

Chapter 1 : WebLOAD - Performance Testing Tool Overview - QAInsights

WebLOAD tool (get it here) is definitely worth looking at - particularly if you need the power of scripting to handle more challenging load testing environments. It offers a breadth and depth of features and on the whole, should be closely examined if looking for an alternative to LoadRunner.

What is Load Testing? Without it, your application could fail miserably in real-world conditions. It examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end users. This tool is typically applied when a software development project nears completion. How It Works A load test can be done with end-to-end IT systems or smaller components like database servers or firewalls. It measures the speed or capacity of the system or component through transaction response time. When the system components dramatically extend response times or become unstable, the system is likely to have reached its maximum operating capacity. When this happens, the bottlenecks should be identified and solutions provided. Stress Testing Performance testing aims to examine systems behavior and performance, specifically, response time, scalability, speed, and the resource utilization of the software and infrastructure. Load testing is a subset of performance testing. LoadView measures performance in real browsers not headless phantom browsers. Lastly, LoadRunner can manage thousands of users at the same time. For a comprehensive list of many of the tools available today, check out our list of 50 load testing tools. Possible Use Cases and Examples Below are a few of the many possible use cases: Airline Website During Promotion Period. Evaluating a website of government agencies, particularly during the filing date of income tax returns when traffic will spike. Running numerous concurrent requests on a server, thus, subjecting the server to a massive volume of traffic. The nonstop transfer of multiple files to and from a hard disk. This could take the form of measuring the speed of laptop with the transfer of 8GBGB files to and from a laptop with speeds of 1 mbps. Downloading a huge volume of large files from a company website to test performance. Requesting various jobs on printers in a queue simultaneously. Benefits of Load Testing Benefits include the discovery of bottlenecks before production, scalability, reduction of system downtime, improved customer satisfaction, and reduced failure costs. Discovery of bottlenecks before deployment. Evaluating a piece of software or a website before deployment can highlight bottlenecks, allowing them to be addressed before they incur large real world costs. Enhance the scalability of a system. This can aid in determining infrastructure needs as the system scales upward. Reduced risk for system down time. It can be used to ferret out scenarios that can cause a system to fail. This makes it a great tool for finding solutions to high-traffic problems before they arise in the real world. Identifying concerns at the earliest stage possible, especially before launch, decreases the cost of failures. By contrast, after-launch failures can incur exponentially greater costs. Best Practices for Load Testing Identify business goals. A strong understanding of future goals for scope and volume will draw clear guidelines to inform the process. Determine key measures for the application and web performance. Agree on criteria to track. Some criteria include response times, throughput, resource utilization, maximum user load, and business performance metrics. Choose a suitable tool. Select a tool that best caters to your needs. Create a test case. In writing a test case, make sure both positive and negative scenarios are taken into account. Test cases must be accurate and capable of being traced to requirements. Consider different types of deployments you might want to test. Create configurations similar to typical production. Test different system capacities like security, hardware, software, and networks. During these tests, the system will ultimately fail. One key goal is determining what volume results in failure, and spotlighting what fails first. Always keep end users in mind. The satisfaction of customers and site visitors is crucial to the achievement of business metrics. This plays into their willingness to revisit a site or re-access an application. Additional Resources and Tutorials.

Chapter 2 : Load Testing & Website Performance Testing Tools - RadView

How to do Performance Testing with Selenium using WebLoad Tool. Combining Selenium with WebLOAD for larger-scale load testing Tussen de Vaarten, Almere, PG (Netherlands) + support@calendrierdelascience.com

It can be automated and manual. It almost goes through the normal testing process but the testing techniques may vary for mobile devices. Importance of Mobile Application Testing As we discussed in the last chapter that Mobile market is booming, mobile websites and Apps are exploding like anything in the market. Each and every business is trying to get the maximum benefits out of this mobile technology. With the exponential increase for mobile usage has led to mobile testing. Businesses are focusing on creating mobile strategies and roadmap before implementing the application for their users. It became essential to develop an app with effective features and functionalities which are beneficial to the customer. But it is more critical to have an effective testing plan and strategy before any app is developed. As client is usually in millions for a certain product and a product with bugs is never appreciated. It often results in monetary losses, legal issue and irreparable brand image damage. Challenges in Mobile Application Testing Performing Mobile Testing is not an easy task, as it is different and much more complex from enterprise testing and website testing. Mobile applications have a smaller footprint on the virtual machine as opposed to a desktop application. Mobile Testing needs to be performed on variety of software platform, versions, hardware, network condition and what makes it more complex are the various devices, fast changing mobile OS, the frequent introduction on new devices, and unpredictability of simulators. Different range of mobile devices with different screen sizes and hardware configurations like hard keypad, virtual keypad touch screen and trackball etc. Different versions of operation system like iOS 5. Frequent updates like android- 4. This is to assure the quality of mobile devices. The device including the internal processors, internal hardware, screen sizes, resolution, space or memory, camera, radio, Bluetooth, WIFI etc. The applications that work on mobile devices and their functionality are tested. Even in the mobile applications, there are few basic differences that are important to understand: A native application is created for use on a platform like mobile and tablets. These applications get installed on the mobile device and often delivered from the App Store. Web applications as one accessed using a browser on the mobile device or server-side apps to access websites on mobile using different browsers like chrome, Firefox by connecting to a mobile network or wireless network like WIFI. These are combinations of native app and web app. Hybrid embeds a browser in a native application. These also run on devices, online or offline. It is enormously challenging to verify enterprise applications across myriads of handsets with different screen sizes and operating system flavors. Therefore, testing teams need to find better and more cost-effective solutions to avoid any compromise on quality. This tutorial on Mobile Application Testing will cover QA challenges and discusses mobile testing strategies, mobile testing types and manual mobile testing tools.

Chapter 3 : WebLOAD - Website and application performance testing

Within the endless number of performance testing options available today, WebLOAD is probably the industries best kept secret. It is part of the earliest group of load testing tools that were launched in the 90's.

Click on Thread Group The Thread Group has three particularly important properties influence the load test:

- Number of Threads users: The number of users that JMeter will attempt to simulate. Set this to 50
- Ramp-Up Period in seconds: The duration of time that JMeter will distribute the start of the threads over. Set this to The number of times to execute the test. Leave this set to 1. This is particularly useful if we want to send multiple HTTP requests to the same server as part of our test. Setting the server here makes it the default server for the rest of the items in this thread group. Note that you do not need to specify the server in this item because it was already specified in the HTTP Request Defaults item. Every thread will perform all of the requests in this test plan. There are a variety of listeners available, and the other listeners can be added by installing plugins. We will use the Table because it is easy to read. First, save the test plan by clicking on File then Save, then specify your desired file name. Then select on View Results in Table in the left pane, then click Run from the main menu then click Start or just click the green Start arrow below the main menu. You should see the test results in the table as the test is run like: Interpreting the Results You will probably see that the Status of all the requests is "Success" indicated by a green triangle with a checkmark in it. After that, the columns that you are probably most interest in are the Sample Time ms and Latency not displayed in example columns. The number of milliseconds that elapsed between when JMeter sent the request and when an initial response was received
- Sample Time: This is a reasonable response time for a basic homepage which was about 55 KB. If your web application server is not struggling for resources, as demonstrated in the example, your Sample Time will be influenced primarily by geographical distance which generally increases latency and the size of the requested item which increases transfer time. Your personal results will vary from the example. So, our server survived our simulation of 50 users accessing our 55 KB WordPress homepage over 10 seconds 5 every second , with an acceptable response. In the Thread Group item in the left-pane, change the Number of Threads users to Now click View Results in Table, then click Start. On our example server, this results in the following table: As you can see, the sample time has increased to nearly a second, which indicates that our web application server is beginning to become overburdened by requests. Log in to your web server via SSH and run top: You should see the resource usage increase: In order to meet the demands of this simulation of 80 users in 10 seconds, we need to either increase our CPU or optimize our server setup to use less CPU. If you are curious, you may adjust the number of threads in the test to see how many your server can handle before it begins to exhibit performance degradation. In the case of our 1 CPU droplet example, it works fine until we use 72 threads over 10 seconds. Conclusion JMeter can be a very valuable tool for determining how your web application server setup should be improved, to reduce bottlenecks and increase performance. Now that you are familiar with the basic usage of JMeter, feel free to create new test plans to measure the performance of your servers in various scenarios. For example, JMeter can be configured to simulate a user logging into your application, client-side caching, and handling user sessions with URL rewriting. There are many other built-in samplers, listeners, and configuration tools that can help you build your desired scenario. Additionally, there are JMeter plugins to enhance its functionality that are available for download at [http:](http://)

Chapter 4 : How To Use Apache JMeter To Perform Load Testing on a Web Server | DigitalOcean

Load Testing Tools: NeoLoad: NeoLoad is the enterprise-grade load testing platform designed for Agile and DevOps. NeoLoad integrates with your continuous delivery pipeline to support performance testing across the complete software life cycle - from component to full system-wide load tests.

Join For Free Senu is an open source monitoring event pipeline. Performance testing is an important component of the application development lifecycle. Without it, your application could fail miserably in real-world conditions. It examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end users. This tool is typically applied when a software development project nears completion. How It Works A load test can be done with end-to-end IT systems or smaller components like database servers or firewalls. It measures the speed or capacity of the system or component through transaction response time. When the system components dramatically extend response times or become unstable, the system is likely to have reached its maximum operating capacity. When this happens, the bottlenecks should be identified and solutions provided. Load testing is a subset of performance testing. LoadView measures performance in real browsers not headless phantom browsers. Lastly, LoadRunner can manage thousands of users at the same time. For a comprehensive list of many of the tools available today, check out our list of 50 load testing tools. Possible Use Cases and Examples Below are a few of the many possible use cases: Airline Website During Promotion Period. Evaluating a website of government agencies, particularly during the filing date of income tax returns when traffic will spike. Running numerous concurrent requests on a server, thus, subjecting the server to a massive volume of traffic. The nonstop transfer of multiple files to and from a hard disk. This could take the form of measuring the speed of a laptop with the transfer of 8GBGB files to and from a laptop with speeds of 1 Mbps. Downloading a huge volume of large files from a company website to test performance. Requesting various jobs on printers in a queue simultaneously. Benefits of Load Testing Benefits include the discovery of bottlenecks before production, scalability, reduction of system downtime, improved customer satisfaction, and reduced failure costs. Discovery of bottlenecks before deployment. Evaluating a piece of software or a website before deployment can highlight bottlenecks, allowing them to be addressed before they incur large real world costs. Enhance the scalability of a system. This can aid in determining infrastructure needs as the system scales upward. Reduced risk for system down time. It can be used to ferret out scenarios that can cause a system to fail. This makes it a great tool for finding solutions to high-traffic problems before they arise in the real world. Identifying concerns at the earliest stage possible, especially before launch, decreases the cost of failures. By contrast, after-launch failures can incur exponentially greater costs. Best Practices for Load Testing Identify business goals. A strong understanding of future goals for scope and volume will draw clear guidelines to inform the process. Determine key measures for the application and web performance. Agree on criteria to track. Some criteria include response times, throughput, resource utilization, maximum user load, and business performance metrics. Choose a suitable tool. Select a tool that best caters to your needs. Create a test case. In writing a test case, make sure both positive and negative scenarios are taken into account. Test cases must be accurate and capable of being traced to requirements. Consider different types of deployments you might want to test. Create configurations similar to typical production. Test different system capacities like security, hardware, software, and networks. During these tests, the system will ultimately fail. One key goal is determining what volume results in failure, and spotlighting what fails first. Always keep end users in mind. The satisfaction of customers and site visitors is crucial to the achievement of business metrics. This plays into their willingness to revisit a site or re-access an application. Learn moreâ€”download the whitepaper. Read More From DZone.

Chapter 5 : 11 Best Performance/Load Testing Tools of

4 This guide will help you plan, design and execute efficient load testing using WebLOAD. It will help you: Understand the methodology and essential steps for.

WebLOAD includes extensive scripting capabilities, massive load generation locally or on the cloud , analytics and reporting, and built-in integrations with APM solutions, functional testing tools and open source platforms. This is quite useful, since often pages change between the time you record and play back your script, which in turn saves you the pain of editing the script to adjust for modified resources. WebLOAD provides multiple ways to edit and enhance your script – starting of course, with direct editing and debugging of the script itself. Several other options for manipulating the script include HTTP configuration options for browsers, caching, cookies, etc. Automatic Correlation of Dynamic Values: The Achilles heel with automatic correlation is its accuracy, as well as the effort required to tweak and make edits. The automated correlation process seems to be quite accurate with WebLOAD, in a sense that it correctly identifies many types of dynamic values within URLs, headers and cookies. The code it generates is relatively concise so that it can be fine-tuned quite easily and does not require lengthy manual edits. The ability to work with Java objects provides the flexibility to address more unique testing requirements, such as homegrown applications. By activating a real best place to buy modafinil online us browser, you are able to measure its behavior and examine the real user experience while your system is under load. During execution WebLOAD measures performance statistics that are collected by the browser, which enables viewing side by side the load size, Hits per Second of the test and the real user experience on the same graph. To generate virtual user load you can use both on premise machines and the cloud. You can define a variety of scripts, browsers, and network characteristics in a single test run. Using a scheduler, you can control the load build-up to mimic real-life conditions, using linear, steps, and increments of virtual users. Following a load test session, WebLOAD Analytics generates a set of reports and indicates places with threshold violations. You can choose from over 80 different reports and graphs to analyze test results from different angles. Reports are customizable letting you identify specific areas and compare metrics across load sessions. You can choose from a variety of templates, apply filters and control the layout. An additional analysis tool, the Web Dashboard, lets you view performance test results from any web browser or mobile device. Sign up just providing your email address below: Enter your email address: Check email in your inbox for confirmation to get latest updates Software Testing for free.

Chapter 6 : Testing Thoughts: WebLOAD Open Source

Performance testing aims to examine systems behavior and performance, specifically, response time, scalability, speed, and the resource utilization of the software and infrastructure. Load testing is a subset of performance testing.

In-house expertise, best testing practices, product announcements, etc. Case studies Patient education - a learning website where people can train their skills. Load balancer - an insurance service with up to 10, concurrent user sessions. CRM system - a universal bank with broad range of products and services. WAPT tool is a perfect combination of efficiency and flexibility with ease of use. The product is backed by a team of professionals with over 14 years of experience in load testing. If you need assistance working with WAPT, our experts are ready to provide it to you from the very first day you run the demo version. We offer a lineup of components to address all technology requirements and scale the test volume up to 1M of concurrent virtual users. WAPT employs a simple test design approach by recording real life web sessions from a browser or a mobile application. It represents each recorded session as a sequence of HTTP requests, which is easy to understand and modify through a convenient user interface. The emulation is based on spawning multiple copies of your original session. Dynamic values inserted into each new session make it unique and the whole emulation realistic. Most of these insertions are done automatically, so you only need to specify how to handle application-specific data. Scripting is not required, but may be used to handle extraordinary complex cases. This approach lets you create tests really fast. A simple test scenario can be implemented within minutes after a short acquaintance with the tool. Test results are presented for your review and analysis in form of descriptive graphs and report tables. They show how the performance of your system depends on the load being generated in different test periods. Additional information can be found in the comprehensive test logs that are easy to read and navigate through in the integrated viewer. Ability to adjust the execution options on the fly in order to isolate each possible system bottleneck will let you complete the testing in shortest time. Our clients choose WAPT for a number of reasons, but we believe that those listed below differentiate it from most commercial alternatives as well as from popular free tools: Ease of use from learning to the full employment in the QA process. Rich functionality ensuring applicability for all types of tests. Low cost of ownership with on demand option for cloud users. High efficiency and scalability of load generation. Outstanding technical support and additional services. For those customers who look for high capacity load tests or fine tuning of complex test scenarios we offer the Pro version.

Chapter 7 : Load Testing Services & Tools - Web Performance

Below is a comprehensive list of most widely used performance testing tools for measuring web application performance and load stress capacity. These load testing tools will ensure your application performance in peak traffic and under extreme stress conditions.

Chapter 8 : JMeter and Performance Testing for DevOps | BlazeMeter

WebLOAD is an enterprise-scale load testing tool which features a comprehensive IDE, Load Generation Console, and a sophisticated Analytics Dashboard. Application: WebLOAD has built-in flexibility, allowing QA and DevOps teams to create complex load testing scenarios thanks to native JavaScripting.

Chapter 9 : TOOLSQA | Free QA Automation Tools Tutorials

WebLOAD Free Edition Experience enterprise scale load testing now! WebLOAD's IDE offers correlation, parameterization, response validation, messaging, native JavaScripting and debugging.