圖4.1 Redirector與Proxy Server 架構圖

4.2 代理伺服器設定

此法對於在一些機關團體學校等,可對一些含有不當內容的網頁(如色情、暴力等)進行過濾篩選。對於任何一個瀏覽器的使用者都必須經過代理伺服器之過濾。使用者可設定以指定之代理伺服器連線或可能沒有設定代理伺服器。如果沒有設定使用代理伺服器者將如何被強制導向代理伺服器接受過濾呢?

第一種方式是透過在網路節點的透通快取(transparent caching)。所謂透通快取就是說：使用者端的瀏覽器或網頁伺服器根

本不用做額外的設定，即可使用快取伺服器，主要原理是由支援網路傳輸層的路由器或網頁交換器等設備，將使用者的網頁需求導向到快取伺服器，在使用上，使用者將以為直接跟網頁伺服器連結。如cisco router以route-map的access control方式 redirect到proxy server，即支援Web Cache Communication Protocol (WCCP)的router可以強制將使用者連線導向一個proxy server。另一種方式是利用DNS Server可以簡單地將使用者連線導向一個proxy server。

4.3 使用工具介紹

Perl(Practical Extraction and Report Language)，是一個直譯式的高階程式語言，由 Larry Wall 所設計發展。發展迄今已變成了網路上最常用的 script語言，而且大多數的 CGI 程式也是使用 Perl 所寫的，而除此之外，它也是許多 UNIX系統管理者的最愛。Perl 雖然是源起於 UNIX，但是我們在其他系統還是可以使用，這是因為它是直譯式的語言，所以可以很輕易地移植在一些系統。Windows 使用者也可以使用，不過它的執行方式就比較不同了。

Perl 起初最主要的功能是以實用性為主，對於文字檔的強大處理能力，可以讀入文字檔後分析裡面的資料後印出所需要的資訊。Perl 結合了 C, sed, awk 以及 sh 的一些好的功能，Perl 的描述語法與 C 非常類似，除此之外，Perl 有個好處，只要你有足夠的記憶體，否則 Perl 並不會限制你資料量的大小，它將一整個檔都視為一組字串，它的遞迴也不限制長度。而除了對文字檔，它對二元(binary file)也能夠做處理，對於 dbm 檔案也能以雜湊表的方式來做處理，所以有不少人會喜歡使用它來做一些檔案處理工作或是管理系統的日常工作。

Perl 最強大的功能是以處理字串的能力聞名的，除了它的 scalar variable 可以很輕易的存取字串，特別的是 Perl 提供和 UNIX 系統很相近的 Regular Expressions語法，這種格式辨認 (pattern-matching)的語法很複雜但十分強大，大大地增強了 Perl 處理字串的能力。

正由於Perl的特性，容易處理字串參數的傳遞，自然成了某些特殊用途的新寵，尤其是CGI，事實上CGI本身定義是和使用的程式語言種

類無關的。那為什麼大家現在都採用Perl呢？原因有下面幾點：

- Perl有非常傑出的文字處理能力，能輕易地產生HTML文件，尤其它具有完整的Regular Expression語法，使得在C語言看來必需要寫一大串程式碼的功能，Perl幾行就解決了。
- Perl是一種直譯式語言，因此可以避免在各種不同系統上，還需要重新編譯的麻煩。
- Perl的檢查旗標(-T)可以保護你的Web Server，避免被未經授權的client破壞。

4.4 系統製作

首先修改squid.conf之中的redirect_program欄位指向程式所在目錄,如: redirect_program /path/to/redirector.pl

將要過濾掉的網頁網址列於資料檔之中,如:

http://*.adver.com.tw/

http://kurt.serveftp.org/test.html

整個程式(此程式與資料檔列於附錄中)是一個無窮迴圈,不斷從標準輸入讀入使用者端傳送給Squid的URL,然後依序比對此URL與資料檔,若相符的話,就把此URL改成一個空的網頁,如此Squid就不會去下載這個要過濾掉的網頁。圖示如下:

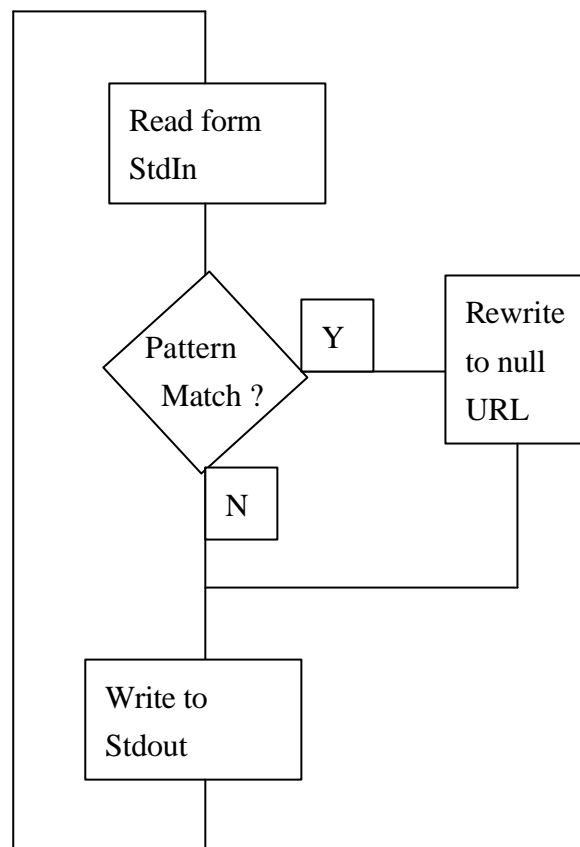


圖4.2 程式流程圖

第五章 系統測試

下頁幾張圖分別表示原始網頁與掛上代理伺服器之後的情形,例如圖5.1是一個原始的測試網頁,圖5.2表示掛上代理伺服器後經過重導機制重新導向另一個網頁,並顯示此網頁被封鎖了

如圖5.5當中,其廣告位址為http://ad*.adver.com.tw/ , 所以將http://*.adver.com.tw/ 加入我們的資料檔之中,則結果如圖5.6所示,相關的廣告橫條已經消失了。

圖5.7本身為一個影音網站,然而其部分網頁內容卻有成人內容,因此我們將這些限制級的網頁導向一個空的網頁,如圖5.8所示。由此可看出此方法只是阻擋URL位址, 所以我們只是阻絕此網域(<http://www.gotowatch.tv/>)底下的部分內容, 若是檔IP位址的話, 則整個網站皆無法瀏覽。

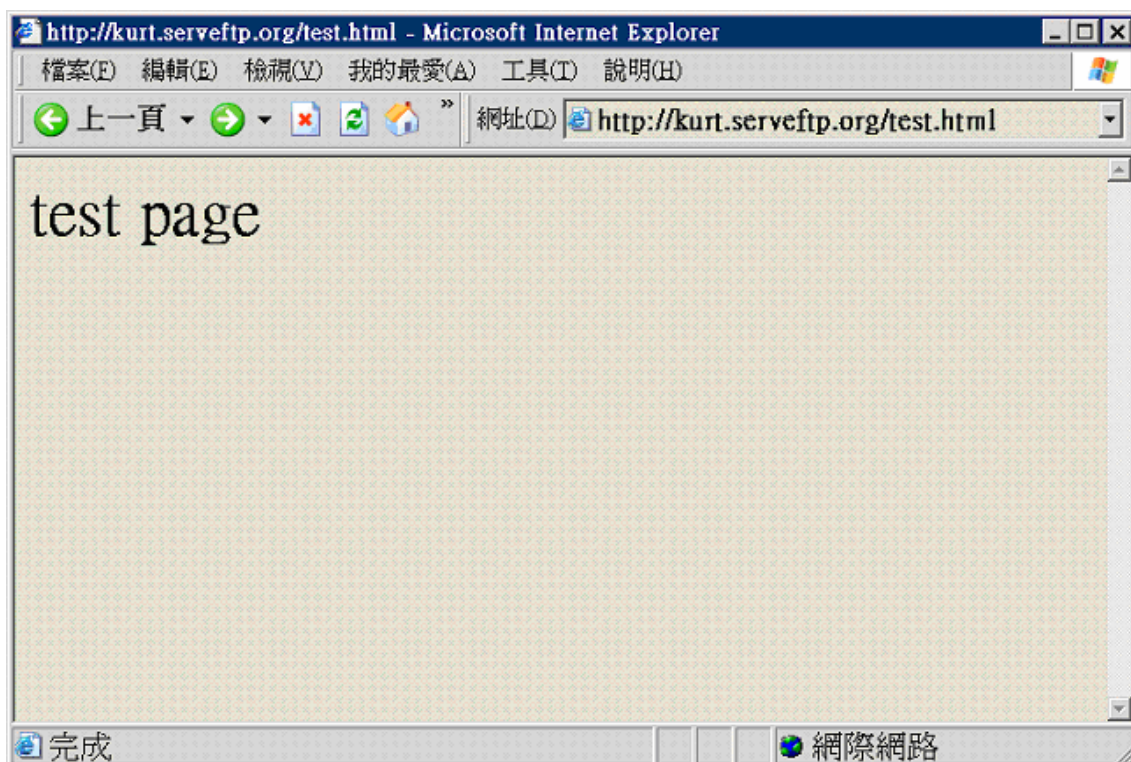


圖5.1 測試網頁(<http://kurt.serveftp.org>)

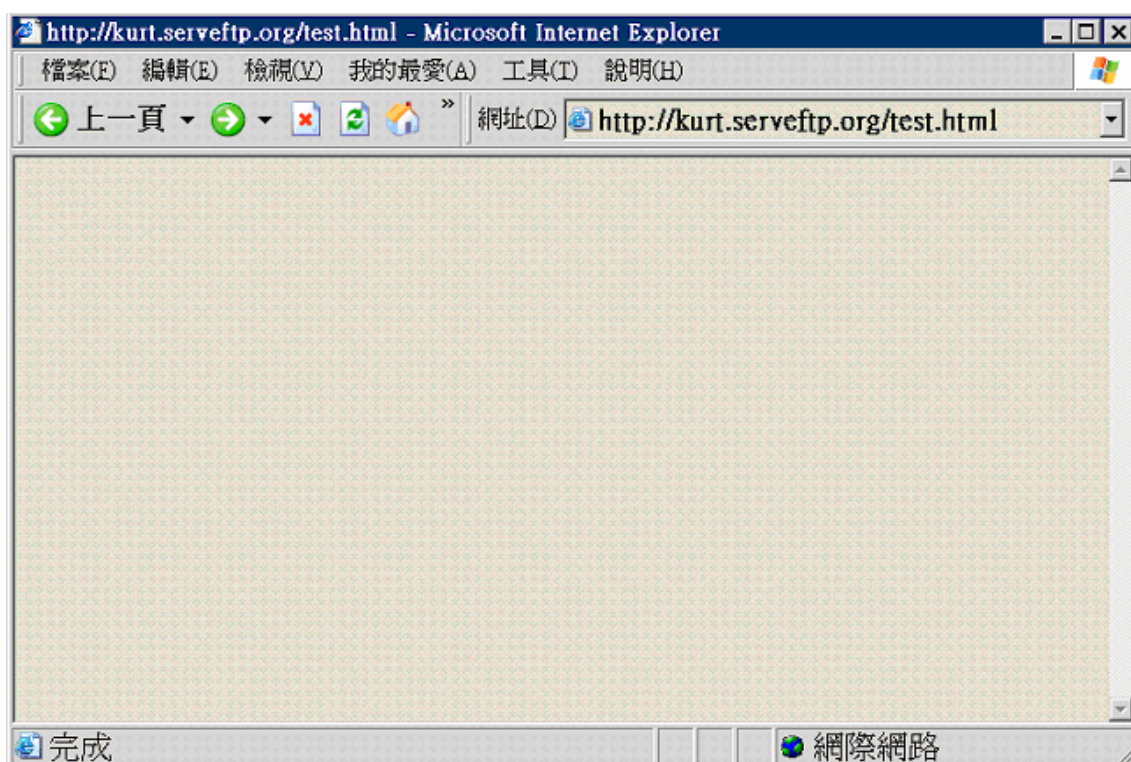


圖5.2 掛上代理伺服器後的網頁

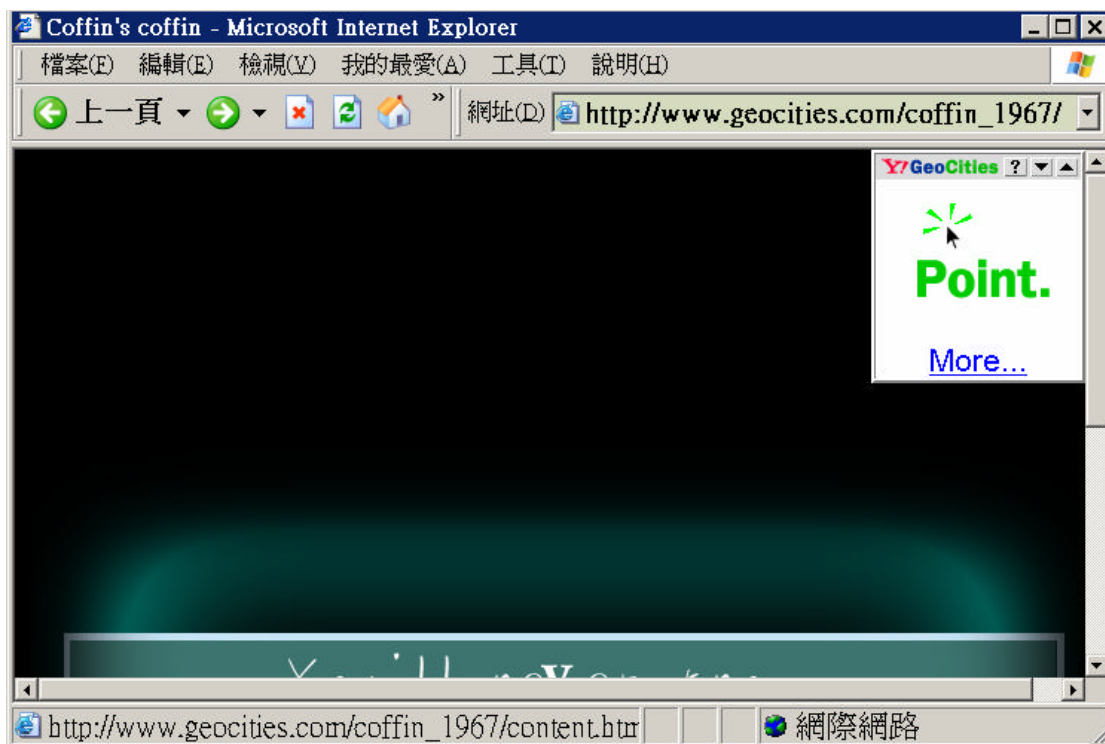


圖5.3 測試網頁(http://www.geocities.com/coffin_1967)

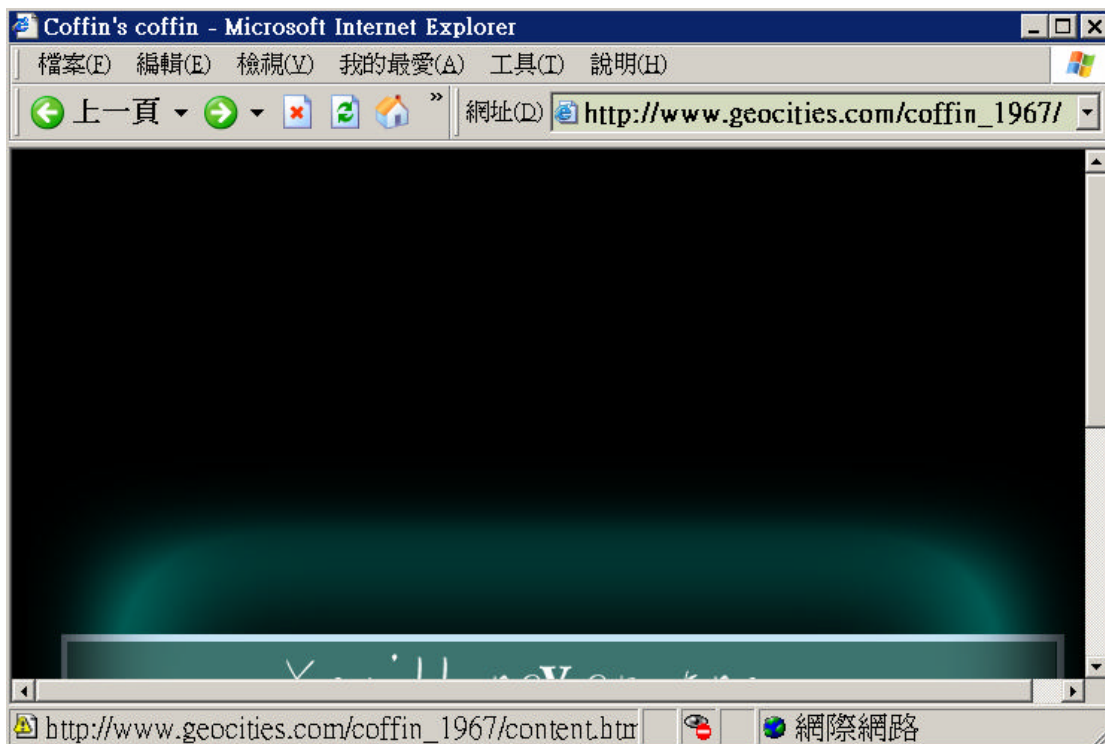


圖5.4(原圖5.3右上角的廣告已經被檔掉)



圖5.5 測試網頁

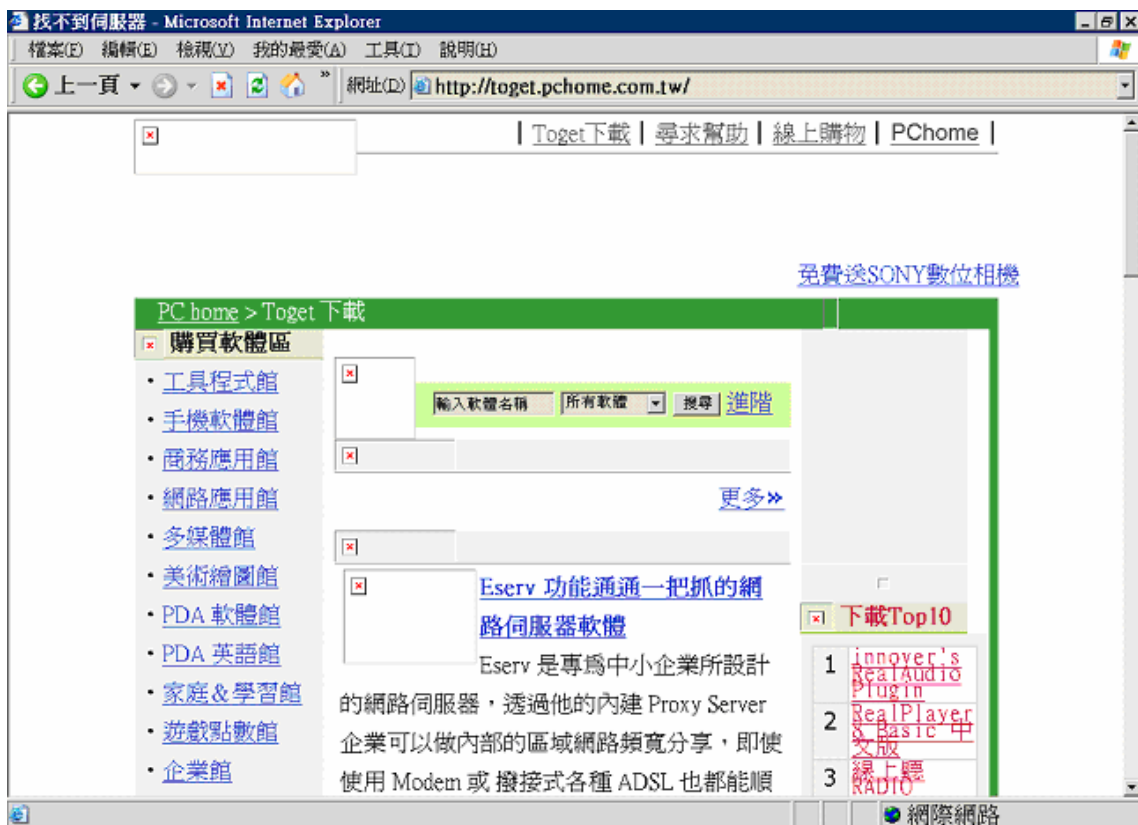


圖5.6 掛上代理伺服器後的網頁(原圖5.5的廣告已經被檔掉)



圖5.7 測試網頁

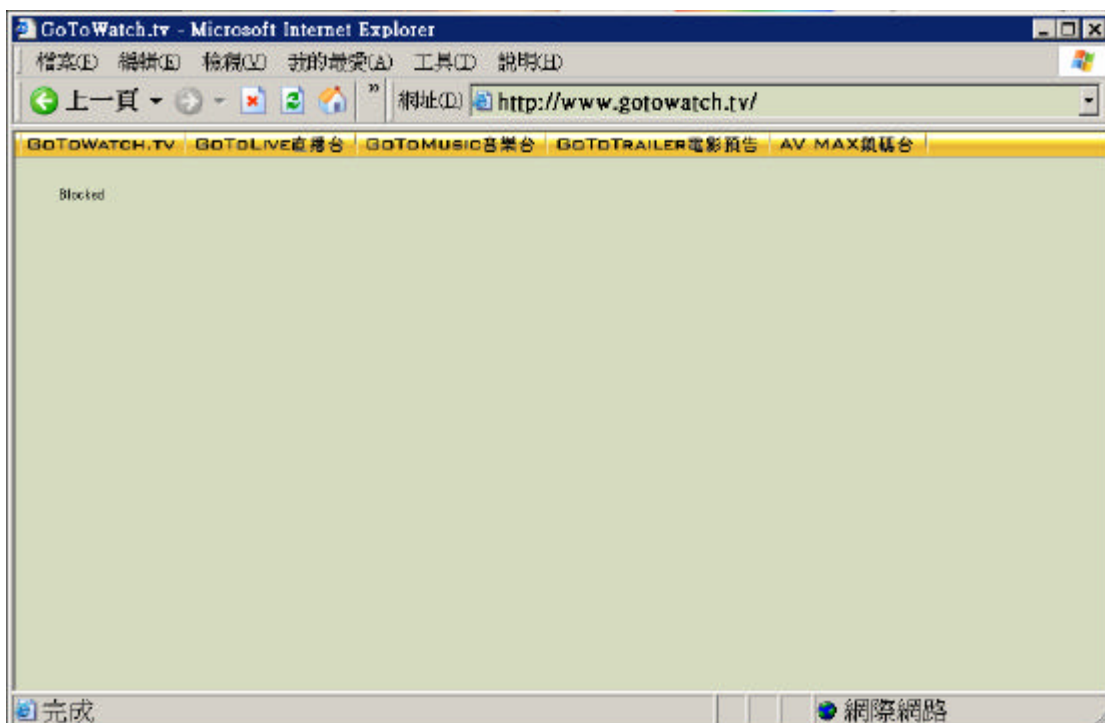


圖5.8 無法存取的連結(原圖5.7右邊的本週推薦連結無法存取)

第六章 心得與未來展望

一開始會選這個題目,主要是對一個代理伺服器的運作方式蠻有興趣的,經過了一番的摸索與資料的找尋之後,對Squid這個程式,大致上有了一點概念,但對原來的目標感到超出我所能掌握,經過修正之後改成目前的一個應用。日後若想對此系統做一些功能上的改進,我想可以做一些web形式的管理介面,如此方便把一些新出現的廣告網址加入拒絕存取清單之中。另外可能有一個缺點,若清單內容有上百筆資料時,每一個使用者要求就須經過上百次的比對,對於一個大型的應用來說,可能會拖慢代理伺服器的速度。

參考資料：

- [1] Squid Internet Object Cache:
<http://www.squid-cache.org/>
- [2] Squid Programming-Guide
<http://www.squid-cache.org/Prog-Guide/prog-guide.html>
- [3] David A. Curry原著,蕭伯剛編譯, UNIX系統程式設計, 松格, 1997
- [4] Martin Brown著, 陳俊銘等譯, Perl完全程式設計手冊,
美商麥格羅 希爾, 2000
- [5] Douglas E. Comer and David L. Stevens, Internetworking with
TCP/IP volume III Client-Server Programming and
Applications, BSD Socket Version with ANSI C, 2nd edition,
Prentice-Hall, 1996.

附錄：

程式：

```
#!/bin/perl

while (<>) {
    my @input=split;
    my $url=$input[0];
    my $newurl='';
    my $log='/proj/proj.log';

    my $server= ' http://kurt.serveftp.org ' ;
    (@list,@pop,@frame)=((()),(),());
    my $DATA= ' /proj ' ;
    my $listfile='list.dat';
    my $popfile= ' pop.dat';
    my $framefile='frame.dat' ;

    # Read the list of URLs

    open(FILE,$DATA.$listfile) || die("Can't open $file");
    while(<FILE>) {
        chop;
        push(@list,$_) if($_ && !(/^#/));
    }
    close(FILE);

    open(FILE,$DATA.$popfile) || die("Can't open $file");
    while(<FILE>) {
        chop;
        push(@pop,$_) if($_ && !(/^#/));
    }
    close(FILE);

    open(FILE,$DATA.$framefile) || die("Can't open $file");
```

```

while(<FILE>) {
    chop;
    push(@frame,"$1\t$2") if($_ && !(/^#/))
    &&/^(\S+)\s+(\S+)$/);
}
close(FILE);

```

Pattern Matching

```

foreach my $file (@frame) {
    my($x,$r)=split /\t/, $z;
    if(($x=~/(.*)\*(.*)/ ? (index($url,$1)+1 &&
index($url,$2)+1) : index($url,$x)+1)) {
        $newurl=$server.$r;
        $islog=$x;
        last;
    }
}

foreach my $file (@pop) {
    if(($x=~/(.*)\*(.*)/ ? (index($url,$1)+1 &&
index($url,$2)+1) : index($url,$x)+1)) {
        $newurl=$server.'pop.html';
        $islog=$x;
        last;
    }
}

foreach my $file (@lists) {
    if(($x=~/(.*)\*(.*)/ ? (index($url,$1)+1 &&
index($url,$2)+1) : index($url,$x)+1)) {
        $newurl= ' ' ;
        $islog=$x;
        last;
    }
}

```

```
$newurl=$url unless($newurl);

# Logging to proj.log

if($islog){
    open(LOG,">>$log");
    print(LOG "$_[0] matches $_[1]\n");
    close LOG;
}

# write new URL to stdout
print("$newurl\n");

}
```

資料檔:

<http://www.gotowatch.tv/class.asp?cctpct=4>
http://www.gotowatch.tv/product.asp?cctpct=4&progid=*
http://*.adver.com.tw/
<http://kurt.serveftp.org/test.html>
<http://ads.>
<http://ad.>
<http://adserver.>
<http://k5ads.>
[/cgi-bin/nph-adclick.exe](#)
[/Ads/Media/Images/](#)
[/SmartBanner/](#)
[.com/ads/banners/](#)
[/apfbanners/](#)
[/RealMedia/ads/](#)
[/event.ng/](#)
[/pics/banner/](#)
[/viewCGI?pool=](#)
[/ClickTrack](#)
[/HitTrack](#)
[/images.go2net.com/go2net/ads/](#)
[doubleclick.net/ad/](#)
[doubleclick.net/adi/](#)
[doubleclick.net/viewad/](#)
[ad2.doubleclick.net/](#)
[m.doubleclick.net/](#)
[ln.doubleclick.net/](#)
[adfu.blockstackers.com/servfu.pl](#)
[209.207.224.220/servfu.pl](#)
[smartage.com/ads/](#)
[www.bravenet.com/affiliate/affiliate.php?](#)
[adforce.imgis.com](#)
[imgis.com/images](#)
[view.avenuea.com/view](#)
[gm.preferences.com/image](#)
[204.152.166.208](#)
[www.ad.tomshardware.com/cgi-bin/bannerdisplay](#)

st.valueclick.com
Ogilvy.ngadcenter.net/ads
www.appleinsider.com/banners
www.canoe.ca/AdsCanoe
wdvl.internet.com/ads
webfarm.sfo.mediaplex.com/banner_request
bannervip.webjump.com/ads
bannervip.webjump.com/webjump/valet/images
was.ads.link4ads.com
www.maximumcash.com/cgi-bin/maxcash.cgi
www.babylon-x.com/cgi-bin/click
macnn.com/media
macnn.com/gphs
www.insanely-great.com/banners
img.cmpnet.com/ads/graphics
209.185.222.45/creatives
ad1.mamma.com
www.burstnet.com/gifs/
209.249.169.51/banners
www.teknosurf3.com/cgi-bin/ads.pl
adimages.go.com
thumper.hlservers.com/cgi-bin/network/banner.cgi
dizzy.hlservers.com/banners
travel.pagecount.com/travel/counter.gif
edu.pagecount.com/edu/counter.gif
admedia.xoom.com/Banners
zdads.e-media.com
avenuea.com/Banners
bilbo.counted.com
images.slashdot.org/banner/
adimage.blm.net/
www.starsads.com/cgi-bin/serve.cgi
/images.cnn.com/ads/
/cnfn.cnn.com/ads/
.yimg.com/us.yimg.com/a/pr/promo/
/us.yimg.com/a/
a32.g.a.yimg.com
/jeeves.flycast.com/rich/

ad-flow.com/
/ias.hitbox.com/
.adbureau.net/
global.msads.net/
www.namezero.com/img/ads/
altfarm.mediaplex.com/ad/
/media.admonitor.net/creatives/
usatoday.com/sponsors
wetteronline.de/werbung
mds.centreport.net
ads1.condenet.com
ads2.condenet.com
gezondheidsplein.nl/adimages
gezondheidsnet.nl/images/banner
web.planet.nl/adv
leadtrace.net/ad.asp
wemnet.nl/ads
altavista.com/static/ave/ads
ilse.nl/icons/adverts
webads.nl/ads
worldonline.nl/ads
hardwareinfo.net/ads
/image.atdmt.com/Banners/
qfc.com/pub_mcc/
.com/banners/
.com/httpads/