PAESSLER®

The Basics of Webserver Performance Testing

by Dirk Paessler

Why Testing?

Most web sites and web applications run smoothly and correctly as long as only one user (e.g. the developer) is using it. But what happens when thousands of users access the website simultaneously?!

The Business View

Today many websites have a serious business background and are mainly run to sell a service or a product to make money.
Everybody running a business on the web has seen statistics how fast a customer clicks away to your competitor if your server reacts too slow or not at all.
Aren't you cancelling the checkout of a shopping cart if the webserver does not react within 5-20 seconds?! If the visitors to your web site know about your slow website before you do, it may cost you a lot of money!
For the owner of a website this means: Test and monitor your website!
Not only functionality testing (does the website do what it is meant to do) and usability testing (does the user understand what is going on in his browser window) but also performance testing (does the user get results in an acceptable time) is important.
Your goal must be to ensure your customer always gets the right answer

to his mouseclick within a few seconds at anytime! Ensure that 95% of requests are processed in less than 10 seconds.
Jakob Nielsen[1] (see below) suggests the following views:

Download Time	User's View
< 0.1 s	User feels that the system is reacting instantaneously.
< 1.0 s	User experiences a slight delay but he is still focused on the current website.
< 10 s	This is the maximum time a user keeps focused on a website, but his attention might already be distracted.
> 10 s	User is most likely distracted from the current website and looses interest.

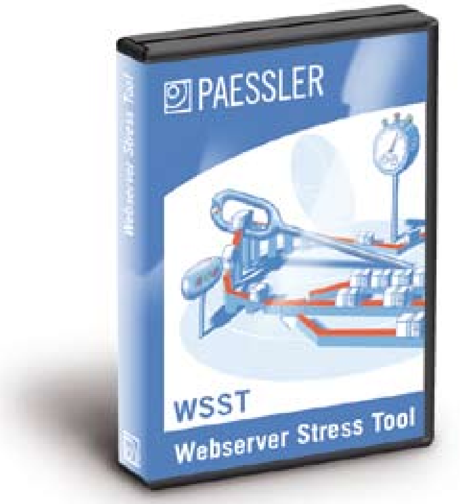
Using Webserver Stress Tool you are able to know if you have a performance problem due to load problems and you will be able to drill down to the problem!
By the way: Paessler also offers a solution to monitor your webserver after it has been deployed to the web, visit http://www.paessler.com for details on our product PRTG!

The Technical View

Moving the focus of our thoughts from the business to the technical viewpoint we discover the following questions that need to be answered to ensure that the goals from the business viewpoint are met:

- Is your webserver prepared for as many users as you expect?!
- Is your webserver prepared for increasing load over the month/years?
- Does your webserver survive a massive peak of users (e.g. if your website is mentioned on national TV)
- How many users can your server handle before users will get error messages or timeouts?
- How long does it take for a visitor to receive a page under no load, little load and heavy load?!
- Does your application or shopping cart support simultaneous users at all?!
- Are the CGIs, databases or scripts programmed as fast as possible and do they interact with each other correctly under heavy load?!
- Is the hosting service doing a good job?!
- Is the internet connection bandwidth sufficient?!
- Is that expensive hardware too small or too powerful?!

You need to find out all of these answers!



Using Webserver Stress Tool you are able to know if you have a performance problem due to load problems

Performance-, Load- or Stress-Testing?

These three words are often used synonymously, but there are differences:

Performance-Tests

Are used to test each part of the webserver or the web application to find out what parts of the website are slow and how you can make them faster. Think of it as testing of single webpages/scripts.
Webserver Stress Tool supports this type of test by the ability to running some 20-40 simultaneous requests on one url and recording the average time until the requests are answered. By changing the programming code you will be able to find the points you have to take care of to gain more website performance. Usually this type of test is run without requesting all the images etc. in a webpage to get more exact results about the script processing

Load-Tests

Are done by testing the website using the load that you expect to have on your site. This is something like a real world test of the website.
First you have to define the maximum request times you want the customers to experience, this is done from the business and usability point of view, not from a technical point of view. At this point you need to calculate the impact of a slow website on your company sales and support costs.
Then you have to calculate the anticipated load and load pattern for your website (see next section for details on load calculation!) which you then simulate using Webserver Stress Tool.
At the end you compare the test results with the request times you wanted to achieve. You know you have some work to do when requests take longer than time out or even generate error messages!

Stress-Tests

Are simulated brute force attacks with excessive load on your webserver. In the real world situations like this can be created by a massive spike of users — far above the normale usage e.g. caused by a large referrer (imagine your website being mentioned on national TV).
The goals of stress tests are to learn under what load your server generates errors, whether it will come back online after such a massive spike at all or crash and when it will come back online.

Calculation of load and load pattern

This is probably the trickiest question in conducting performance tests on websites (you have already found the answer to all the other questions in Webserver Stress Tool).
First, remember that there is a difference between users, transactions, pageviews and hits:

- One user creates several (trans-)actions (e.g. visit Homepage, search product, view product's details, buy a product etc.)
- One (trans-)action creates several pageviews (e.g. add products to shopping cart, go to checkout, enter creditcard etc.)
- One pageview creates several hits (e.g. framesets, images, applets etc. of one webpage)
- If you already have your website online it is a good start to use a good logfileanalyzer on your logfiles. You can find out how many people access the site per day and per hour, what pages/scripts are used how often etc.

If you are working on a new website you have to find the numbers and pattern yourself. One way to define the load pattern could be:

- Step 1: From a business point of view come up with the target number of users.
- Step 2: Define a couple of different „model users" (e.g. teen kid, mature business customer, senior citizen etc.) and surf from their point of view through the website. Track the webpages they access and gather these stats.

At the end you should have a list of URLs and their frequency of use.
There is a difference between the number of users using your website at one time and users sending simultaneous requests to your website!
If you have 200 simultaneous users clicking a link every 20 seconds you have 600 clicks per minute (3 per user per minute) or 10 simultaneous users per second. This number would be the number you simulate with Webserver Stress Tool.
Remember to take into account that additionally to the average load of your website there could be spikes in usage generated by marketing activities (e.g. newsletters, banners, TV commercials etc.) or simply by seasonal situations (e.g. Valentines Day for a flower shop website).
Now feed this data into Webserver Stress Tool and hit Start Test and keep your fingers crossed!

When does testing start?

The answer is simple: You cannot start performance testing early enough when building web applications!
It is even a good idea to start performance testing before a line of code is written at all! Early testing the base technology (network, load balancer, application-, database- and web-servers) for the load levels you plan for

can save a lot of money when you can already discover at this moment that your hardware is to slow. Also the first stress tests can be a good idea at this point.

The costs for correcting a performance problem rise steeply from the start of development until the website goes productive and can be unbelievable high for a website already online.

During software development all programmers should have access to performance test tools from an early stage on to test their own code for performance and for parallel execution problems (e.g. caused by database locks or other mutexes). Each developer should be responsible not only for the functionality of his code but also for good performance of his pages!

As soon as several webpages are working the first load tests should be conducted and from there on should be part of the regular testing routine each day or week or for each build of the software.

How to start?

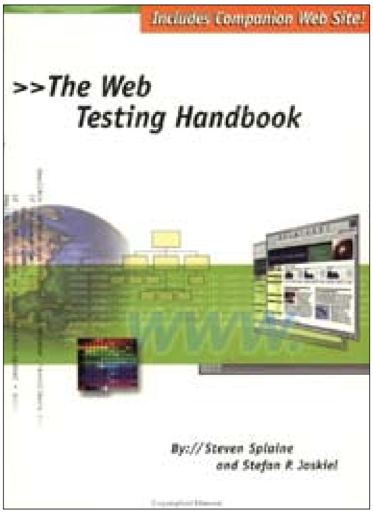
We suggest to download the Trial Version of our product Webserver Stress Tool and find the answers to the questions above asap!

Another good first step could be to take our Guided Tour of Webserver Stress Tool.

Last but not least the book mentioned below is a good way to start also!

[1] Jakob Nielsen, **Designing Web Usablity**, New Riders Publishing, 2000
The Web Testing Handbook
by Steven Splaine, Stefan P. Jaskiel, Alberto Savoia

The Web Testing Handbook is the definitive resource for testing Web sites and Internet-based applications. Many developers and testers are making the transition from traditional Client/Server, PC, and/or Mainframe systems to testing rapidly changing Web sites and applications. The Web Testing Handbook can help make this transition easier by explaining these new technologies and suggesting test cases and techniques that can be included in a Web site‘s Functional, Performance, Compatibility, and Usability test plans.


Includes Companion Web Site!
>>The Web Testing Handbook
By:// Steven Splaine and Stefan P. Jaskiel

The Web Testing Handbook is the definitive resource for testing Web sites and Internet-based applications

PAESSLER®
the network monitoring company

Paessler AG · Burgschmietstraße 10
90419 Nuremberg · Germany
www.paessler.com · info@paessler.com