



Load Testing with the Cloud

A Neotys White Paper

Table of Contents

Overview	3
Benefits of Load Testing with the Cloud	3
Perform Large Scale Tests	3
Perform More Realistic Tests.....	4
Save time and Reduce Costs with Pay-as-you-go	4
Challenges of Load Testing with the Cloud	4
Isolating Root Causes	4
Reproducing Tests.....	5
Conducting Small Scale Tests.....	5
Testing Inside the Firewall.....	5
Best Practices	5
Employ a Two-Stage Process	5
Use Different Cloud Providers	6
Secure Your Data.....	6
Tune Load Generators	7
Monitor Your Servers	7
How to Choose a Cloud Testing Solution	7
Integration with the Cloud Platform	8
Realistic Tests.....	8
Unified Lab Testing and Cloud Testing	9
Ease of Use	9
Analysis, Monitoring, Scheduling, and Reporting	10
Support for Web Technologies	10
Conclusion	10
About Neotys.....	11

Overview

Everyone, it seems, is talking about *the cloud*. End users see the cloud as a way to access their documents and applications remotely from anywhere and from any device. IT Managers see the cloud as a means of rapidly adapting their infrastructures as needed via virtualization and a pay-per-use-model. But what about load testing engineers? Can they seize the opportunities afforded by the cloud to better test the performance of web applications?

As with past overhyped trends in IT, it is important to see past all the talk and look for concrete ways to take advantage of this new technology's flexibility and scalability to save time, reduce costs, and improve the way your organization works.

This white paper describes how the cloud is revolutionizing load testing and the advantages it provides in many situations for ensuring your web applications performs well in production. It also covers the limitations of a test strategy that relies *solely* on cloud-based testing, highlighting the need for a complementary internal load testing solution. After describing several best practices for load testing in the cloud, this white paper outlines key capabilities to look for in a load testing solution. Without the right tools in place, simply moving your testing activities to the cloud will likely not deliver the results necessary to justify the move. Understanding how to apply the right tools and practices to make the most of the cloud is fundamental to cloud-based testing and vital to ultimately going live with total peace of mind.

Benefits of Load Testing with the Cloud

Load testing with the cloud enables testing teams to take a big step forward in conducting more efficient and more realistic large scale tests. In addition, it enables organizations to realize significant savings in cost and time made possible by cloud technology.

Perform Large Scale Tests

More and more, today's web applications are experiencing sporadic surges in traffic. These traffic spikes can have many causes, including a new advertising campaign, an online article, a seasonal sale, and buzz on Twitter or other social media. If your application is unable to handle the increased load, you run the risk of lost business opportunities and potential damage to your brand.

Generating the load for large scale tests to mimic these unanticipated spikes in production traffic, however, typically requires tens or even hundreds of machines. Purchasing and configuring these systems requires a significant investment of time and money. Once acquired and used for the immediate load testing need, the machines may sit unused for long stretches until they are needed for the next large scale load testing project. With the cloud, you can rapidly set up as many load generating machines as you need, on demand.

Perform More Realistic Tests

When testing a web application using machines inside your firewall, you're not testing the entire delivery chain. Unless all of your end users will also be within your firewall, such tests are inherently limited and may fail to reveal all performance issues.

With the cloud, you can execute load tests that access your web application as your users will—from outside of your firewall—and validate all components of the delivery chain, including the firewall, DNS, network equipment, and ISP. These tests are more realistic, and they enable you to evaluate the real-world effects of third-party components, such as content delivery networks, analytics servers, and ad servers.

Your users won't all be accessing your app from the same fixed location across the same network, so a *realistic* load test cannot be completed from a single location. That's why it is important to test your application and its components from different locations and geographic regions and assess its performance as network bandwidth and latency changes.

Save time and Reduce Costs with Pay-as-you-go

When load testing with the cloud, there is no need to spend weeks setting up and configuring dozens of real machines. You can create and configure the machine image you need once and then replicate it in the cloud as many times as needed. Often, the cloud testing provider will automate this process as well, saving you even more time.

Further, the substantial up-front costs of purchasing and maintaining machines that may be used only infrequently are eliminated with the cloud. Using the pay-as-you-go model, you can rapidly set up the testing infrastructure you need, when you need it, and only for as long as you need it. From a business standpoint, the cloud lowers total cost of ownership, while increasing flexibility.

Challenges of Load Testing with the Cloud

The advantages to load testing with the cloud are clear, but internal testing still has its place in the overall test plan, particularly when testing from outside the firewall is not feasible. Internal testing also helps you to isolate effects that are due to your own application or infrastructure from those that are outside your firewall and potentially beyond your control.

Isolating Root Causes

When load testing uncovers a problem, the next step is identifying which layer in the delivery chain is causing the problem. You can use monitors to check performance metrics such as hits, average response time per request, and average bandwidth for each layer or piece of equipment in the chain. You can also employ application performance management (APM) solutions to identify bottlenecks in the code. These techniques work reasonably well when there is a single source for the performance slow-down.

When there are multiple problems, both inside and outside the firewall, it can be difficult to sort out the root causes because the symptoms are often commingled, making them difficult to isolate and resolve individually. For this reason it is important to have a cloud load testing solution that you can also apply within the firewall. You can then separate the problems that exist within the firewall from

those caused by layers outside it. The ability to test a subset of the delivery chain in this way makes it much easier to find the root causes of performance problems.

Reproducing Tests

Often, you need to precisely measure the effect of changes made to the application code or settings. For example, you may need to determine the performance improvement that results from resolving a specific defect or evaluate performance for a range of cache sizes and other settings.

With cloud load testing, such precise measurements are difficult because of variations in Internet traffic and bandwidth availability at the data center level. Such variations can make it almost impossible to duplicate conditions from day to day or even within the same day.

As with isolating root causes, this situation also highlights the need for internal testing, in which you can better manage the conditions of the test, stabilize the testing environment, and take precise measurements to get more reliable performance metrics for comparison.

Conducting Small Scale Tests

Not all load testing requires hundreds of load generators. Even applications that anticipate many thousands of concurrent users are initially tested with a small population. These smaller scale tests require only a few machines may be easier and less expensive to conduct internally if the machines have already been purchased and are available for use. These tests can be carried out within the firewall to conduct tests that require neither a heavy load nor the full delivery chain. To keep cloud expenses down, use available real machines when they can provide the information you need, and employ load testing with the cloud when necessary for larger scale, more realistic tests.

Testing Inside the Firewall

Of course, some testing use cases preclude the use of the cloud. If you're developing an enterprise web application that was not designed to be accessed from the Internet, then it is not a good idea to expose it outside the firewall solely for the purpose of load testing with the cloud. Likewise, if your pre-production environment is not set up to be accessed from the Internet, you'll want to have an internal testing solution that can be used within the firewall. Ideally, you want to use the same load testing solution for both internal testing and testing with the cloud, so that the scripts you developed for internal pre-production testing can be reused in production cloud-based testing. Using different tools for internal and cloud testing would not only require a rewrite of the scripts, it would also increase licensing and training costs.

Best Practices

The following best practices can help you maximize the advantages—and minimize the challenges—of load testing with the cloud.

Employ a Two-Stage Process

A two-stage process for load testing enables engineers to employ internal and cloud testing in the situations for which they are most effective and appropriate. In the first stage of the process you conduct internal tests with a medium load to quickly identify and resolve preliminary performance issues. Then, increase the load incrementally with one or more load generators in the test lab. After

the performance has been validated internally, proceed to the second stage, cloud-based load testing, for large scale tests that validate the entire delivery chain of the application.

This hybrid approach addresses the key challenges facing organizations that attempt to rely on testing from the cloud only:

- **It enables teams to isolate problems.** The source of any performance issue identified in the first stage is clearly within the firewall (because no other systems are involved in the test). It's easier to pinpoint and fix internal problems when they are not being compounded by other issues that originate outside the firewall.
- **It enables earlier testing.** With the two-stage process, you don't have to wait for the application to be deployed and accessible from the Internet to test it. You can test internally earlier in the application lifecycle, when defects are easier and less expensive to fix.
- **It enables reproducible tests.** With internal testing you have much more control over the environment, so you can precisely measure the effect of code or configuration changes on application performance.
- **It provides a better understanding of each major area of the delivery chain.** You can compare the results of the same test scenario run internally and from the cloud, to get a clearer picture of how the application server and network infrastructure contribute to overall response times.
- **It lowers costs.** Cloud testing is based on a pay-per-use model. When you can test internally on hardware you already have, you can reduce the amount of testing that you need to perform from the cloud and cut costs.

Use Different Cloud Providers

There are several advantages to using multiple cloud providers. First, it helps you test from more geographical regions, which provides more realistic results that capture the effects of various third-party servers and content delivery networks. Second, it is more scalable. For exceptionally large scale tests, you can engage multiple providers simultaneously to bypass limitations that a single provider may place on bandwidth or the number of machines in use. Third, it enables you to detect potential network issues at the cloud provider level. If test results from virtually all providers show acceptable performance, but you're seeing significantly worse performance from machines on a particular provider, then you can safely conclude that there is a temporary problem with only that provider, not your application. Load testing solutions that are locked into a single provider limit the test engineer's ability to conduct realistic, reliable, large-scale tests.

Secure Your Data

In internal pre-production testing, the data used is often fake—not actual customer or user information. Further, you can be reasonably assured that any real data used is safe because testing is being conducted within the firewall. This is not the case when testing from the cloud on production data. You'll need to take steps to ensure that any accounts, scenarios, detailed error messages, and other sensitive data involved in your tests are secured.

Encrypt the communication between your controller and load generators. This helps secure data sent to the load generators during the test (including account information) as well as the data that is retrieved (including error messages). If possible, use SSL to secure the communication between the

browser and the tested server. Lastly, ensure your load generators are secured with their own firewalls to protect them from outside threats.

Tune Load Generators

To ensure that your load generator machines in the cloud are capable of generating large loads, you must properly tune the system to support the creation of a high number of sockets and threads per process. Additionally, allocate an appropriate heap size for Java based load generators. The default settings for a typical machine allow all programs to share its resources fairly. In the case of load generators, the machine is dedicated to a single task, so you can improve performance by allocating a significantly larger share of the available resources to the load generation tasks.

Monitor Your Servers

Once you've identified a performance bottleneck, you need information to track down its root cause. This information should be gathered during the test by monitoring each component of the infrastructure including application servers, database servers, and so on. Specifically, you want to monitor both the system—including the operating system, disks, and network—and the server software—including connection pools, threads, cache hits, and indexes.

Linking all the information gathered during the tests with the tests themselves is much easier when the monitoring is integrated with your load testing tool. This enables you to correlate the response times and errors generated by load testing with the monitored data to track down the cause of problems quickly. A cloud testing solution that has no ability to monitor activity inside the firewall cannot integrate and correlate the tests it initiates from the outside with what is happening on the inside. With such a setup, test engineers will not have all the information they need to quickly identify the sources of performance problems.

How to Choose a Cloud Testing Solution

While all cloud load testing solutions will enable you to make use of the cloud in some way, comparatively few enable you to follow all of the best practices outlined here and capitalize on the opportunities that load testing with the cloud offers. A highway lets you travel faster than a side street, but the vehicle you use makes a big difference in how quickly and how reliably you arrive at your destination. In much the same way, load testing with the cloud offers clear advantages over traditional load testing, but the tools you use are even more important to the quality of your tests.

When considering a cloud testing solution, ask the following questions:

1. To what extent does the solution integrate with the cloud?
2. Will the solution enable us to conduct realistic tests?
3. Does the solution support unified tests inside and outside the firewall?
4. Is the solution easy to use, or will we spend weeks learning and configuring it?
5. Does the solution include full-featured reporting and decision-making modules to help our team make the most of the results?
6. Does the solution support the technologies we used to build the application?

Integration with the Cloud Platform

If you opt for a solution that is not integrated with one or more cloud platforms, you'll need to handle several time-consuming tasks on your own. First, you'll need to learn how each platform you'll be using works, including its limitations and constraints. Second, you'll need to build, test, and maintain your own virtual machine images.

Load testing solutions that offer integration with the cloud simplify and accelerate the steps needed to use the cloud infrastructure. These solutions offer one or more of the following advantages over non-integrated alternatives:

- Fast provisioning using preconfigured images. You can set up the infrastructure you need in minutes.
- Simplified security. All required protections are set up by default, including firewall, certificates, and encryption.
- Improved scalability. Leading load testing solution providers have negotiated with cloud providers to allow users of their software to employ more virtual machines (for the purpose of load testing) than are allowed by default.
- A unified interface for multiple cloud providers. Load testing solutions can hide provisioning and billing details, so you can take maximum advantage of the cloud in a minimum of time.
- Advanced test launching. You can save time and effort by defining and launching load generators in the cloud directly from the load testing interface.
- Advanced results reporting. Distinct results from each geographic region involved in the test are available for analysis.

Of course, few solutions include every one of these integration capabilities. Most solutions fall somewhere on the spectrum between little or no integration and full-featured integration with multiple cloud platforms.

Realistic Tests

Although testing from the cloud is, in many cases, more realistic than testing in the lab, simply moving to the cloud is not enough to ensure the most realistic tests. Real users often have access to less bandwidth than a load generator in a cloud data center. With a slower connection, the real user will have to wait longer than the load generator to download all the data needed for a web page or application. This has two major implications:

- Response times measured as-is from the cloud with virtually unlimited bandwidth are better than for real users. This can lead test engineers to draw the wrong conclusions, thinking that users will see an acceptable response time when in reality they will not.
- The total number of connections established with the server will increase, because on average, connections for real users will be open longer than connections for the load generator. This can lead to a situation in which the server unexpectedly refuses additional connections under load.

When choosing a load testing solution, look for one that provides a bandwidth simulation feature that limits bandwidth to ensure that the virtual users download the content of the web application at a realistic rate. This capability is particularly important when testing mobile applications, because mobile devices typically operate with less bandwidth than laptops and desktops.

Similarly, look for a solution that can parallelize requests. Modern browsers have the ability to parallelize HTTP requests as they retrieve a web page's static resources. These parallel requests require more connections with the server and can lengthen response times. Load testing solutions that do not parallelize requests are incapable of producing truly realistic performance tests for web applications.

Unified Lab Testing and Cloud Testing

Organizations that use only lab testing or only cloud testing are at a disadvantage. So are companies that use different tools for these activities.

A solution that supports lab testing enables test engineers to begin verifying the performance of an application internally, before it is ready to be made available via the Internet. This makes it possible to find and fix performance problems earlier in the application lifecycle. Such a solution also lowers cloud costs by enabling teams to conduct internal performance tests on existing hardware when available.

More importantly, a single solution that supports lab testing *and* cloud testing enables test engineers to reuse scripts for both kinds of tests, which saves time and effort. Reusing scripts also helps pinpoint performance problems that show up in cloud testing but not in internal tests. Lastly, a unified solution lowers licensing and training costs, and enables test engineers to use their existing skill set for both types of load testing.

Ease of Use

Testing, with its natural position towards the end of the application lifecycle, is almost always performed under tight time constraints. Delays in the requirements or implementation phases of a project usually result in less time for the test engineers to do their jobs. The pressure is on to deliver results as quickly as possible. This environment is no place for a tool that is difficult to use and configure.

In developing and executing performance tests (either internally or via the cloud) several key features go a long way in improving test engineer productivity, including support for:

- Easily launching the recording of a virtual user profile (preferably in one click).
- Defining advanced behaviors (with structures such as conditions and loops) via a graphical interface, complemented by the ability to use a scripting language (JavaScript, for example) for more complex cases.
- Automatic handling of dynamic parameters. This includes a set of correlation rules for well known server frameworks. Ideally, the solution will dynamically detect and handle custom parameters specific to your application.
- Sharing common script parts, such as login or logout transactions, between multiple virtual user profiles.
- Comparing results. Sifting through results to determine the effect of a particular application or infrastructure change can be a time-consuming and arduous task without a dedicated comparison tool.

This is a not exhaustive list of usability features that can help test engineers work more efficiently; rather it should be considered as a baseline of minimum required capabilities for an efficient load testing solution.

Analysis, Monitoring, Scheduling, and Reporting

Recording a virtual user profile and playing it back to get raw results is only the beginning of an effective performance test. You need tools to help you analyze the results (in real-time when possible), find the root cause of problems, and produce actionable results.

Real-time analysis enables you to detect and understand issues while the test is running. With real-time analysis, you don't have to wait for the test to finish to detect an issue, correct it, and restart the test. When testing in production, real-time analysis enables you abort a test if it threatens to affect the performance experienced by real users.

A comprehensive monitoring system is essential when you need to find the root causes of a problem. Predefined performance counters and threshold alerts based on industry best practices make it easy to define and analyze counters. For a nonintrusive solution that is easier to set up, look for a tool that supports agentless remote monitoring.

If your organization performs regular regression tests—and even if it doesn't—you may want to schedule performance tests and execute them automatically via the command line to complement functional testing. Regularly scheduled load tests with automatically generated reports can help organizations detect performance regression as soon as it starts to occur, which makes it easier to pinpoint and correct.

Lastly, reporting is a key capability and essential for communicating test results to others on the team, including management. Because reporting needs change, it is a good idea to keep your options open with a tool that supports multiple formats, including PDF, Word, HTML, and XML for integration with other systems.

Support for Web Technologies

To test Siebel applications or applications built with Adobe Flex, Microsoft Silverlight, Real-Time Messaging Protocol (RTMP), Oracle Forms, or AJAX push technologies you need a load testing tool with built-in support for the technologies you're using. Without this specialized support it can be very difficult, if not impossible, to effectively test the performance of your applications.

Similarly, the load testing solution you choose should provide support for the authentication mechanism employed by your applications, whether it is Basic, Digest, NTLM, or Kerberos. Otherwise, you will not be able to set up a virtual user profile that tests the application as a real person would use it.

Conclusion

The cloud is opening new opportunities to improve the scale and realism of load testing while saving time and lowering costs. Even with all its advantages, cloud testing itself cannot meet all the challenges facing performance test engineers. In practice, cloud testing is most effective when combined with internal load testing in a two-stage process that makes use of multiple cloud providers.

In selecting a cloud testing solution, keep in mind that the primary factor in your success will not be simply the move to the cloud itself, but rather the tool you use and how well it uses cloud technology.

About Neotys

Since 2005, Neotys has helped more than 1,000 customers in more than 60 countries enhance the reliability, performance, and quality of their web and mobile applications.

NeoLoad is a best in-class load testing solution from Neotys that enables teams to efficiently apply best practices for load testing with the cloud. It is integrated with multiple cloud platforms, and supports realistic, large-scale tests across multiple geographical zones with bandwidth simulation and parallelized requests. It complements cloud testing with full support for internal lab-based testing, and enables engineers to reuse test scripts across these domains. NeoLoad is a full-featured, secure, and easy-to-use solution that accelerates load testing with advanced real-time analysis, agentless monitoring, scheduling, scripting, and reporting capabilities. It also provides extensive support for a wide range of web technologies including Adobe Flex and AIR, Silverlight, RTMP, SAP, and Siebel among others. NeoLoad enables testing organizations to make the most of what the cloud makes possible, and helps them use the cloud to make substantive improvements in load testing.

Neotys solutions are backed by a dedicated team of Neotys professional services and support engineers to ensure your success.

For more information about Neotys and NeoLoad visit: <http://www.neotys.com>.

Neotys and NeoLoad are registered trademarks of Neotys SAS in the USA and others countries. All other trademarks are the property of their respective owners.

Copyright © 2011 Neotys. All rights reserved. No reproduction, in whole or in part, without written permission.

"Load Testing with the Cloud" – novembre 2011 – version 1.0.

NEOTYS USA, Inc.

241 Boston Post Road West
Marlborough, MA 01752, USA
Tel: +1 646 502 8061

NEOTYS (HQ)

30 avenue du Château de Jouques
13420 Gemenos, France
Tel: +33 442 180 830

NEOTYS Germany (Sales office)

Waldstrasse 50a
Neu-Isenburg, 63263, Germany
Tel: +49 (61) 02 43 27 38

<http://www.neotys.com>

E-mail: sales@neotys.com