Cache: *n., 1: a hiding place. 2: a secret store*

Ivan Trundle

Manager, systems and publishing
ivan.trundle@alia.org.au

The other day I was asked why ALIAnet, the website, does not bother to have counters anywhere across the plethora of pages offered. The answer is very simple (they don't work), but the question did lead me to delve more deeply into the reason why: caching. It also led me to consider the issue of copyright, which is mercilessly sandwiched in the middle of it all.

Have you noticed how slowly the first page you visit from ALIAnet loads? Have you noticed how much faster subsequent pages load? And have you noticed that when you revisit the site a short time later it loads infinitely faster, and yet a page from a site that you have never visited before loads much slower than anticipated, even though it may have no graphics to slow it down?

This activity is all a direct result of caching, either on your own computer or elsewhere, or a combination of both. Caching is a weapon used to battle one of the biggest enemies known to the internet: lack of bandwidth. So how does it work? Every website address specifies or implies a unique reference point, and opening a web browser and entering the address will force a request to be made to the web server containing that URL. That request is in the form of 'please send me a copy of the page in question so that I can display it on screen at this end' — and if the host computer (web server) is happy to comply, a copy is sent to the client computer for display and subsequent storage. At this stage, those with concerns over copyright are probably apoplectic — but we will press on for now.

Client caching

The client machine's web browser is generally set (by default) to store a copy of the page for that entire 'session' — defined as the period in which the browser application is running. On the other hand, it is entirely possible for the browser to be set to store the page indefinitely (even through restarts and exits/quits — and thus not requiring any further transmissions), or to not store the page at all (requiring a full retransmission of the page at a subsequent request).

The advantage in storing a page throughout a session is obvious: there is generally no need to download an entire page and its contents more than once every session, unless the page in question changes frequently. For example the ALIAnet 'what's new' page changes frequently, and so to assist those who visit the page, there is code attached within the page which forces a reload of the page every 1044 seconds (let's not get sidetracked into asking 'why this number?'). However, most website pages do not contain code to reload or refresh the data content.

By not forcing a reload, bandwidth is preserved for all users since less requests are made to transmit data repeatedly across a finite resource — the 'pipes' of the internet.

Proxy server caching

There is another form of caching which demonstrates why web access counters are practically meaningless — proxy caching. Proxy server caching is becoming popular as a form of preserving bandwidth, and is used to minimise data traffic by and within government, corporations, businesses and educational institutions.

Data traffic costs money — internet telecommunications providers charge per byte, usually for incoming data. Here is a typical scenario: the National Library of Australia (NLA) has many members of staff who are ALIA members — and many of them like to keep abreast of what is happening within the Association by browsing ALIAnet's homepage each morning. This could result in a lot of traffic, with subsequent high charges for incoming data, multiplied by the number of people viewing the site.

To minimise costs, and to increase the speed of the download, a proxy server can take the first request and store a copy of the ALIAnet home page locally. Thus, every subsequent request by other NLA staffers will draw upon the recently-stored page sitting on the NLA proxy server, rather than download time-consuming and expensive copies. Congestion of the pipes of the internet is reduced, and caching also decreases the amount of processor cycles required on the host machine, thus allowing the host machine to deliver documents to other users. In short, everyone wins — in theory.

However, some proxy servers do not retrieve the most up-to-date information often enough. Thus, caching causes websites to lose control of their content, and the timeliness of that content. This can have serious ramifications on share-trading or auction sites, less so on more static information sites. It is possible for host server pages to be coded in a way that forces a frequent refresh of data, but this can be abused to the extent that pages are needlessly retransmitted (wasting further bandwidth).

Page counters

Why does caching wreak havoc on page counters? And where can it go wrong? Page counters calculate 'hits' on the basis of the number of requests made for a document, and the page count itself is only incremented locally. If a page is copied and stored off-site (in a proxy server), then the page count becomes meaningless. Ten thousand users may view the page through a proxy server and the original page will not reflect this number. At best, page counters and even log files are highly-dubious 'eyeball' registers.

Copyright infringement — or technological solution?

Then there is the issue of copyright — and what constitutes copyright infringement in this context. Is caching an example of copyright infringement? Almost undeniably so, according to current definitions. Yet caching is a technological solution to a pre-existing technical problem, without which the internet would surely shrivel up and die — or be choked with endless page requests.

In any event, those page counters are doomed. At least until a technological solution presents itself that can aggregate proxy server 'hits' with real 'hits', at least... ■

Caching is a weapon used to battle one of the biggest enemies known to the internet: lack of bandwidth...