Measuring Web traffic, Part 1

Getting to know your audience is the key to online success – make the numbers work for you

Andrei Malacinski (malacins@us.ibm.com), IBM Application Integration Middleware Lab
Scott Dominick (scottdom@us.ibm.com), IBM Application Integration Middleware Lab
Tom Hartrick (thartric@us.ibm.com), IBM Application Integration Middleware Lab

March 2001

The best way to know whether your Web site is achieving its goals is to gather extensive traffic data -- not just how many hits you're getting, but which pages are popular, who's visiting your site, when do they visit, and a host of other data that can give you a clearer idea of what's going on. In this article, Scott, Tom, and Andrei introduce you to the benefits of measuring Web traffic, exploring the importance of Web metrics, and describing various approaches of collecting Web data. They also show you how to choose whether to tackle this effort in-house or turn to an application service provider. Finally, they tell you how to make use of this data once it's been collected.

Introduction

"How many hits is my Web site getting?" "How many visitors are going to my site?" "What pages are people looking at?" "Where are my Web customers coming from?" "Are my links broken?" Do you find yourself asking these questions? Are you finding that it is becoming more and more important to measure the traffic on your Web site? Do you find yourself asking "How do I go about measuring the usage traffic on my site?" and "What information can I get?"

This article answers these questions and more. There are several Web measurement approaches that have been adopted by the industry, such as network monitors, single-pixel solutions, and HTTP server log analysis. We begin this article with a discussion of why Web metrics are important and then provide an introduction to the various types of Web measurement approaches with site traffic metrics obtainable with these approaches. When choosing a Web measurement approach, decisions have to be made about the technology to employ, its cost, its setup, and coordinating it with existing enterprise data. This article looks at these and other decisions related to collecting Web measurements.

We provide an introduction to measuring Web traffic that includes a brief description of various Web measurement approaches. This article and its follow-on, "HTTP server log analysis approach," are designed for Web site developers, those in IT who deploy Web sites, company executives interested in site performance, and anyone who wants to know more about Web site tracking capabilities and techniques currently available in the industry. The subject matter is technical, but it does not require any prerequisite skills or experience to be comprehended.

Why Web metrics are important

The power of the Web is astounding. As an extension of other sources, or perhaps as the only source, the informational, operational, and marketing aspects of Web media are potent. Having a Web presence is not sufficient, though: e-businesses must exploit the latest technology and evolve more quickly than their conventional competitors and customers or risk getting left behind. Recognizing that Web site behavior has a direct effect on business success and customer loyalty, companies are now required to better understand their users so that their sites are responsive, easy to navigate, and present what the users are interested in purchasing. However, many e-businesses have little or no idea what customers and potential customers are doing on their Web sites. Web sites that are slow to respond and contain hard-to-find items will drive away customers. To justify Web site changes, though, using the right quantitative and
Qualitative information is essential in optimizing a site, which means there is a growing need for new tools that analyze Web site effectiveness -- that form of analysis is known as **Web analytics**.

Consider this scenario: If a Web site is promoting a sale through an advertisement that is valid for only one day, the Web site owner might want to know if the campaign was successful. One indicator would be whether the campaign generated additional site traffic. The following Web metrics chart can help illustrate whether or not the campaign was successful in generating increased Web traffic on the day of the promotion.

**Hits by Day of the Week**

Web analytics is the monitoring and reporting of Web site usage so that there is a better understanding of the interactions between Web visitor actions and what the Web site offers, as well as leveraging that insight to optimize the site for increased customer loyalty and business benefits. Web analytics can assist with the following tasks:

- planning Web infrastructure capacity to handle future growth;
- understanding qualities of new and repeat visitors;
- targeting offers and campaigns to categories of visitors;
- determining appropriate investment in online advertising campaigns versus other channels;
- identifying which e-business partners to work with based on generated referral traffic and realized profits;
- decreasing investments in, or altering the navigation to, Web pages that get little traffic;
- and much more.

The diagram below illustrates how Web analytics can be used in an organization:
Methods of data collection within a Web environment

The capability, comprehensiveness, and timeliness of a Web metrics solution depend on the methods used for capturing Web usage data. We introduce several methods, all of which are viable options to use -- either alone or in combination -- along with some discussion of their benefits and drawbacks. A prioritized list of Web metrics requirements should be agreed on and used to help decide which method or combination of methods best answers the Web metrics questions with sufficient details and accuracy in the appropriate time frame.

To help set some context, there are three main locations where Web data can be captured within the Web environment:

- The backend servers that deliver the Web content, including the array of HTTP servers, application servers, commerce servers, and so on.
- The Internet Service Provider (ISP) handling the data flow between the Web servers and a user's browser.
- The client browser that displays the Web pages.

The analysis options overviewed here are: HTTP server log analysis, server and network monitors, and single-pixel analysis.

HTTP server log analysis

This form of Web traffic measurement, or Web analytics, involves the analysis of log files produced by the HTTP servers in your Web server environment. Each HTTP server vendor (such as Apache, Microsoft Internet Server, Netscape, Domino GO, and so on) provides logging capabilities with their products. These logging capabilities typically include configuration options that enable and disable logging as well as specify the type of data logged and the quantity of data logged. Data is logged to files in one or more file formats. As Web server software has evolved, so has the variety of logging options and logging implementations. Over this period the log file formats that servers log to has settled on a small set of formats, namely:

- NCSA Combined Log Format
- NCSA Separate Log Format (3-log format)
- NCSA Common Log Format (access log)
- W3C Extended Log Format

A typical HTTP logging configuration would result in a log entry for each HTTP request or hit to the server. This entry contains detailed information about the resource request. Web analytics software can parse and process these logs files, in batch, to combine information from each request to give a view of the Web site's traffic, including basic metrics.
such as the number of hits, visitors, visitor duration, visitor origin (subdomain, referral link), visitor IP address, browser type and version, platform, and cookies, as well as more advanced metrics derived from the manipulation of data through techniques such as categorization and aggregation. Categorization is the process by which similar items (such as URL, browser, platform, and so on) are grouped together based on pattern matching. Aggregation is the process by which all combinations of entities and their resulting measures are combined. Categorization and aggregation will be further defined in a follow-up article, “Measuring Web Traffic: HTTP Server Log Analysis Approach.” The following diagram captures the essence of the HTTP server log analysis process.

The figure below shows the logfile-based traffic analysis process.

### Server and network monitors

A server monitor typically runs as a plugin to the Web server, getting information about each event through an application programming interface (API). Server APIs are proprietary, so the events and data seen by server monitors depend on the Web server release. Usually, a server monitor can get unique visitor IDs, referrer pages, and more. Some data isn't available to server monitors. For example, when a visitor interrupts the transmission of a page by hitting the stop button, typing in a new URL, or clicking on a Web shortcut, this has the effect of a "stop request" being sent to the Web server. The Web server then interrupts the transmission of the page being sent. Interrupted transmissions are very informative, as they may indicate that a particular resource is taking too long to generate, or that the whole Web server is overloaded. Unfortunately, Web servers typically don't notify plugins or record when transmissions end prematurely. Installing a server monitor does introduce some risk to the Web server, because a problem with the monitor could crash the Web server. Also, a server monitor that directly calls a database server introduces even higher risk. If it is a production Web site and requires a server monitor, the monitoring and recording processes should be separated, and/or the server monitor should be isolated in a separate process.

Network monitors perform "packet sniffing" through an application that registers a function with the operating system (OS) called to view each packet as it crosses the wire. A network monitor should be installed on each Web server; however, with traditional Ethernet LANs a single network monitor could report on every HTTP event on the subnet. Network interfaces give you the choice to either sniff packets from a single machine or sniff all packets on the wire. When network monitors sniff all packets, they consume much more processor time, making it nearly impossible to implement on high-traffic sites. In this case, the only practical solution is to install a network monitor on each Web server machine. A network monitor can see everything, including client requests, server responses, cookies, and HTML files; it can also track stop requests issued from the browser, making it possible to list the pages that are taking too long to generate. It can measure the Web server's response time to different requests. Some network monitors can report on...
content-related HTML tags and capture "form data" transmitted via a POST request when the visitor hits a submit button.

A network monitor significantly reduces risks to Web server operation by placing data on its own machine to isolate the traffic analyzer from the Web server. If the network monitor crashes, the Web server is unaffected. The risk is minimal, even when running on the same machine, since the network monitor is a separate process operating independently from Web-server processes. One significant drawback to network monitors, however, is that they can't track encrypted information from secure Web servers.

**Single-pixel analysis**

One of the newer methods for Web site data collection, referred to as single-pixel technology, enables on-the-fly data collection of page view information. This method provides an alternative to batch log data collection, lending itself to high traffic Web sites where continuous collection of page view information helps keep up with the data volumes. It usually provides more timely access to usage and visitor data than batch processing of logs. For these reasons, it is a popular choice among Web analytics service providers.

The single-pixel terminology originates from the transmission of data during the request for a one-pixel image placed somewhere on a Web page. Single-pixel technology is instrumented by adding HTML and JavaScript code to the enabled Web pages via manual modification or specialized tooling. Some or all of the pages on a Web site can be instrumented for single-pixel data collection specific to any or all of the individual pages on the site.

The data collected includes basic information similar to that logged by an HTTP server, as well as client-side behavior, and can even be further customized. The information is collected when the enabled page is loaded in a browser, and then sent back to the analysis server for analyzing and reporting processes. Single-pixel tracking has negligible effect on a visitor's Web page usage. The tag itself is not visible to the user and does not modify the look or layout of the tagged page in any way. Keeping the associated JavaScript code very small, usually around 500 bytes, further minimizes the impact on page download speed, and ideally the data is POSTed to the server with no content returned to the client browser. Note also that the tag gathers statistics only on the page containing the embedded tag. When the browser leaves a tagged page, the JavaScript halts.

Single-pixel information collects information by page views, rather than by "hits," as recorded in HTTP server logs. This detailed page view information is available once the page is fully loaded, the JavaScript is enabled, and it's determined which of the objects loaded in the Web page are page views within the Web site (and not references to other Web sites). The benefits that single-pixel technology brings are powerful, and an analysis solution isn't required to "stitch together" multiple Web logs, which may need to be collected from around the world.

The JavaScript collects a number of statistics relevant to the tagged page, including:

- time it takes the page to load;
- whether or not the page loads with errors;
- whether or not the page load is aborted by the user;
- referring page of the loaded page (if there was one);
- link destination clicked to leave page (if there was one);
- usage information about any forms on the page;
- session state (for example, prior visits, duration, number of page views, and so on);
- traversal path;
- and more.

The locations of the tags and the number of tags embedded on your site depends entirely on the extent to which you opt to customize.

If JavaScript is disabled or not supported by the browser, single-pixel methods can still be supported, but require different capture processing involving an HTML IMG component (to request the single-pixel image), HTTP GETs (instead of POSTs), and use of servlet redirection (to prevent loss of single-pixel data due to caching).

**Summary**

To determine which Web metrics analysis method (or combination of methods) is appropriate, it is important to understand the strengths and weaknesses of each. This must be done in conjunction with a comparison of the
requirements and investment being made for implementing a Web metrics solution. For example, overall, log analysis implementations are generally less expensive than implementing network monitors or real-time analysis systems. Log analysis solutions can give fairly robust historical trending information, as well as the ability to filter, categorize, sessionize, and aggregate the results. However, there are some Web server issues to keep in mind: It may need to reconfigure its logging; it's limited to what is captured in a standard HTTP log; it can be tedious and resource-consuming to analyze the volumes of data; and it's a batch process so it isn't real-time. On the other hand, packet-sniffing solutions provide real-time capabilities but bring higher setup and maintenance costs, and could introduce risks to the production Web-serving environment. Single-pixel implementations offer powerful potential for real-time analysis, especially when combined with the positive aspects of log-based analysis, but this requires up-front page instrumentation.

For example, if you're just getting started with understanding the power of and the required infrastructure for Web metrics -- and the goal is to regularly report on overall Web metrics such as hits, page views, visit trends, and referrals for a site -- then a log analysis solution may best meet your needs, since HTTP logs contain this data and analysis poses no risks to the production servers or page instrumentation. However, if the requirements are for real-time analysis, for understanding what is happening on a site at a given moment, or for feeding a personalization engine, then a packet-sniffing or single-pixel implementation may best meet the objectives if the higher risks and lesser amount of historical data capture are understood. A customer may also decide that the additional investment needed to implement a combination of these methods is what is most desirable.

Also, although we haven't covered it in detail, a byproduct of Web analytics that deserves consideration is persistent data storage. There are methods that do analysis in-memory, but after the reports are run, the data isn't preserved as it would be with a persistent database solution. However, as Web site traffic and analysis requirements continue to grow, continually increasing data processing and storage creates a burden on the system that requires decisions about whether to keep all data or only summary information to be made. Keeping all the data, though more resource intensive, may be a requirement for some of the advanced query and reporting systems that are used to understand the analysis results.

Web metrics delivery methods -- in-house product and service providers

One of the biggest decisions to make when determining a Web metrics solution is whether to implement it in-house using an available or home-grown software package, or to use a service typically run by an application service provider (ASP). Each option requires that decisions be made about location, setup, resources, pricing, and customization. Each of these elements is briefly considered below.

Location

An in-house software implementation keeps both the Web data and the results of analysis in-house, whereas an ASP solution requires that data and results be transferred between the customer and the ASP. This distinction is important. For example, for some customers like financial institutions, having visitor and enterprise data leave the enterprise may be reason enough to want to implement an in-house solution. In an ASP solution, visitor data collected in logs or via a real-time mechanism is regularly transferred, analyzed, and stored by the ASP itself. For the other kinds of customers, dealing with the higher resource aspects of implementing an in-house solution may be reason enough to use an ASP. An ASP manages the databases and regularly delivers Web analytics data in secured reports to their customers as stipulated by a mutually agreed upon contract.

Setup and resources

Due to the requirements of investigating available Web analytics packages, securing and maintaining the necessary hardware, setting up and maintaining the software, and so on, an in-house software implementation may take weeks. Whereas an ASP solution comes with a vendor who handles most of these tasks, making it possible to setup in just a few days. An in-house implementation likely has more autonomy and flexibility than that run by an ASP, but a company needs to understand that it must allocate the appropriate time and resources (both hardware and people) to run it. A company running an in-house Web analytics solution will fail if they use an old processor that just happens to be available or do not recognize

Glossary of terms

Browser: The Web browser used by a visitor to access the Web site.
Bytes transferred: The number of bytes transferred to the client Web browser as a result of a request.
Domain: The unique name that identifies an Internet site (EDU, ORG, COM).
Duration: The amount of time spent on a page, in seconds.
Duration per visit: The amount of time spent in a given visit, in seconds.
Entry resource: The first resource viewed as part of a visit.
Exit resource: The last resource viewed as part of a visit.
Hit: A browser request for any one item, such as a page, graphic, or other resource. It may take several hits to bring up a single Web page as displayed in a browser.
Hits per visit: The number of deliberate requests to a given URL. For example, one Web page that contains three frames and 12 artwork files would generate one page view, but 15
that it takes people to manage the growing data analysis responsibilities.

**Pricing and costs**

An in-house implementation primarily incurs its costs upfront for the software, hardware, and so on. It also incurs the ongoing costs associated with the personnel and systems maintenance, while an ASP normally charges a setup cost followed by a regular monthly fee. An ASP is likely to charge based on the amount of data analyzed, the types of reporting capabilities requested, and the length of time the raw data is preserved. These ASP costs could be several thousand dollars a month, and may grow higher as Web usage and/or reporting requirements increase. An in-house implementation could be $20,000 to $100,000 initially, but it offers full control over the amount, type, and frequency of analysis options and reporting.

**Customization**

With an ASP solution, a customer is limited to the analysis and reporting options it offers. Although the ASP may offer a wide variety of choices, unless the ASP setup is brought in-house, there is only so much that can be customized specifically for a single customer -- or tied in to other customer business applications. An in-house implementation is more highly customizable, allowing analysis and reporting on unique Web site activities and tying the results of Web analytics into other e-business applications, reporting tools, and so on.

**Summary**

Both in-house and ASP Web analytics solutions have their strengths and weaknesses. In-house solutions can be combined with other e-business and customer information, while ASPs can get up and running more quickly, give a good analysis base, and don't require so many in-house resources. As businesses develop a Web analytics strategy, they may even decide to use a combination of both -- using an ASP to get a timely overview of Web usage and visitor behavior, and implementing an in-house solution for more advanced business intelligencetechniques (for example, data mining, personalization, and so on).

**I've got all this data . . . now what?!**

Web analytics offers the opportunity to capture a lot of information on your Web visitors and content. So what next?

To start, you need to decide how you might use the information that's recorded about your visitors. Then write a privacy statement based on the visitor's point of view, and make that statement available on your Web site. Visitors prefer to view products and pages that interest them, so they usually share information for that purpose. However, visitors typically prefer that they be asked for their permission to send them marketing e-mail or to share their contact information with partner companies. If a site provides a privacy statement documenting the intended uses, and gives visitors an e-mail address for comments, visitors can determine whether the policy is acceptable. A World Wide Web Consortium (W3C) project called "Platform for Privacy Protection" (P3P) is helping to define and standardize the policies for data collection and the legitimate uses of this data.

Next, consider the benefits of advanced **data warehousing** and **data-mining** techniques. Data warehouse reporting systems, such as those provided by traffic analyzers, aggregate and report facts over different dimensions. These warehouse reporting systems are commonly called **on-line analytic processing** (OLAP) systems. OLAP systems can report only on directly observed and easily correlated information. They rely on the user to discover patterns and decide what to do with them. To solve this problem, marketers and business analysts use data-mining techniques. These are machine learning algorithms that find patterns buried in databases and report or act on those findings. When Web data is combined with other enterprise data (such as customer and product profiles, order and fulfillment system data, and so on), it makes for a powerful business intelligence solution.

An example of a data mining scenario would be a business that sells blue jeans through both storefronts and online ordering capabilities. If the online system records its revenue and other business-related information in a database, and
that data is merged with the storefront's business data, a complete picture of the company's business can be found in one system. If the business reports an increase in the number of sales of blue jeans for the month, the business manager can determine how many of those sales were attributed to on-line sales versus the traditional storefront sales. Also, Web analysis and data mining can provide insight into the demographics of customers purchasing blue jeans and other items that are likely to be purchased with blue jeans.

There are many data-mining techniques available. To use data mining on your Web site, you have to establish and record data in several areas, including visitor demographics (for example, address, household income, and so on), psychographics (for example, technology interests), and technographics (for example, system specs, browser type and version, and so on); item characteristics include Web content information (for example, URL, product information, and so on) and information about visitor interaction with the company (for example, purchase history, preferences, and so on). Even if a data-mining process isn't in place yet, this data will be a gold mine for implementing a data-mining system in the future. Data mining can reveal relationships and patterns that would go undetected by manual analysis techniques, and provide a wealth of information for personalizing Web content with products and customized information, and targeting specific user categories and visitors for campaigns, product associations, and predictions.

A technique called text mining could be used to further classify Web data. Consider the example of categorization, which uses pattern matching to group data together. If the user does not know which patterns to use to create categories, text mining could be used to search the data for the highest frequencies of a specific text string. Text mining software can "discover" text strings that occur most frequently in data. The user could interpret the mining results to determine the patterns that should be used for categorization (for example, "My Web site sells blue jeans but there seem to be many people interested in shirts as well."). Further integration between text mining software and Web analytics software could produce automatically-generated categorizations.

As a Web analytics solution is implemented and begins to mature, consider how this plays into personalization and customer relationship management (CRM) applications. These systems benefit businesses by helping to deliver better service, along with targeted marketing and content, to improve the user's experience and loyalty.

Conclusion

As you have seen, there are several approaches for choosing a Web measurements strategy. For starters there are technologies -- such as server and network monitor, single-pixel analysis, and HTTP server log analysis. Once you select a technology, there are other decisions you must make, such as whether to implement the solution in-house, or contract a service provider. Finally, once you begin gathering basic Web traffic measurements, you may want to expand the reaches of your solution to include advanced technologies such as data warehousing and data mining. It is our hope that this article will help you understand Web measurement. Happy analyzing!

Next time

Our next article, "Measuring Web Traffic: HTTP Server Log Analysis Approach," builds on the concepts presented here, but concentrates primarily on the approach of obtaining measurements from Web server logs, specifically HTTP server logs. To begin documenting that approach, the article explains HTTP logging (what it is, what the log formats are, what information is available, and so on). After that, we a look at the metrics available from the raw log data. These include basic metrics such as the number of hits, visitors, visitor duration, visitor origin (subdomain, referral link), visitor IP address, browser version and type, and platform, as well as more advanced metrics derived from the ability to manipulate the data through techniques like categorization and aggregation. Finally, it contains a brief reference to two solutions that use the server log analysis approach: IBM WebSphere Site Analyzer and IBM Surfaid Analytics.

Trademarks

The following are trademarks of International Business Machines Corporation: IBM, OS/2, and WebSphere.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of the Open Group in the United States and other countries.
Resources

- The National Center for Supercomputing Applications (NCSA) provides a great deal of information on Web metrics.
- IBM WebSphere Site Analyzer is a tool that can assist in collecting Web data.
- IBM Global Services Surfaid Analytics can assist companies that are interested in a Web metrics solution.
- Visit the Aberdeen Group to learn more about Translating Clicks into Business, December 2000.

About the authors

Andrei Malacinski is a software engineer for IBM in the Web software development area. He works at the IBM Application Integration Middleware Lab in Research Triangle Park, NC. Andrei has been both a developer and team lead on a number of IBM application development products for Windows, OS/2, and UNIX platforms. He is currently the team lead and lead developer of the IBM WebSphere Site Analyzer. He can be reached at malacins@us.ibm.com.

Scott Dominick is a software engineer for IBM in the IBM Application Integration Middleware Lab at Research Triangle Park, NC. He received his Bachelor's degree from North Carolina State University in 1992. He is currently working on the IBM WebSphere Site Analyzer in the WebSphere Tools Development area. He can be reached at scottdom@us.ibm.com.

Tom Hartrick is currently a software engineering manager for the IBM WebSphere Site Analyzer product, working at the IBM Application Integration Middleware Lab in Research Triangle Park, NC. Tom received his Bachelor's degree in Computer Science from Rochester Institute of Technology, and has previously been development manager for WebSphere Application Server as well as other software projects. He can be reached at thartric@us.ibm.com.

What do you think of this article?

- Killer! (5)
- Good stuff (4)
- So-so; not bad (3)
- Needs work (2)
- Lame! (1)

Comments?

Submit feedback