



**keynote™**

The Mobile & Internet Performance Authority™

## The Science of Web site Load Testing

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Key Variables and Metrics for Internet Load Testing</b>	<b>4</b>
2.1. Basic Server-side Web site Usage Variables and Metrics	4
2.2. Total Page Views Per Week	4
2.3. Total Hits Per Week	5
2.4. Total User Sessions Per Week	5
2.5. Average Session Duration	5
2.6. Average Page Size	5
2.7. Average Hit Size	5
2.8. Derived Web site Usage Variables and Metrics	6
2.9. Web site Specific Variables and Metrics	6
2.10. Client-side Variables and Metrics	6
2.11. Determining The Values of Client-Side Variables	7
<b>3. Web site Usage Signature</b>	<b>8</b>
3.1. Page Requests Distribution	9
3.2. Tracking WUS Changes	10
<b>4. Creating Realistic Loads</b>	<b>12</b>
4.1. Developing Scripts and Scenarios	12
4.2. Modify Scripts and Scenarios to Account for Client-side Variables	13
4.3. Comparing Web site Usage Signatures	14
<b>5. Conclusion</b>	<b>15</b>

## 1. Introduction

The concept and practice of software load testing has been around for many years, but the advent and exponential growth of the Internet has created a situation that has taken the magnitude and complexity of load testing to a whole new level. Traditionally, systems that required load testing existed in private LAN/WAN networks and were accessed at rather predictable times, with rather predictable patterns, by a relatively well known and predictable group of people. A Web site on the open Internet, on the other hand, is subject to highly unpredictable load patterns by a widely heterogeneous and unpredictable group of users. In the extremely competitive world of the Internet, unacceptable Web site performance and availability because of excessive loads can cause serious harm to a company's bottom line, market value and brand. For these reasons, knowing the capacity and scalability of business and mission critical Web sites is extremely important and proper load testing is the best way to acquire this knowledge. When load testing is not done properly, the results are at best useless and, in the worst case, misleading, causing a company to either underestimate or overestimate a site's capacity. This wrong result could cause unnecessary expenses, delays or potentially disastrous business decisions.

In this paper, we present the overview of a method for approaching the very new and challenging task of load testing Internet Web sites in a rigorous, systematic and repeatable manner. Scientific load testing allows companies to collect realistic, useful and reliable data about a Web site's capacity and scalability.

In the first part of the paper we explain the key variables and metrics we use to generate highly realistic load tests for Internet Web sites. While some of these variables and metrics are the same ones used in more traditional, non-Internet, load testing situations, most have either been tuned for the specific requirements of Web sites or have been developed specifically for them.

In the second part of the paper, we introduce and explain the concept of Web site Usage Signature (WUS), a very effective and systematic way for combining and presenting the above mentioned metrics.

In the final section, we describe the process for developing and testing load scripts and scenarios that accurately match the way a Web site is used and navigated by real users.

## 2. Key Variables and Metrics for Internet Load Testing

In this first section of the paper we introduce and explain some of the variables and metrics that are used to analyze and reproduce Web site loads. We partition these variables and metrics into the following categories:

- Server-side variables and metrics
  - Basic variables
  - Derived variables
- Client-side variables and metrics
  - Online behavior variables
  - Client system variables

### 2.1. Basic Server-side Web site Usage Variables and Metrics

This first set of variables and metrics provides a very high-level description of a Web site's traffic and usage patterns from the server's point of view. This description is easy to understand. Like a car's MPG rating, taken in isolation, these are coarse averages, measured over a wide range of conditions, so "your actual mileage may vary."

Nevertheless, like an MPG rating, their usefulness and directness make up for their lack of detail. Furthermore, when used as a set rather than in isolation, their interrelationships weave a picture that provides real insight into the way a Web site is used. These basics variables and metrics are:

- Total page views per week
- Total hits per week
- Total user sessions per week
- Average session duration
- Average page size
- Average hit size

All of these metrics can easily be derived from log files using a basic log analyzer. They are described below.

### 2.2. Total Page Views Per Week

A page view is the request of a single Web page with all its embedded objects. This simple metric is the number you'd like to have if you can only have one number to gauge a Web site's traffic. We chose page views (rather than page hits or bytes transferred) as the parameter for the numerator because this is the most intuitive and most commonly used unit for describing the traffic or size of a Web page. We chose one week as the denominator because using anything less than that for a sample (i.e. a day or an hour) fails to take into account some very common cyclic patterns that happen within a week, such as a much lighter volume on week-ends for business or financial sites. Longer time samples (i.e. a month or a year) are also inappropriate because in the highly dynamic world of the Internet, things can change dramatically in such time frames.

### 2.3. Total Hits Per Week

A hit is any request for a file received by the server, including images, sound files and any other type of file that is requested along with a page request.

### 2.4. Total User Sessions Per Week

A user session is the visit to the Web site where the interval between page requests from that user does not exceed a pre-determined time (we use 30 minutes).

Ideally, a user session would be defined as a visit by a unique user. The problem with this definition is that the concept of unique user is easier to describe than to measure. It's important to understand that, unless you do a retinal scan on your visitors, the information you get is not going to be as accurate as you might like. However, knowing unique users is still extremely useful and necessary for load testing. Having ruled out retinal scans, the best way to know that a user is unique is to request a user name and password every time the user visits the Web site. This is clearly slow and impractical for most Web sites. The next most accurate way to determine the uniqueness of a user is to use cookies (pieces of identification information put on the user's hard disk by the Web site.). The intrusiveness of this method, the fact that it's possible for a single system to be used by multiple users and the reality that many users disable cookies altogether makes this approach less than ideal and inaccurate. The final way to determine the uniqueness of a user is to use IP addresses. While this approach is the easiest and least intrusive of the three, it is the least accurate because most Internet Service Providers (ISPs) use caching and load balancing schemes that can make a single user appear as multiple users, or multiple users appear as a single user.

Despite possible inaccuracies due to the difficulty in identifying unique users, average user sessions per week is a critical metric for Web site load testing because the only practical way to load test a Web site is to simulate actual users and actual user sessions navigating from one page to the next (rather than, say, simply requesting disconnected pages). Fortunately, the inaccuracies introduced by leveraging cookies or IP addresses can be mostly neutralized by being aware of their potential inaccuracies and, more importantly, by standardizing on one method and definition for identifying unique users and using it consistently throughout the entire load testing cycle.

### 2.5. Average Session Duration

This is the amount of time the average user session lasts, measured in minutes and seconds, from the first page request until the last byte of the last requested page is served. All the caveats about user session measurements also apply.

### 2.6. Average Page Size

This is the size, in Kbytes, of a page view. This includes all frames, images, etc. for that page.

### 2.7. Average Hit Size

This is the average size of a hit measured in Kbytes.

## 2.8. Derived Web site Usage Variables and Metrics

The basic variables and metrics described in the previous session can be combined to yield an additional set of very useful metrics:

- **Average pages per session**—calculated by dividing total page views by total user sessions.
- **Average hits per page**—calculated by dividing the total number of hits by the total number of page views.
- **Average page viewing time**—calculated by dividing average pages per session by average session duration.

## 2.9. Web site Specific Variables and Metrics

The metrics introduced in the previous sections are common and should be used on all Web sites, but, in addition, each Web site should track some metrics that are specific to its mission. An E-Commerce Web site, for example, should track what percentage of user sessions result in an actual purchase, since such a transaction exercises and loads very specific subsystems (e.g. credit card authorization, secure server) whose performance is critical to the success of the Web site. Similarly, an online broker might want to keep track of how many stock quotes per session are requested by the average user, ratio of quotes to trades, etc.

Once the values for these variables have been collected, you begin to form a basic picture of what the Web site traffic looks like and what a realistic load simulation should look like.

## 2.10. Client-side Variables and Metrics

In our context, clients are people who visit a Web site and navigate it using a Web browser. The loads generated and the Web site response time experienced by these people can vary greatly depending on a number of what we call “client-side” variables. These variables fall into two major categories: online behavior variables and client system variables.

### Online Behavior Variables

As the name suggests, online behavior variables deal with behavioral differences between users. Some users, for example, read Web pages and navigate Web sites faster than others; we call this difference User Interaction Speed. Interaction speed is very relevant because, in a given amount of time, a fast user is able to go through more Web pages than a slower user. This results in more requests and therefore a higher load, for the Web site under test. These variables are used in combination with other Web site Usage Signature variables to create realistic distributions.

Let's say, for example, that the average viewing time for a Web site's home page is 53 seconds. Although this number is extremely useful, it's just an average. A slow visitor unfamiliar with the Web site may spend 2 minutes on the home page, while a fast visitor who is already familiar with the site may stay on it for just a couple of seconds before navigating to another page. A load test that does not consider these variations is simply not realistic and will generate misleading results. By combining the average viewing time with the user interaction speed variable and taking into account the user's familiarity with the Web site, you create loads that simulate real usage much more accurately.

Some of the key online behavior variables to consider for Web site load testing are:

- **Interaction speed**—as already mentioned, this is a measure of how rapidly a user processes a Web page and navigates to another page.
- **Latency tolerance**—a measure of how long a user will wait for a Web page to load before taking some action (e.g. abandon the Web site, hit reload). At the present time, there is a widely used rule of thumb which says that most users will wait approximately 8 seconds for a Web page to load and, after that time, they will start thinking about taking some other action.
- **Tenacity**—a measure of how determined a user is to accomplish something on the Web site. A user may have a low latency tolerance, but if the task they want to accomplish is extremely important to them (e.g. sell a stock during a steep market correction), the user will adjust his or her tolerance accordingly and will endure waits longer than usual.
- **Familiarity**—a measure of how well a user knows the Web site. It can be assumed that frequent visitors to a Web site know how to navigate it and where to go; they will process certain Web pages more rapidly than new users.

### Client-System Variables

Online behavior variables deal with differences between humans; client-system variables deal with differences between the hardware, software and location of the client system.

Some of the key client-system variables to consider for Web site load testing are:

- **Connection speed**—how quickly the user can access the site. In terms of load and user experience, the difference between a 56K modem and a T1 line are very significant. It's important to know the percentage of users in each connection speed category and use that distribution in the scenarios.
- **Location**—in what geographic region is the user. A user's geographic location affects a number of key variables that have a significant impact on load. The number of "hops" and the backbone speeds in the path between the Web site and the client system, for example, determines how fast the packets will travel and how many packets are dropped.
- **Software/Hardware configuration**—depending on the Web site, variables such as type of browser, browser plug-ins, type of OS, or CPU speed, may have a significant impact on load.

### 2.11. Determining The Values of Client-Side Variables

Several methods and assumptions can be used to determine the values and ranges for these variables. Again, a careful analysis of the Web site log files provides you with some basic guidelines. There are many commercial and freely available tools that directly provide you basic information (e.g. browser type, geographical location). Other variables may require a bit more work. To determine the interaction speed values and distribution, for example, you could select a few specific Web pages and extract from the log the average and standard deviation for the viewing time for each of those pages.

With this information, you can easily create a statistically significant model for interaction speed. Some of the variables require significantly more work. To identify the variables and distribution for familiarity, for example, you may need to first distinguish new users from returning users and then analyze how their page viewing times vary.

In the absence of log files, or the lack of time or resources for detailed log analysis, you can leverage some of the metrics and statistics collected by companies such as Nielsen//NetRatings, Keynote, or MediaMetrix. Nielsen//NetRatings, for example, provides data on average page viewing times and user session duration based on a very large sample space of users and Web sites. Although these numbers are not from your specific Web site, they can work quite well as first approximations.

You can also run simple in-house experiments using employees and their friends and family to determine, for example, the page viewing time differences between new and returning users. As a last resort, you can use your intuition, or best guess, to determine these variables' average value and standard deviation and assume a normal distribution for their ranges. For realistic load tests, even this last approach is preferable to ignoring these variables and creating a load test where every user comes from the same location, uses the same access speed, spends exactly 57 seconds on each page and waits until a server timeout to abandon the Web site.

### 3. Web site Usage Signature

A load test that does not reflect actual usage is at best useless and it could be dangerously misleading (e.g. causing a company to either overestimate or underestimate the capacity and scalability of their Web site, with potentially disastrous consequences). In the previous section we introduced and defined some of the key variables and metrics used in Web site load testing; in this section we describe how such metrics are used systematically to design, develop and test load scenarios that simulate, as accurately as possible, real load situations.

To measure how closely a load test matches real world usage, we developed the concept of Web site Usage Signature (WUS). The WUS is a set of metrics and measurements that, taken collectively, yield a very comprehensive picture of Web site activity. If the WUS from a load test does not match closely the WUS from actual usage, you should seriously question the validity and applicability of any conclusion drawn from the tests. Conversely, if the two WUS's are well matched, then your level of confidence that any results apply to real world situations increases considerably.

Since every Web site is unique and needs to track site-specific metrics, there is no standard format or standard set of metrics for the WUS. Factors unique to each Web site determine which metrics to include. The important thing is to include enough measurements to make it very difficult for two signatures to match closely unless the two loads are also closely matched. If, as a trivial example, we use a WUS that tracks pages/session, but do not track average page size our scripts may be biased toward substantially smaller pages and capacity predictions from the load test could be dangerously over-optimistic.

The following is an example of a WUS:

Web site Usage Signature for sample.com	
Metric	Results from Actual Usage (Averages for 1 hour)
Page view per hour	97,200
Hits per hour	602,640
Average session duration	6:36
Average page views per session	7.20
Average hits per page	6.2
Average number of quotes per session	4.5
Average number of trades per session	0.34

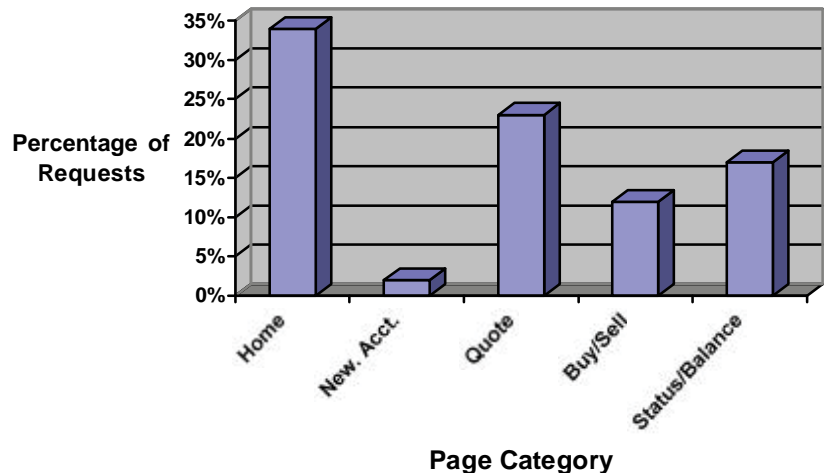
You can increase the precision of a WUS by applying additional statistics to some of the metrics. Instead of simply reporting average session duration, for example, a WUS might include all of the following statistics for session duration.

Session Duration Statistics	
Average	6:36
Median	6:02
Standard deviation	3:18
Minimum	0:32
Maximum	32:56
Percentage less than 2 minutes	12%
Percentage more than 12 minutes	8%

### 3.1. Page Requests Distribution

The WUS metrics we have discussed so far are extremely important to determine the size of the traffic. However, since not all pages or hits are equal, it's equally important to analyze what pages (or types of pages) are requested and in what percentages. To collect and present this information, we use the concept of page requests distribution. A page request histogram, shown below, provides an easy way to look at the distribution of pages.

Page Request Distribution



Page request distribution should be a key part of any WUS, because when planning and designing load test scripts and scenarios matching page requests is one of the most important factors in achieving realistic loads.

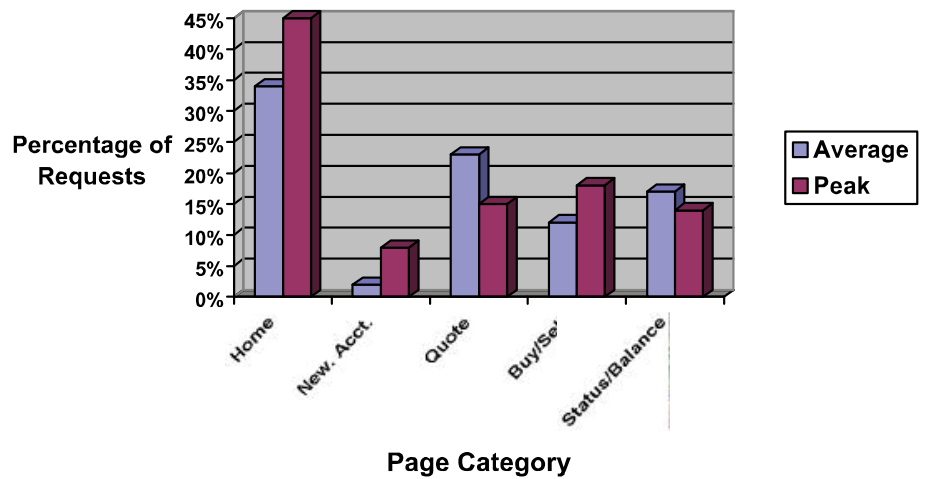
### 3.2. Tracking WUS Changes

A Web site's WUS can change dramatically during peak load periods. During a normal trading day, for example, the average session on an online brokerage Web site may have a ratio of quotes to trades of 14 to 1, because most people are simply checking on their stocks rather than buying or selling them. On a day when the market is dropping sharply, on the other hand, the quotes-to-trades ratio may change to 5 to 1, as many people decide to sell in a panic. Since trades put a significantly higher load than quotes on the Web site, this would be a key WUS to track and analyze. A load scenario for a sharp market correction should be significantly different than a load scenario for an average high-volume day. Similar dramatic differences in WUS can be expected on E-Commerce sites during special promotions or advertising campaigns or on portals or news sites when big news breaks.

Sharp increases in volume are accompanied by substantial changes in WUS. Designing and tracking a WUS with the right variables provides you with invaluable insight for creating highly realistic loads for very different scenarios. The following table and chart show how the WUS can change as the load shifts from average to peak.

Average WUS vs. Peak Load WUS			
Metric	Average Load Metrics	Peak Load Metrics	Comments
Sessions per hour	10,340	105,232	The average session duration, page views and hits are all considerably smaller during peak loads.  Most notably, a 10 time increase in number of users caused only a 3 time increase in hits
Page view per hour	97,200	456,708	
Hits per hour	602,640	1,806,343	
Average session duration	6:36	3:23	
Average page views per session	7.20	4.34	
Average hits per session	58.28	17.16	

**Page Request Distribution (Average vs. Peak)**



As the table and chart show, the WUS differences between average and peak load can be dramatic; if they are not studied and taken into account, the conclusions from the load test would be highly questionable. In the example above, for instance, there is a difference of 2.86 page views per session between average and peak load. If you do not account for an approximately 40% reduction in page views between these two scenarios, for example and use the 7.20 page views per session number in a peak load scenario (instead of 4.34) the session capacity reported by the test will be less than the actual capacity under peak load behavior.

## 4. Creating Realistic Loads

Having a Web site Usage Signature reduces the guesswork and greatly simplifies the task of creating and testing realistic load scenarios. This task can be systematically approached as follows:

1. Develop the scripts and scenarios to achieve the desired page request distribution.
2. Modify the scripts and scenarios to take into account client-side variables.
3. Execute the scenarios and compare the resulting WUS to the target WUS. If the differences between the two WUS's are not within your specified tolerances make the necessary adjustment and go back to step 1 and 2.

### 4.1. Developing Scripts and Scenarios

The first step is to develop a set of session scripts that can be combined in various proportions to achieve the desired page request distribution. Many log analyzers report the most common paths through a site and, in many cases, the top 10 or 20 paths through the site are representative of 80% or more, of all visits. If a report on the most common paths is unavailable or too broadly distributed to be helpful, you can deduce the required scripts by using the page request distribution, as the following trivialized example shows.

Assume that you have a Web site with three pages: Home, Product Info and Buy and that an analysis of the log files results in the following page request distribution:

Page Request Distribution	
Page	% requests
Home	58%
Product Info	31%
Buy	11%

To be able to achieve this distribution you need to develop at least three scripts:

- Script 1: Home→Exit
- Script 2: Home→Product Info→Exit
- Script 3: Home→Product Info→Buy→Exit

After deciding what scripts to develop, you must decide in what proportions those scripts should be executed to achieve the desired page request distribution. In this case, the script distribution should be:

Script Distribution	
Script	% execution
Script 1: Home→Exit	47%
Script 2: Home→Product Info→Exit	35%
Script 3: Home→Product Info→Buy→Exit	18%

If 100 scripts are executed using this script distribution, your page requests will be:

- Home Page: 100 requests (47 from script 1 + 35 from script 2 + 18 from script 3)
- Product Info Page: 53 requests (35 from script 2 + 18 from script 3)
- Buy Page: 18 requests (from script 3)

The total number of pages requested from all these scripts is 171 (100 Home + 53 Product Info + 18 Buy). A simple calculation shows that the target page request distributions will be achieved when the load is executed (e.g. 53 Product Info page requests represents 31% of all page requests, which is the target distribution.)

Clearly, most Web sites are significantly more complicated than our three-page example, but the same principle can be applied to sites of arbitrary complexity.

#### 4.2. Modify Scripts and Scenarios to Account for Client-side Variables

The next step in creating a realistic load test is to take into account what kind of users would be executing these scripts. This is where additional data from the WUS and the client side variables come into play. Let's consider as an example, two client side variables: latency tolerance and familiarity, then see how they would impact the realism of the load.

Latency tolerance is an indicator of how long a user will wait for a page to load before becoming dissatisfied and, possibly, abandon the Web site. Different users will have different tolerance levels; some may tolerate loading times of 20 seconds and others will leave the site if they are made to wait more than 8 seconds for a page to load. A load scenario can be made significantly more realistic by simulating the behavior of tolerant and intolerant users. This can be done by assuming a simple distribution and by modifying the scripts to take latency tolerance into account (e.g. by terminating a script if the simulated user experiences response time past its threshold for two consecutive pages.)

The following table shows a sample latency tolerance distribution and script modification:

<b>Latency Tolerance Simulation</b>			
<b>Latency Tolerance</b>	<b>% of Users</b>	<b>Page Response Threshold (sec)</b>	<b>Effect on Script</b>
<b>Low</b>	25%	8	Terminate script if 2 consecutive pages exceed the page response threshold
<b>Medium</b>	50%	16	
<b>High</b>	25%	24	

Familiarity is a major factor in how quickly a simulated user navigates from one page to the next. As with latency tolerance, different people will behave in different ways: users that are very familiar with the Web site move more rapidly (therefore creating more load per unit of time) than users who are visiting the Web site for the first time and need to read and understand how the Web site is organized to go from one page to the next. You can use a familiarity simulation table to partition the different types of users and define how familiarity impacts the script execution:

<b>Familiarity Simulation</b>			
<b>Familiarity</b>	<b>% of Users</b>	<b>Read, Write, Think Time Multiplier</b>	<b>Effect on Script</b>
<b>First-time users</b>	35%	2.0	The read/write/think time is adjusted for each type using the specified multiplier.
<b>Returning users</b>	55%	1.0	
<b>Experienced users</b>	10%	0.5	

The scope and length of this paper prevents us from going through all the possible client-side variables, but these two examples are indicative of how the simulation of other client side variables can be approached.

### **4.3. Comparing Web site Usage Signatures**

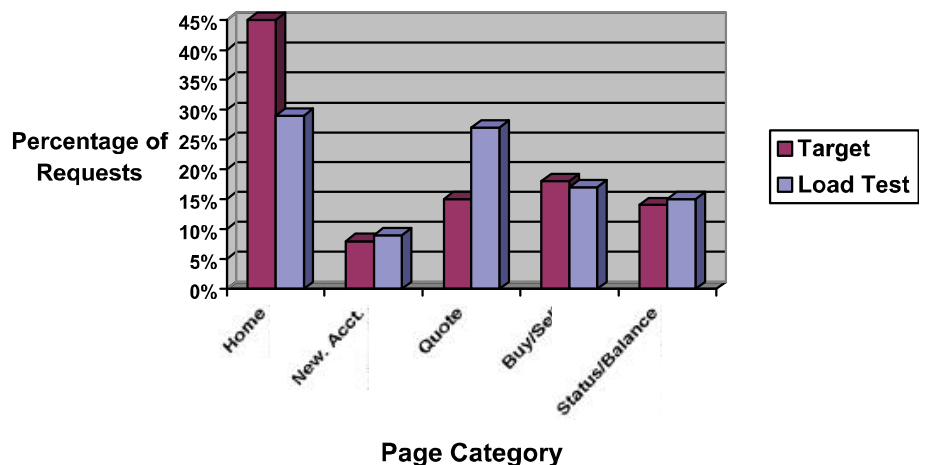
After the scripts and scenarios have been developed and modified to take into account client-side variables, they are executed and the resulting log files analyzed to determine how well the simulated load matches the target WUS. A WUS comparison chart and page request distribution table makes it easy to spot any differences.

In the following WUS comparison chart, for example, we see that although the page views per session and hits per session are both within five percent of each other, the session duration in the load test is 43.7% shorter than during real usage. This major difference would significantly impact the realism of the test and the accuracy of the results. The read/write/think time in the scripts should be modified to make the sessions last longer.

WUS Comparison Chart			
Metric	Results from Actual Usage (Averages for 1 hour)	Results from Load Test	Difference
Page view per hour	97,200	95,665	-1.6%
Hits per hour	602,640	580,443	-3.6%
Average session duration	6:36	3:43	+43.7%
Average page views per session	7.20	7.08	-1.6%
Average hits per session	58.28	56.23	3.5%

The following page request distribution chart shows another discrepancy between the load scenario and the target WUS. The percentage of home pages requested by the load test is significantly less than what the target WUS calls for, while the reverse is true for quote pages. The script distribution scenario should be changed to make the simulated load match the target load more accurately.

### Page Request Distribution (Target vs. Load)



Achieving a perfect match between load and target WUS is practically impossible, but you should strive to have all the key WUS components within 5-15% of each other.

## 5. Conclusion

Load testing is the most effective way to gauge a Web site's capacity and scalability, but load tests that don't simulate real scenarios can be dangerously misleading. In this paper we introduced an approach for planning, developing, executing and validating load scenarios that can accurately replicate real loads.

Our systematic method for standardizing, collecting, organizing and comparing load testing variables makes it possible to approach load testing in a rigorous, scientific and repeatable manner. The huge numbers and ranges of variables involved in Web site load testing will always present challenges and surprises. This framework gives load testing practitioners the necessary tools to study, understand and control these variables. Some amount of guesswork will always be required, but we firmly believe that techniques like the Web site Usage Signature can be used to reduce guesswork significantly and achieve results that are more reliable.