# Why Performance Test Outside the Firewall?
Exposing What You Have Missed

Introduction

The testing of web systems for performance is becoming a somewhat regular exercise for many enterprises. This is great progress from the miniscule amount of performance testing that was done just two years ago. With that additional testing has come increased stability and scalability. But, there is a fundamental piece of most tests that is missing: the link to the end-customer.

Issues such as: slow page downloads, lost connections, failed downloads, 'strange' error messages, browser timeouts, missing images, failed transactions, and others can all be caused by network infrastructure that exists between your application and the end user. It is critical to test these.

For web applications that are not exposed to the outside world, this may not be as large an issue. However, for those applications that are exposed, either to end consumers, business partners, other physical locations of your firm, or other customers, this missing piece of test infrastructure is critical. Why? Well, if you consider that about 20% of all web application performance problems are due to network issues[1] the importance starts to become clearer. So let's continue on with a discussion of the types of 'performance' testing and how these relate to this issue.

Where Web Performance Problems Reside

Network 20%
Web Server 10%
Application Server 40%
Database Server 30%
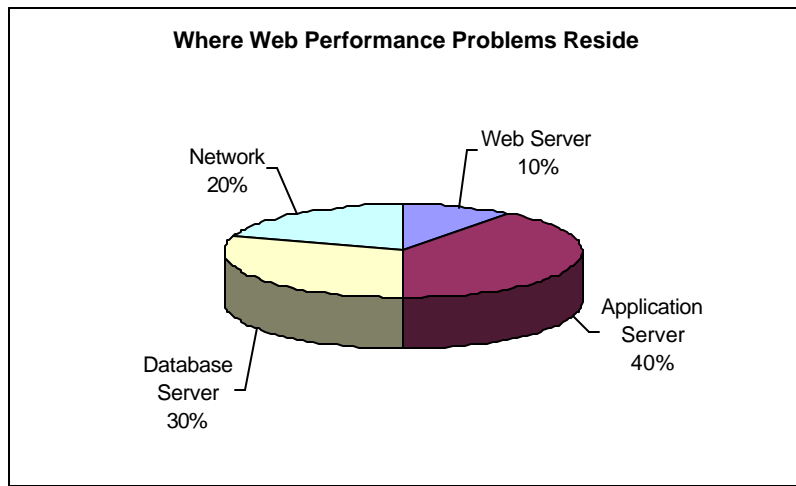
The Different Types of Performance Testing

Performance testing is an often used term used to describe many different types of tests that are all related to performance. Each of these styles of testing will find and locate different issues within the system under test. The types of testing listed here that are most

[1] Source: Empirix e-LoadExpert testing service and Empirix Professional Services, using data collected over 1000's of engagements.

often used while performing tests from outside the firewall are stress testing, load testing, and performance testing. Let's take a look at how these are defined:

Performance Testing
Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify that the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "workload" on the system. The initial test should be performed with a benchmark workload (best case workload) and a nominal workload that emulates the normal workload experienced (or anticipated) on the target system.

Load Testing
Load testing subjects the system-under-test to varying workloads to evaluate the systems ability to continue to function properly under these different workloads. The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload. Additionally, load testing evaluates the performance characteristics (response times, transaction rates, and other time sensitive issues).

Stress Testing
Stress testing is intended to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the software that aren't apparent under normal conditions. Other defects might result from competition for shared resource like database locks or network bandwidth. Stress testing identifies the peak load the system can handle.

Where In the Infrastructure Performance Testing Occurs
As discussed earlier, most testing in a corporate environment is performed on a system that mimics, as closely as possible, the production environment. This is fine, except that there is a key area of the system that is being ignored; the connection to the outside world. What many people do not realize is that this 'weak link' can have a dramatic effect on the user perceived Quality of Experience (QoE). Your servers and database may be lightening fast, but if your users perceive it as slow, that is what matters. And slowness can be caused by a number of factors.
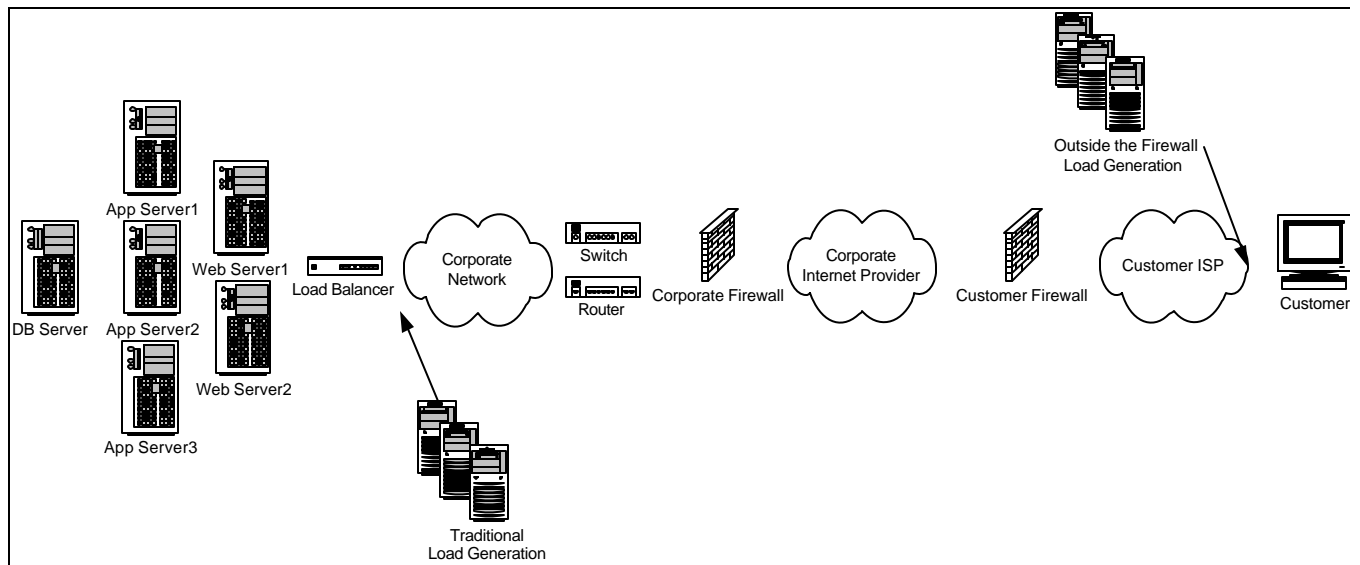
Take a look at the above system diagram. As you can see, internet systems are combinations of many pieces of infrastructure. Each of these pieces can and will have an impact on the performance that your end-user perceives. If there is a piece missing during your testing, you cannot rule out that this area of your application will not be the cause of some future problem.

So Why Does Testing Outside the Firewall Matter?

Let's say, for example, that you are designing a performance test. As part of that test, you lobby for a test environment to be built that will nearly mimic the production environment. You begin testing, as shown in the above diagram, using the method of "traditional load generation." That is, you are instilling load into the system at a point right in front of the servers to be tested. This is a good thing. And testing should *start* in this manner. But this should not be the end here. As your testing progresses, you will perform many iterations, with each iteration fixing a problem and retesting.

The next step is to move further back in the system infrastructure. In this case, we are going to inject load at the point furthest away as possible from our servers. In the example above, this is at the customer ISP. The goal here is to verify that the pieces between the customer and the servers are functioning correctly.

It is at this stage were we see about 20% of system failures occur.

Throughput vs. Concurrency – Why This is Critical

The other critical issue that comes up when it comes to testing the networks and infrastructure between the end user and the servers is throughput and concurrency. In fact, these two issues make up most of the problems that we see from this part of the infrastructure. First, some definitions:

> Throughput: The measure of the flow of data, e.g.hits/second, pages/second, Mbps (megabits/second)

Concurrency: The measure of independent users that a system can support. On the system level, concurrency is limited by sessions, and socket connections. On the application level, flaws in the algorithm can limit concurrency.

So what types of problems can occur as a result of throughput problems? This area of system performance is the easier of the two to track down. In general, if you are running a performance test, and you see that your user response time begins to slow down, but you also notice that all key indicators on your servers look fine, this is usually a throughput problem. Another way to test this is to place large files of known size on the web server and then validate with a tool the speed at which these can be downloaded.

As an example, if your system is designed to sustain 10mb/sec of traffic. However, you find that your link to the internet is only a 1.54mb/sec line, you will never reach that peak, because of a lack of available bandwidth. Only testing from outside the firewall is going to tell you this.

The second piece is concurrency. Concurrency errors happen when a system cannot handle the number of connections it needs to maintain adequate performance. Indicators of concurrency problems tend to be a bit more difficult to track down, but key causes tend to be incorrectly configured web servers, firewalls, and load balancers.

So What to Do?
Test from both inside and outside the firewall. For your systems that have uses that access through the internet, outside a corporate WAN, or in another physical location on the corporate WAN, you must understand the effect of the network on your web system.

Doing this may be as simple as running performance tests from another physical location. Or, setting up servers at an ISP outside of your ISP. If you don't have these available to you, there are other options, such as, engaging an outside firm to assist you in performing these tests. Whatever you do, make sure you understand the risks of NOT testing outside the firewall and understand the benefits of performing these tests. In our experience, the benefits far outweigh the risks.

How Empirix e-Load Expert Can Help
e-LoadExpert (**http://www.empirix.com**), offered by Empirix, is a managed, hosted load testing service that provides the fastest and most accurate way to test the scalability of your Web applications. The key difference here is that we will assist

Some of the most successful customers Empirix has have performed their own testing inside their firewall, and then engaged us to assist with testing outside the firewall. And, the cost to do this is probably less that you might think. The engagements are fast, and full of useful information. In addition, you will work directly with a dedicated consultant who will help apply their knowledge of testing to your application.

Where Web Performance Problems Reside

Web Server
10%

Network
20%

Application
Server
40%

Database
Server
30%