Dynamic SiteStats vs Log File Analysis

June 2004

Executive summary

Dynamic SiteStats' audience-analysis technology and traditional log-file analysis tools may produce greatly different traffic statistics for the same site. This is not surprising - by collecting traffic data directly from the browsers of actual users rather than from the log files generated by web servers, Dynamic SiteStats is generally more accurate than log-file analysis, and this difference in accuracy can result in significant differences in statistics.

The difference in page-view counts may be considerable: log-file analysis tools often report 400% more page views than browser-based analysis tools such as Dynamic SiteStats.

As discussed below, this examination demonstrates that the Dynamic SiteStats count is in fact accurate, and shows how various inaccuracies inherent in log-file analysis result in a large overcount.

The matrix below shows potential sources of discrepencies in page-view and visitor counts.

	Influences	Source of error	Effect	Significance
Existence & location of the tracking code	Page views and visitors count	SiteStats	Undercount	+
HTML Frames	Page views	Log file analysis	Overcount	+++
Cached pages	Page views	Log file analysis	Undercount	++
IP Address pools	Visitors count	Log file analysis	Overcount	++
False page views	Page views	Log file analysis	Overcount	++
Artificial traffic	Page views and visitor count	Log file analysis	Overcount	++
Limited display devices & text browsers	Page views and visitors count	SiteStats	Undercount	+
Internet connectivity	Page views	SiteStats	Undercount	+

The tracking code

To implement Dynamic SiteStats on a site, the site owner inserts a section of HTML/Javascript code in the HTML for each page to be monitored. When a page containing this code is displayed on a user's browser, the code collects page-view and other traffic data.

Two issues regarding the code are relevant: pages that do not contain code, and the location of the code within each page.

Pages without code

When implementing Dynamic SiteStats on a site, the site owner may choose to include the code on certain pages and omit it from others. Dynamic SiteStats collect statistics only for those pages that include the code.

In contrast, log-file analysis tools usually provide statistics for all pages unless configured otherwise. This can be a significant - and often overlooked - source of differences between statistics from Dynamic SiteStats and log-file analysis.

Location of code

In order to improve accuracy, for slow loading pages it is desirable to insert the Dynamic SiteStats code near the beginning of the HTML for the page. This ensures that the code is executed whenever the page is displayed. If the code is located later in the HTML - especially after a large graphic or other slow-loading element - a user might enter and leave the page before the code is executed. In this event, the page view would not be recorded, resulting in an undercount.

HTML frames

Frames are independently controllable areas on a web page, used to provide added flexibility in display and functionality. Typically, there is a separate HTML file for the page itself and for each frame on the page.

Frames can be a problem for log-file analysis. When a user requests a page containing one or more frames, the typical server log records one request for the page itself, plus one additional request for each frame. Log-file analysis tools generally count each request as a separate page view even though only one page is displayed to the user, resulting in a significant overcount.

Dynamic SiteStats does not have this problem. When implemented correctly, the code appears just once in the HTML for the page and all its frames, so it is executed only once for the entire page. As a result, Dynamic SiteStats records the entire page as a single page view regardless of the number of frames it contains.

Studies show that frames are the largest source of overcount by log-file analysis tools.

Cached Pages

Many ISPs maintain proxy servers that store millions of pages copied from the web. When a user requests a page stored on a proxy, the ISP delivers the page quickly from the proxy rather than using the web server to actually retrieve the page from the web, which can take much longer. Surveys indicate that proxies serve 15 to 20 percent of the page views for a typical site. Log-file analysis cannot detect all page views served by proxies, resulting in significant undercounting of page views. Dynamic SiteStats detects all displayed pages regardless of the source (including pages served by the browser's cache), resulting in accurate page-view counts.

IP address pools

Many ISPs have a pool of IP addresses that are dynamically assigned to individual users. In this situation, a single user may use multiple IP addresses over time - even during a single visit to a site. Since log-file analysis identifies individual users by their IP addresses, it cannot track a user whose IP address changes.

As a result, counts of unique users and measurements of how long users spend on a site and on individual pages may be grossly inaccurate. In contrast, Dynamic SiteStats utilizes an internal session cache and does not only depend on the IP address to identify individual users, so it provides correct values for these statistics.

False page views

Web visitors following familiar paths often jump between pages very quickly without viewing their content. Log-file analysis cannot detect this activity, and consequently counts each jump as a page view even though the user does not actually view the page, potentially resulting in significant overcounting of page views. Dynamic SiteStats provides a unique solution to this problem: The site owner inserts the tracking code at the end at the HTML for a page, or following key content. If a user leaves the page too quickly, the code will not be loaded, in which case Dynamic SiteStats will not record a page view. This technique makes it possible to obtain more realistic page-view counts in this problematic situation.

Artificial traffic

Another common source of excess page-view counts by log-file analysis is artificial traffic - automated programs that request web pages from servers but do not display those pages to users. Log-file analysis tools generally count such requests as page views even though they are not true views by users. We examined the effects of two types of artificial traffic: monitoring tools and robots.

Monitoring tools

Many sites use proprietary or commercial tools to monitor various aspects of site performance. Such tools may request pages from the web server. Log-file analysis tools incorrectly count these requests as page views.

Dynamic SiteStats does not count such requests as page views. This is because the code is technically an image. Since monitoring tools typically do not execute image code, Dynamic SiteStats correctly disregard the page view for the pages they request.

Robots

Robots (also called “spiders” or “crawlers”) are programs that surf the web automatically, following hypertext links and scanning site content. Since robots are not actual users, their activities need to be excluded from traffic statistics.

This is difficult with log-file analysis: in order to identify the activity of a robot, a log-file analysis tool needs to know about the robot, much as anti-virus software needs to know about a virus in order to detect it. Since there are thousands of robots - and new ones appear every day - log-file analysis tools cannot identify every one. In fact, recent studies have identified hundreds of robots not detected by popular log-file analysis tools.

Dynamic SiteStats does not have this problem. Like the monitoring tools described above, robots do not execute the Dynamic SiteStats code. As a result, Dynamic SiteStats automatically excludes robots’ activities from its traffic statistics without the need to identify specific robots.

Limited-display devices

PDAs and other limited-display devices such as text browsers are often configured not to display images. This does not affect log-file page-view counts - the log file records each page request by such devices just as it would for any device, and the log-file analysis tool counts each request as a page view. In contrast, since the Dynamic SiteStats code is technically an image, it is not executed when the page is displayed without images. As a result, Dynamic SiteStats does not count page views for such devices.

This difference cannot be unequivocally identified as an overcount by log-file analysis or an undercount by Dynamic SiteStats; this depends on the requirements of the site owner. If site owner's intent is to display advertising (usually in image form) to users - which is typical for commercial sites - then Dynamic SiteStats is correct to omit these non-image page views from the count.

Internet connectivity

Although many users enjoy high-speed access to the web, others contend with slow access due to low- speed connections, congested networks and other impediments.

In extreme cases, conditions like these may prevent Dynamic SiteStats from recording traffic statistics for these unfortunate users. Although we did not observe clear evidence of this effect in our study, it may have contributed to the difference between the two techniques.

Conclusion

Our comparison of Dynamic SiteStats' browser based tracking and a popular log-file analysis tool underscores two important points:

The differences in traffic statistics between Dynamic SiteStats and log-file analysis tools are to a great extent the result of their different data-collection methods.

Inaccuracies inherent in log-file analysis can produce significant errors even for typical sites.

This study is by no means an exhaustive comparison of the two techniques. There are many other differences, including accuracy of other statistics, level of detail, speed, accessibility, reliability, ease of operation - and ultimately, value to the site owner. Future studies will compare Dynamic SiteStats and log-file analysis in some of these other important areas.