TagMan latency test

(A study into the discrepancies caused by different methods of tracking tag deployment in relation to data quality and page download speeds)

Public report, April 2009

Introduction

TagMan is the single-tag solution to the problems of campaign tracking (seeing the true path to conversion) and slow pages loads that discourage customer acquisition. A single TagMan universal container tag replaces all the tracking pixels on a retailer or e-commerce web page used to track the path to conversion through natural search, paid search, affiliates, display, email and the website.

As part of our ongoing commitment to both technical and marketing best-practice, we undertook a study into how the various methods of collecting data using tags can affect the loading of pages and the quality of data collected.

The purpose of the study was to help companies understand the reason for discrepancies in data reported by third-party tracking solutions and how to avoid them.

Main findings

- The most popular way to deploy conversion tags (via an iFrame container positioned at the bottom of the page body (the solution used by all the major container tag solutions)) is actually the worst for accurately reporting traffic, introducing a 15-60% discrepancy
- The biggest factor in the size of discrepancies between the number of users arriving on a site and the number reported by a page tag was page download time, with approximately 10% of traffic lost for every extra second the site took to load
- Most of the slowest-loading assets on the pages were third-party tracking tags, taking as much as 250ms
- The largest discrepancy between the actual traffic arriving on a website and the figures reported by page tags was 60%, this was comparing a Javascript call at the top of the page head (best) to an image pixel inside an iFrame at the bottom of the page (worst)
- Several tags were found of which were for services that were no longer being used
- Moving the Google Analytics tag from the bottom to the top of the page increased the reported traffic by 20%

Key conclusions

1) Tags really do slow down your page

The slowest-loading assets on site pages in our study were tracking tags, taking as much as 250ms. Based on our study, a delay of around one second causes approximately 10% of users to abandon the page, suggesting that four tags on the page could lose you 10% of your visitors. While the figures would depend on the value attached to the page by the user, such an adverse affect on user experience would cause a significant loss of revenue for online retailers. When we spoke to advertisers about their tag nightmares we found evidence that tags from the major vendors can introduce significant delays to page downloads.

2) Tag position needs careful consideration

If page load speeds are slow then very different figures will be reported depending on where tags are placed and the relative effect of the tag will be far less. In our test, hecklerspray.com achieved a 20% increase in the traffic reported by Google Anayltics when the tag was placed at the top of the page, which would be a good argument to put the code higher up the page.

Clearly, given the impact of page tags on page download times AND accurate reporting, a balance needs to be found between the quality of data collected and the customer experience.

3) Page optimisation is critical

The rate at which users abandon slow pages underlines the need to ensure they are written with a view to loading as quickly as possible. Both sites in the test contained third-party analytics tags that were no longer in use. An easy place to start would be to identify any tags that are no longer in use and remove them. (Links to tools and tips for optimising pages are included in the appendix.)

Marketers should keep a close eye on site monitoring reports as any tag performance issues are likely to affect the quality of their data as well as the number of users who convert from campaigns. They also need to keep track of tags they've put on the site and review periodically so that old tags are removed.

TagMan recommends that controlled tests are performed from time-to-time that intentionally delay page loading and measure abandonment rates at different points in the conversion process.

The results of conducting your own latency test will provide a cast-iron business case for investment in software or hardware solutions that can improve tag management – and thus website performance. It will also allow your business to take an informed decision on where tags should be inserted into the page.

4) IFrame container tags suffer the most

The worst performing method of including tags in the page was via an invisible iFrame at the bottom of the page – the method used by the major container-tag solutions. The tests showed the most effective way to collect data is by using a blank JavaScript call, particularly if the tracking code is placed at the end of the page.

On Askaprice.com, there was a 9% difference between the traffic reported by an image pixel that was coded in the page versus one that was served through an iFrame. However, when the iFrame was at the top of the page the difference was less than 1%.

The discrepancy on HecklerSpray when using iFrames was even more significant. The figures reported for the pixels contained with the iFrame at the bottom of the page were 37% less than for a pixel included directly into the page. This compared to a difference of around 3% when placed at the top of the body.

Given that pixels loaded within iFrames do not block the parent page from loading it would seem wise to move iFrame-based container tag solutions to the top of the page.

Methodology

The study was undertaken during July 2009 using two websites:

- www.askaprice.com – a lead generation website that allows users to suggest a price they are willing to pay for a car, which is affiliated with www.carkeys.co.uk
- www.hecklerspray.com – a celebrity news and gossip website appealing to, predominantly, a US audience

Different types of tags were used at different positions in the page using JavaScript. Some tags were written directly into the page, whilst others were written inside iFrames in order to mimic the way tracking tags are delievered via popular container tag solutions such as the Atlas Universal Action Tag and DoubleClick Floodlight.

The JavaScript code was tested for compatibility with the most popular 24 browsers and operating systems using www.litmusapp.com.

Browsers have a limit of 2 concurrent requests per domain. To prevent the tests blocking each other the tags were put onto 6 separate domains.

As well as testing tags included directly into the page body, we also investigated the effect of loading tags after the page had been completed by using the "DOM ready" event provided by the popular JQuery library. Including tags after the page has been completed is popular because JavaScript calls can block the page from loading and adversely affect the user experience.We prepared a JavaScript library and provided each site with 10 tests, the JavaScript and checked them for cross-browser compatibility. A copy of the source code for the library, latencytest.js, can be found in the appendix.

There were 4 test points:

1. Start of the head (immediately after <head>)
2. Start of the body (immediately after <body>)
3. End of the body (immediately before </body>)
4. After the DOM complete

There were 3 ways to collect the data

1. Conventional 1 x Invisible Image Pixel
2. JavaScript pre-loading of image pixel
3. JavaScript call that returns blank text

In addition to delivering tags directly into the page, some tags were delivered within IFrames in to mirror the popular container tags solutions bundled with Ad Serving solutions such as Atlas' Universal Action Tag and DoubleClick's Floodlight. Separate static HTML files were created for each iFrame and included cache-busting code.

Although some of the tests were image calls, JavaScript was used to write out all of the tests in order to prevent skewed results:

- to ensure a “cachebuster” was correctly used to prevent browser caching
- to ensure all of the people exposed to all of the tests accepted javascript

The JavaScript library (latencytest.js) was compressed down to just over 2k in size and loaded before any of the tests were commenced.

The initial tests included code to detect whether the page DOM had completed, which proved unreliable. These results were discarded and the DOM readiness detection provided by JQuery was used to perform the task.

Rather than just using the existing TagMan JavaScript tag delivery libraries, all of the JavaScript used for delivering the tags was written specifically for the project. This ensured independence and enabled us to publish the code so that the results can be validated and replicate the experiment if needed.

To count the number of times each tag was requested, we used TagMan campaign tracking with no filtering. The figures reported using this technique are equivalent to those that would be obtained from using static files and web server log analysis.

Last, the content was served from eight different domains to prevent hitting the two concurrent requests per domain limitation that most browsers apply. All of the aliases used were for either our global CDN or our UK data centres.

How TagMan addresses tag latency issues

First up, TagMan houses all of a page's tracking tags in a single, JavaScript tag, reducing the number of tags on the page and thus addressing the problems with page download times.

A deeper solution for avoiding tag-related latency is to wait for the page DOM to complete before loading the tags. The current version of TagMan can do this and our tests showed the figures for tags deployed in this way were almost identical to those for tags placed at the bottom of the body.

We are seeing more companies waiting for the DOM to complete before inserting their tags, however, not all tags can be added in this way and, under the worst-case scenario, will blank the page. Another issue with including tags in this way is that if there are any errors in other JavaScript code then it can prevent DOM readiness from being detectable, which means the tags are not fired at all.

TagMan already allows some control over where tags are deployed in the page and we are extending that in our next version to allow companies further flexibility in this area.

The main issue around tag delays is the asynchronous way JavaScript assets are loaded into the page. Thankfully, solutions are in the pipeline that could provide a solution to this problem that will mean tags can be deployed at the top of the page without risking latency issues. One technology is called Web Workers and it is supported already by the latest versions of Firefox and Safari.

JavaScript Web Workers allows developers to run JavaScript in parallel on a web page, without blocking the user interface. JavaScript is loaded and executed in the background but cannot access the parent page so we are researching the best ways to wrap this feature around existing tracking tags so that our customers can take advantage of this handy new technology.

We are committed to best practice around tag delivery and will continue to investigate ways to offer our customers considerably better ways to deploying tags through their CMS or 'piggy-back' tag solutions.

For a copy of the full report, including full results and the source code used in the test, email contact@tagman.com

About the author

Paul Cook is the founder and CEO of one-tag campaign tracking solution TagMan and a pioneer in web analytics.

In 1997, he founded RedEye, the first company in the world to offer post-click / post-impression tracking solutions.

Having sold RedEye in 2004, he founded TagMan, which, in 2008, won the Econsultancy Innovation Award for web analytics, where the judges said: "The creation of a single system and interface through which tags can be deployed is a significant innovation that can help remove the burden caused by the proliferation of tags as well as enabling complete campaign tracking."

Follow Paul on Twitter at: http://www.twitter.com/tagmanceo

About TagMan

TagMan is a one-tag solution to the problems of online campaign tracking. By acting as a single system through which tags can be deployed to an advertiser's web site, online marketers can save time and money in the way they track campaigns and see how all online channels are working together.

A single TagMan tag is installed on any page that needs tracking and all other tags that need to sit on that page - whether to track natural search, paid search, affiliates, display or site analytics - can be plugged into the site through it.

Since the tags from all channels sit in the same system, advertisers and their agencies can track the full customer journey and tell which channel delivered a particular user much more effectively. This allows them to plan their future activity more effectively and eliminate duplicate commission payments where more than one channel claims the same sale.

The browser-based technology enables tags to be added, edited or removed direct from the web page in question in minutes and without adjusting the code on the site. This minimises errors, increases site security, and reduces the huge amount of time and energy it can take to implement online campaigns.

Also, since tags can be easily added and removed, it allows agencies and advertisers to move between tag providers such as ad servers and affiliate networks as they see fit. And, as the data sits in a system controlled by the advertiser/agency, they are able to control which campaign data they share with different partners and suppliers.

Last, page download times and data accuracy are increased since just one tag sits on any one page. Clients include advertisers Thomas Cook, Alliance & Leicester, and Christy Towels and agencies Media Contacts, TBG London, Blue Barracuda and Didit.

Find out more at http://www.tagman.com
Follow TagMan on Twitter at: http://www.twitter.com/tagman_news

Further Reading

Web page download optimisation

- http://yuiblog.com/blog/2008/07/22/non-blocking-scripts/
- http://ejohn.org/blog/web-workers/
- http://developer.yahoo.com/performance/rules.html
- http://developer.yahoo.com/yslow/
- http://www.httpwatch.com/
- http://getfirebug.com/

Google Analytics:

- http://alex.dojotoolkit.org/2009/04/ending-the-gajs-wait/

Reconciling data between 3rd party tracking solutions:

- http://www.kaushik.net/avinash/2008/11/ultimate-web-analytics-data-reconciliation-checklist.html

Latency study into tracking redirects:

- http://www.mikeonads.com/2008/09/16/cant-we-all-just-302-report-on-redirect-timings/

For a copy of the full report, including full results and the source code used in the test, email contact@tagman.com