Why Load Emulation in the QA Lab Isn't Enough: Ensuring application performance to remote end-users

A Shunra Software Technical Note

Spring 2004

Shunra Software Ltd - US Headquarters
555 Eighth Avenue, Suite1102
New York, NY 10018, USA
Tel: 212 279 8895; Fax: 212 279 9561
info@shunra.com; www.shunra.com

Introduction

Rigorous performance testing and optimization is a critical factor in the successful delivery of any business application. Yet frequently the performance of deployed applications doesn't live up to business requirements or end-user expectations. One reason behind these unpleasant "surprises" is the fact that most performance staging labs only test the application with local users (in a local area network (LAN) environment), while the fully deployed application is used by a variety of users, some local and others accessing the application remotely over different network links. The different network conditions that exist between end-users and application servers have a tremendous effect on the overall performance that remote end-users experience. This deviation from performance test results obtained in the lab is further exacerbated for N-Tier applications where each tier may reside in a different geographical location with it's own unique set of network conditions.

Preventing Future "Surprises" For Remote End-Users

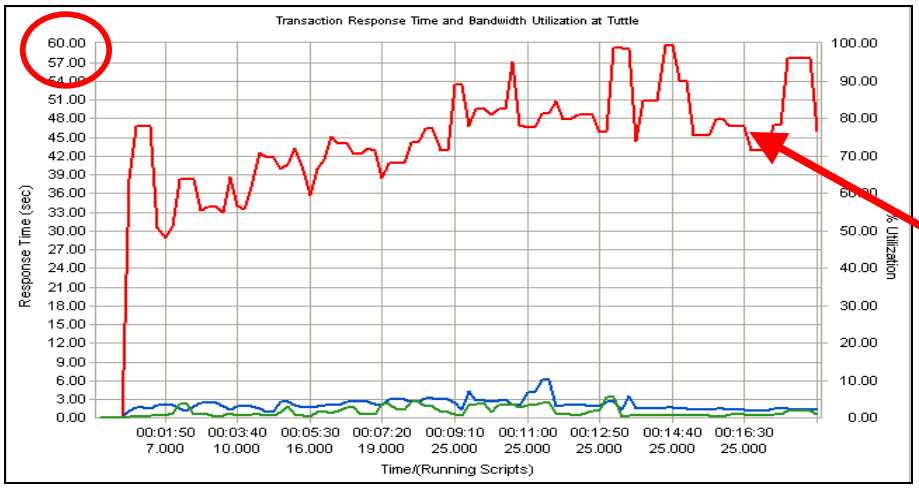
A simple example of such a deviation can be seen from the results of the following test conducted at a major North American financial institution. The application chosen for performance testing was assigned an initial service level objective (SLO) of no greater than an average 16-second response time. Instead of testing the application with only local users, the exercise also included users from remote offices in Florida and Texas.

The results (Figure 1) show a substantial deviation from the performance of the local users on the local area network to that experienced at the remote locations on the production network (also know as the Wide Area Network (WAN)). While increasing the local users load did not seem to strain the application server or backbone network, the same gradual load increase from the remote locations negatively affected performance due to overwhelming strain on the network and the application servers as well. Furthermore, if testing were only conducted on the local users, it would appear that the initial SLO would be met, but by adding the remote users it is clear that this is not the case.

Location	Max Throughput (Kbps) & Bandwidth Utilization (%)		Scripts		
	In	Out	Name	Max Run	Transaction Response Time(sec.)
Local Users in OH with local Data Center (LAN)	3838.29 (0%)	658.84 (0%)	Financial Transaction	25	15.00
Remote Office Users in FL	1555.45 (100%)	473.58 (33%)	Financial Transaction	25	16.98
Remote Office Users in TX	1168.66 (100%)	656.86 (80%)	Financial Transaction	25	24.48

Figure 1. Summarized Results of Financial Application Performance Testing.

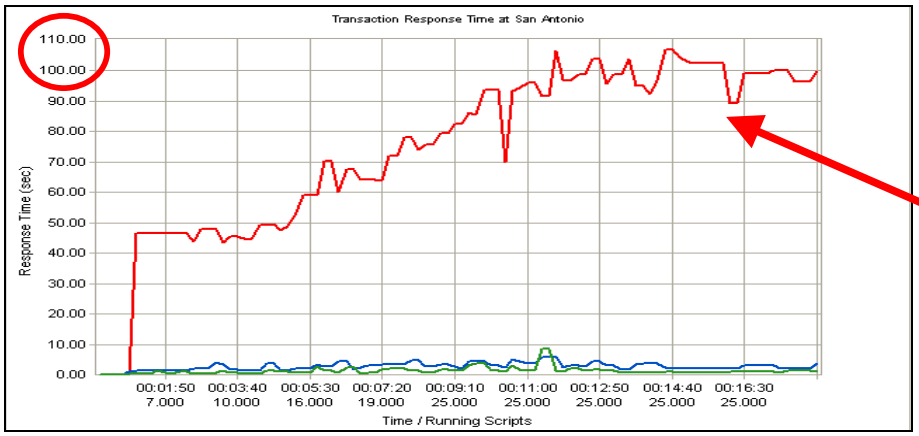
--Tested over a local Area network --



As the end-user load increases, the performance impact is not severe. Average transaction response time increases by only 25%.

*Figure 2 - Performance Testing Results from Local Users on the **local area network.***

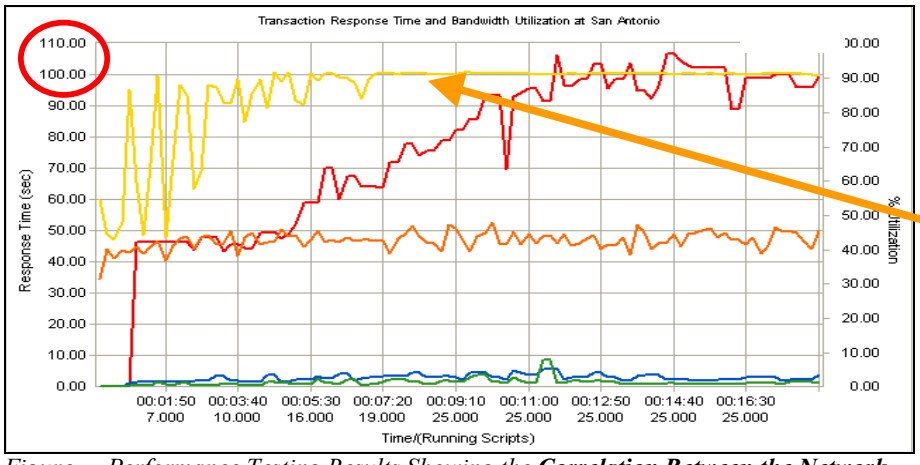
--Tested over an emulated production network --



As the end-user load increases, end-user performance is severely impacted. Average transaction response time increases by 120%.

*Figure 3 - Performance Testing Results from Remote End-Users on the **production network.***

--Tested over an emulated production network --



As the end-user load increases, the network load increases to the point of saturation, negatively affecting the remote end-users application performance.

*Figure -. Performance Testing Results Showing the **Correlation Between the Network and End-User Load and Performance.***

It's Not Just the Network

The previous example clearly showed that existing network conditions can significantly impact the performance of distributed applications, but it doesn't tell the whole story. In addition to the delay from the constrained network resources, the stress on the servers also increases since more resources are needed to support remote users. Sessions are open longer, OS resources are occupied for more time and more concurrent threads are needed. None of these factors are taken into account when testing and optimizing the performance only for local users on the local area network. Thus, remote business users are often unpleasantly "surprised" by poorly performing applications.

One of the primary causes of application performance shortfalls is insufficient hardware capacity. Any capacity planning that is done without considering the remote users is lacking vital information and may result in misleading or incorrect recommendations by ignoring the very real network and server conditions that exist on every corporate network.

I Don't Control the Network – Why Should I Care?

Although the network is a different beast from those that QA professionals typically live with, it is still very important to consider it during application testing. The overall objective is to verify the performance of the application as experienced by the end-user (or specific subgroups of end-users), especially if that end-user is accessing the application remotely. Although the network appears to be someone else's responsibility, (Ah, so that's what those people in the network group do), no one can guarantee that the corporate network will provide acceptable conditions for optimal application performance for all end-user groups. If an application can't handle the conditions on the corporate network, there is very little anyone can do on the network side to fix it. It is really the responsibility of the entire development team, and QA in particular, to verify that an application can cope with the harsh conditions often found on the corporate network. Many solutions that improve the application's networkability rely upon 'tweaks' to the application itself rather than changes to the network. This is true regardless of whether it is in the way the client retrieves information (e.g. entry by entry Vs. retrieving bulks of information), the way the application is broken into different tiers (i.e. keeping the meta data closer to the user, Vs. keeping all tiers in one geographical location) or many other best practices that dramatically improve the delivery and performance of an application to remote end-users.

Delivering an application without considering the target network is just as bad as not testing the application under load. Letting remote end-users uncover the performance implications of not testing against network conditions is never a good idea.

Can I Use My Load Tool's "Think Time" and Bandwidth Option to Create the Network Effects?

Adding "think time" to the end-user scripts, or using the bandwidth option of a load tool basically cause the virtual user's transactions to operate at a slower pace as they advance through the script. Simply adding a fixed transaction delay or artificially lowering bandwidth does not accurately represent real world conditions where remote end-user transactions cause contention for infrastructure resources, introduce bottlenecks along the connectivity path and

experience delay due to the geographic distance and the load on the network. The "think time" option assumes a linear relationship between the effects of being remote and the response time of the application but the real world is much more chaotic and complex than that. The bandwidth option assumes that all end-users (or specific groups of end-users) will be affected equally when, in fact, this is rarely the case under real network conditions. While both "think time" and bandwidth options can mimic some very limited network conditions, they lack the sophistication to adequately reflect real world conditions and are, therefore, of little use when testing remote user application delivery and performance.

As previously mentioned, yet another factor that needs to be taken into account is the differential way the application servers scale under remote end-user load. When the end-users are remote, servers need to maintain each session for a longer period of time, thus, utilizing operating system resources (sockets, threads, processes) for prolonged periods. Any QA test that doesn't take all of these factors into account will generate skewed results, similar to the deviation found between local users and remote users in the previous example.

Taking Remote Users into Account without Leaving the Local Lab

Testing an application for delivery to remote end-users over a production network is a daunting task without the proper approach and solution. All tests referred to in this technical note were conducted in a single lab while only using resources available on the local area network. The effects of the remote end-users were achieved by using Shunra\Storm with its StormTraffic add-on option. Shunra\Storm is a powerful and flexible distributed application performance solution suite that creates a replica of any enterprise network environment including the production network conditions and end-user load. QA professionals, performance and networking managers can then deploy their application over this emulated environment to test the functionality, performance and scalability of any distributed N-Tier application while getting to feel, first hand, the real-life experiences of remote end-users as they interact with the application. Shunra\Storm is ideal for any distributed application such as ebusiness applications for global traders, remote agents and mobile salespeople, ERP, CRM, and SCM applications, convergence initiatives such as VoIP, as well as .NET and Web Services based applications. More information on Shunra\Storm is available at: http://www.shunra.com/products/storm/storm_1.php

About Shunra Software

Shunra Software offers a complete set of solutions that enable software development teams, testing professionals, and performance and networking engineers to design, develop, test and tune their applications for the production environment *right from the start.*

Shunra's products can emulate any IT environment external to the data center, from a single dial-up user to a global enterprise network. IT professionals can then run and observe their application in this replicated environment to accurately test its functionality, scalability and performance. This allows them to pinpoint potential problems in application design early in the development process – while those problems can still be solved with relative ease and minimal expense.

Shunra's products enable companies to ensure a successful, on-time, on-budget global application rollout and:

- Build quality and optimal performance into the application, right from the start
- Reduce downtime and shorten troubleshooting and debugging cycles
- Plan infrastructure resources in advance, and avoid costly or unnecessary post-production upgrades
- Minimize reactive 'firefighting' to more fully concentrate on strategic direction
- Boost productivity of staff, end-users, and customers
- Ensure technology achieves business goals

Shunra's products are used at over 1,200 leading organizations worldwide, including 3M, Alcatel, Boeing, Cisco, Dow Chemical, EMC, FedEx, General Electric, General Motors, JP Morgan, Kelly Services, Merrill Lynch, Motorola, Nortel, Pitney Bowes, and Sun Microsystems. Shunra also partners with some of the world's leading technology vendors, including Accenture, IBM, Oracle, Mercury Interactive and Microsoft who use our products internally as well as to ensure the success of their customer deployments.

Shunra is a privately held company based in New York City, with sales offices throughout North America. Shunra's EMEA operations are based in Kfar Saba, Israel. Shunra is supported by a global network of channel partners worldwide. For more information, please visit www.shunra.com.