

Web Performance Testing - Test objectives and Real Life Monitoring

Articles

Posted by:

Posted on : 2010/1/26 8:00:00

Web Performance Testing is executed to provide accurate information on the readiness of an application through testing the web site and monitoring the server side application. This article describes different techniques and aspects to be considered when performing Performance Testing on Web Applications.

Author: Robin Bortz, QualiTest Group, <http://www.qualitestgroup.com/>

Today, when a telephone system is installed in a new neighborhood, it will always work perfectly. The phone will ring, the line will be OK and people can communicate with each other even if all the neighbors are speaking at the same time. This is because overtime algorithms have been developed so it is known how much hardware is needed to support a specific area. Web technology is very different. There can be a different number of servers, different amounts of resources per server, different and new technologies, many different parameters can be configured to improve performance and naturally code can always be improved. As the number of mission critical applications Web sites increases, so too there is an increasing need to test these applications in order to ensure and guarantee the business and marketing needs. As the web does not have the same loyalty as traditional trading, poor performance will let users easily move over to the competition.

What is Web Performance Testing?

Web Performance Testing is executed to provide accurate information on the readiness of an application through testing the web site and monitoring the server side application. This is done by simulating load as close as possible to the real conditions in order to evaluate if the application will support the expected load. That allows you to guarantee system performances and to identify and help in fixing possible issues, identifying possible bottlenecks and providing useful advice about how to fix problems (tuning of system parameters, modification of software or hardware upgrade).

Testing is an art and science and there may be multiple objectives for testing. It is important to know what the objectives are before testing can begin. Naturally there is always interest in the different page times and specifically the slowest ones. These slow pages can point to the bottlenecks in the application. An objective may also be to make sure pages are downloaded within a specific time. Companies always want to know what this desired time is but it differs from application to application. It is desired to know the number of concurrency that the application can support and at how many users there is a decline in application performance. Does the page time reduced from 1 to 3 seconds which may be acceptable or is it greater than 30 seconds? Another objective may be to know the number of users which can crash the application.

Type of Load Testing

Once the objectives have been set, then the type of testing can be determined. **Smoke Test:** A simple quick test, to check if the application is really ready to be tested. This is not normally mentioned, but without it, much time and resources are wasted.

Load Test: This is the simplest form of performance testing. A load test is usually conducted to understand the behavior of the application under a specific expected load. This load can be the expected concurrent number of users on the application performing a specific number of transactions within the set duration. This test will give out the response times of all the important business critical transactions. If the database, application server, etc are also monitored, then this simple test can itself point towards the bottleneck in the application.

Stress: This testing is normally used to break the application. Double the number of users is added to the application and the test is run again until the application breaks down. This kind of test is done to determine the application's robustness in times of extreme load and helps application administrators to determine if the application will perform sufficiently if the current load goes well above the expected load.

Spike Testing: Spike testing, as the name suggests is done by spiking the number of users and understanding the behavior of the application, whether it will go down or will it be able to handle dramatic changes in load.

Endurance Testing (Soak Testing): This test is usually done to determine if the application can sustain the continuous expected load. Generally this test is done to determine if there are any memory leaks in the application.

Model Your Real Life Conditions

If you want to get an accurate idea of how your Web site is going to perform in the real world, then you need to model the conditions that your site will experience. That means different work flows representing different users and different actions per flow. This will give the confidence that the application will react in a similar way as to the load conditions. In order to model the test correctly and accurately, the following aspects should be considered:

Configuration management: You must make sure the testing environment is ready for the load that will be generated. Ensure the testing hardware is dedicated to the testing effort. Ensure the database is not changed and that it is populated as it would be in production. When changes are made to the application, make these on at a time between test runs.

Application Work Flows (Scenarios): This is most critical aspect of successful application testing and input should be provided by all teams which have the relevant knowledge. It could be marketing, sales or product management. In the case of existing applications, it should be modeled on known usage patterns and log files or other analytic tools which may also be used. Load testing is not functional testing and not every link should be clicked upon. However, the recorded scenarios should reflect typical usage of the application and these scenarios run according the appropriate percentages.

Sleep Times: User spend time on web pages reading, filling in a form, watching a media clip or

interacting with a client side object before the next click which sends a server request. This time parameter is extremely influential on the load created. People interact at different times, some read all the content and some scan briefly over the text. People using the same site repeatedly may quickly click through a number of screens. The Sleep time in the test should therefore be realistic often defined as random number within a specific range based on what is possible on a specific page. Also note that running a test without Sleep times is extremely difficult to quantify in terms of real users. It may be a good stress test but will not provide the answers needed for an accurate performance test. If the tools you are using do not have recorded Sleep time, try to obtain this important information from server logs.

Usage Patterns of Usage in web applications: The web is not a 9:00 – 5:00 interface for doing business. Web sites may have a global reach or may be accessed during typical work hours. Some applications are continually accessed whilst other may be accessed once per week or less than that. For ecommerce, users use this always open interface to shop at any time in the day. There are however times where there may be less or more users. And there may be great variations as well as extreme peaks of load on the application. A marketing campaign for Christmas shopping or a large sporting event with web access can create such peaks. An educational application may have the greatest load at 8:00 in the morning when an online lesson begins. These patterns and peaks should be tested for in order to be ready for changing load as well as peak load periods.

Browsers: The http header sent together with a web request may result in a different response from the server. A request from a mobile telephone may return a different result set than a browser. May require different server side activities and return a different result set.

Connection speeds: In early web testing when homes were connected via a modem at the transmission speed 28.8 Kbps simulating such a slow speed actually made it easier for web applications to respond. The slow times were a result of the slow connection and not the slow response of the application. Today with ever increasing access speeds and fiber to the home, the load should be generated at such speeds and also generated at a mix of different speeds simulating different populations.

Different, Remote Geographic Locations: Typically web performance testing is done in a lab environment. However different locations around the world may take longer due to the distance or web traffic at peak hours. There are ways to test this to get a true report of client side response times. Some tools have a remote roaming client which can be installed anywhere in the web to run at the same time as the load test. If it is an internal application, tests can be run from various locations or branches of the company. You can request from someone at a remote location to access the location whilst testing. This is generally recommended anyway to do even in the lab when testing. Recently with the development of Cloud computing there are some new exciting possibilities to resolve this problem of remote geographical locations as well as the ability to generate large loads with enough bandwidth. Using the services of a third-party company that specializes in testing a Web site from different locations around the world.

Poor performance and slow reaction times will both have negative consequences on companies. These could be decreased sales in web commerce, customer abandonment, harmed reputation and wasted resources. Good test planning where the objectives have been set and the correct tests are based on these objectives will ensure that web application will perform as expected. Once these tests are run as close as possible to real life scenarios, you can be sure that the small insurance paid for the application testing it is a great investment.