Cache Counting and Busting

A little over a week ago, I wrote posted an off-the-cuff explanation to this list about how Imgis & MatchLogic supposedly get around the dilemma of cached ad images without resorting to "cache busting," to which Glenn Fleishman replied, questioning the technical accuracy of my explanation. I realized afterwards he was right, that my explanation was flawed, but I didn't have time to reply till now. Meanwhile, I sent several messages to folks at MatchLogic, asking them to address the question directly, but for whatever reason, they didn't. So, I recently interviewed several folks at Imgis, Accipiter and NetGravity, and based on those conversations I offer here a detailed explanation for how many sites and networks "cache count" and "cache bust." I also add my own reflections on the implications of these practices. I apologize for this message's length and geekishness (I tried to keep it in layman's terms), but this issue keeps coming up on the online advertising list and I believe it is significant for many reasons for online marketers, so I hope you find it interesting. While I think I've got it right this time, if anyone can correct me on any technical points (politely), please do so. That goes especially for the experts at MatchLogic, Imgis, NetGravity and Accipiter whom I've copied on this message.

To review briefly, caching is where browsers and proxy servers save copies of Web page elements (e.g., page source code, images, etc.) to cut down redundant network traffic and to speed browser loading. That is, a browser will save, for example, a page logo on the end-user's computer in the cache directory so that when he visits another page in the site, the browser will use cached logo rather than downloading a duplicate. Similarly, large networks, notably America Online (and ISPs, big companies and others), use proxy servers to show internal members the same Web page out of the network's cache instead of repeating downloads across the Internet.

The problem for the online ad industry, as we've discussed on the Online-Advertising list before, is that banner ads get cached like other page elements, so publishers lose track of those impressions. They can't get paid for them, they can't target them to individual users, they can't control for frequency of exposures, etc. To get around this, many large sites employ what are popularly known as "cache busting" techniques. The wiser ones use a more sophisticated approach, "cache counting" (I've heard this term used, though it's not as generic yet as "cache busting"). It was my explanation of this latter technique that was flawed in my earlier post. I attempt here to explain the distinction between the two.

Cache busting is an attempt to force a duplicate version of an ad (or page code) across the network for every new user who requests the page. One crude way to do this is to use the HTTP server to set the expiration period for the object (e.g., the ad or the page) to zero seconds. That way, in theory, as soon as the proxy server caches an element, it appears to be out of date, so the proxy thinks it needs to get a new object next time. My understanding is that the proxy administrator can over-ride this by setting the proxy server to check for new objects not sooner than every 20 minutes, for example. On a high-traffic proxy server like AOL's, that could mean a lot of additional ad impressions are served out of cache within that 20 minutes.

Another popular way to bust cache is for the ad server to create a unique name for the ad every time it is requested. This may be done with a time-stamp, for example. When a first user requests a page, the ad server inserts into the page source code that the browser should download an ad named "AutoAd030298080702" (i.e., March 2, 1998, 8:07am and 2 seconds). When the server receives the ad request, it understands which ad to serve. Most proxy servers, the experts tell me, will look for updated page source code more often than for updates of other objects (like ads). So, when the proxy lets the next user download fresh page code, the new source code will refer to the same ad by a different name (because it has been time-stamped five seconds later, for example), forcing the browser to download a new copy of the ad from the ad server, thus letting the server count another impression. There are variations on this technique, but I'll spare you more details.

Cache counting, by contrast, has a few important advantages. First, it is much more ecologically sound, so to speak, as it doesn't spam the network with duplicate copies of the same ad, but rather it manages the targeting of ads out of browser and proxy caches. Second, because it works with caching, instead of against it, it provides a faster experience for the user. Third, it appears to be a more accurate way to count ad impressions, in line with the IAB's summer 1997 guidelines for ad counting terminology (see http://www.iab.net/advertise/metricsource.html).

Here's how it works (*finally* the answer to your question, Glenn!). Unlike cache busting, where the ad server typically inserts an exact name of an ad into the source code as the page passes from the Web server to the browser, the page code in the case of cache counting doesn't refer to a particular ad at all. Instead, it contains simply an instruction for the browser to ask the ad server which ad to insert. (Glenn, that's the part I overlooked in my original description.) As a result, as the browser is constructing the page, it queries the ad server asking which ad it should use, and the ad server sends back a small file with targeted instructions as to which ad should be inserted. The ad server companies refer to this response as a "ping." The ping does a few things. First, per the IAB's ad tracking guidelines, the server counts that request from the browser and the server's ping reply as an ad insertion. In a sense, the ping is asking the browser (and the proxy server, which is monitoring the browsers/ad server dialogue) whether it has the ad in cache. If the browser has it cached, it inserts the ad directly and doesn't bother to reply again to the ad server. If it doesn't, it will request back to the ad server. At that point, the proxy server will either intercept the request and supply the ad if it has it cached or let the request go back to the ad server if it doesn't have it in cache, in which case the ad server supplies the ad. The ping, typically written as a "location redirect" command, cannot be cached, so therefore every ad served using this method can be both counted and targeted to individual users, even though the ad itself may be cached. To test the effectiveness of this technique, go to a site served by Imgis, such as http://www.drlovelady.com/ or http://www.rollingstone.com/ and load a page, then hit the reload/refresh button. Unless they have few ads in rotation, you should get a different ad each time you reload. Try the same thing on a site not using the cache counting technique (i.e., most sites) and you'll normally get the same ad out of your local browser cache every time you reload. If you view the source code of both pages, you'll see how the cache counting technique doesn't refer to a specific ad by name, but the standard page code does.

Representatives of Imgis, Accipiter and NetGravity were eager to point out that MatchLogic did not invent this cache counting methodology with its TrueCount service, it was simply the first to gain a lot of publicity about it by issuing a press release. (And good for MatchLogic! I, like other journalists, wrote a sizable story about the announcement in Advertising Age.) Imgis, with its network approach to ad serving, delivers all of its ads using this cache counting approach, while Accipiter and NetGravity enable it with their technology, but it is up to their client sites to implement it on an individual basis.

Although I realize this posting is already long, I would like to conclude with some observations about the significance of these methods. I first joined this discussion when K2 Design's Tom Hespos speculated what would be the impact if cache busting became widespread in the industry. I believe it is inevitable that it will become widespread, and I am hopeful that cache counting will become the preferred method. While the majority of Web sites do not yet partake in either cache busting or cache counting, my impression from various interviews is that most of the top trafficked sites already do engage in one or the other technique. While cache counting as described above appears to province a more accurate count than cache busting, the more significant difference is between sites that engage in whatever form of cache busting/counting versus those who rely only on their server logs to count impressions. MatchLogic claimed in its TrueCount press release (http://www.matchlogic.com/news/prel_01.htm) that on average 76% more viewers are seeing ads served out of cache than sites can count based on their own Web servers. While many suggest that number is exaggerated, I can believe it's easily 50%, after a VP at Warner Bros told me that their ad impressions went up by 30% after AOL alone started sharing their proxy log stats showing AOL member traffic to the Warner Bros Web site.

Of course, from an average publisher's point of view, more accurate counting of cached impressions isn't all good news. First, except for the top handful of publishers, most sites are not selling out anywhere near their existing ad inventory, even according to just what they can count on their server logs. For the sake of argument, assume that on average 50% more ads are viewed than server logs count. That would mean a site that is selling only 60% of its ad inventory by its own Web server's count is actually selling only 40% of inventory when cached impressions are factored in. The standard laws of economics, given a greater supply and a consistent, underwhelming demand, would seem to entail a negative impact on CPM rates. I don't think, however, it's as bad news as some have suggested, namely that advertisers aren't going to be willing to pay any more for the impressions they had been receiving for free via cached views. That argument says that if an advertiser was paying $20 CPM and now learns that in fact he was getting 1,500 impressions for that price, he'll want to pay only the equivalent $13.33 CPM if publishers start tracking those ads. In fact, the newly revealed cached impressions should have a value closer to the original CPM, as the previous freebie 50% was totally untargeted, whereas using sophisticated cache counting and cache busting techniques, publishers can target those newly identified cached impressions as accurately as normal ads (controlling for frequency of exposures, etc.).

More disturbing is what happens to click-through rates. While ad impression counts may vary greatly due to caching, absolute numbers of click-throughs remain constant, as by definition they don't get cached (most publishers route the click back through their own site then on to the advertiser so that the publisher can track the click-through). So, assume a site relies only on its server logs to count ad impressions and measures 20 click-throughs per 1,000 impressions, or a 2% click-through rate. Yet, if the advertiser serves his own ads with MatchLogic's TrueCount, he can count more accurately than the publisher to include the extra 50% of cached impressions, so the same 20 clicks are now divided by 1,500 impressions, or a 1.3% CTR. From the advertisers' point of view, all of that argues in favor of negotiating CPM rates based on the cost per click he's willing to pay, where direct response is the goal of the ad. I would urge all publishers to move towards the cache counting technique described above. Not only is it better for network bandwidth in the long run, it provides users with a faster loading page and sites with a more accurate measure of impressions. Until the industry at large standardizes on one reliable method of counting, CPM rates are unlikely to stabilize at their natural value with such radical disparities remaining between different counting approaches.

What promises to complicate the valuation of CPM rates even more is the recent introduction of Millward Brown's brand measurement methodologies bundled into Accipiter's ad serving software. For the first time sites will be able to offer advertisers a realistic alternative to quantifying the success of ad campaigns beyond click-through by effectively measuring brand impact (see Accipiter's press release at http://www.accipiter.com/press/releases/pr_milwardb.htm). But that's another story...

Rick E. Bruner   rick@bruner.net
Bruner Communications   tel: 1+(415) 351-2489
Strategic Web mktg consultancy   http://209.197.70.109/
(Site under construction to be relauched early April)