

Newport Group, Inc.

IT Trends Research and Reporting

ENTERPRISE LOAD TESTING FOR WEB APPLICATIONS

*Developing an Effective Strategy for Optimizing Application
and Infrastructure Performance*

INTRODUCTION

Web-based systems that perform poorly, function incorrectly or fail completely cost businesses money, reputation and market share. Yet, instances of downtime, unacceptable performance and incorrect application behavior incidents are prevalent industry-wide. For example, application problems with the State of Minnesota's \$27 million online tax filing system delayed tax refunds and diminished consumer confidence for greater than 60,000 tax filers this year. Prior to the 2002 Winter Olympics, an online ticketing page was unavailable during a critical sales period placing a potential of \$1.5 million dollars of revenue at risk. BestBuy.com and BlueLight.com watched their sites stall and/or significantly slow down following the launch of Playstation2 in December of 2001, turning countless customers away at their virtual doors.

Yet, given the collective impact and growing number of examples that demonstrate the rising costs associated with failures, performance problems persist and worsen according to Newport Group research. In a 1999 study on performance and scalability, businesses reported that their web applications could handle only 72% of the concurrent traffic volume originally planned. When the study was repeated in late 2000/early 2001, businesses reported that their web applications could handle only 66% of the concurrent traffic volumes originally planned. These trends indicate that the majority of today's web applications are not meeting expectations for performance, despite the fact that the revenue derived from these web-based systems continues to increase significantly.

To combat the growing problem of poor web application performance and safeguard the rising amount of business revenue gained via online channels, load testing strategies, tools and services have experienced a transformation in terms of both awareness and adoption. The ongoing use of load testing practices throughout the application lifecycle serves to protect the performance and integrity of revenue-generating applications and infrastructures and has become widely recognized as a critical IT function that is strategic to a solid performance achievement and maintenance strategy.

This white paper examines best practices for enterprise load testing in today's web-enabled environments, the appropriate use of automated tools and/or services, the benefits to adopting a lifecycle approach to load testing, and in general, serves as a guide to understanding the ingredients necessary to build and execute meaningful load tests. The conclusions outlined in this paper are the result of extensive industry research among Global 2000 companies and a broad range of direct customer experience working with IT departments who have implemented various approaches to load testing. Lastly, this white paper will also profile the load testing services offered by Keynote Systems.



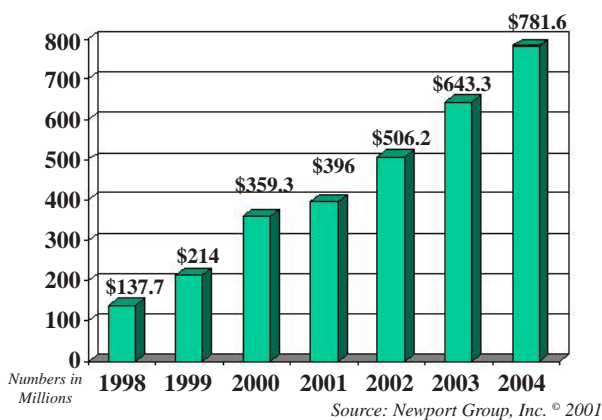
THE BUSINESS-CRITICAL DISCIPLINE OF LOAD TESTING WEB APPLICATIONS:

Building a Business Case

Web-based technologies are enabling the future of global commerce. As they work to enable new business opportunities with lower transaction costs and higher profitability potential, web-based technologies are being absorbed into the fabric of the world's largest and most successful businesses. Recent Newport Group research reveals that on average today's Global 2000 business attributes 22 percent of bottom line revenue to their web-enabled business channel and that number is expected to grow to 27 percent by the end of 2002.

Business requirements for web application load testing as a means for ongoing performance measurement and tuning have become more rigorous over the past several years. This is evident based on the fact that the worldwide market for load testing tools and hosted services reached \$359.3 million in 2000, up 68 percent from its 1999 level of \$214 million. Of significant contribution to the fast-paced growth was the requirement for load testing web applications, a sub-segment of the market that grew by 114%. In consideration of strained economic times and the maturing of the market, Newport Group expects that the market reached nearly \$400 million in 2001 and will climb to slightly more than \$500 million in 2002.

**Worldwide Load Testing Market Growth
Distributed Environments, 1998 - 2004**



Newport Group defines load testing as the process of analyzing software applications and supporting infrastructures to determine acceptable performance, scalability, capacity and transaction handling capabilities by modeling real-world application usage conditions and executing them against the application and supporting infrastructure under test. Measurements de-

rived with load testing tools should provide a clear understanding of where performance bottlenecks reside and aid in infrastructure and capacity planning of computing resources. When derived from meaningful load tests, results serve as a guide to helping IT staff make informed decisions about the performance of their applications and infrastructures. Accurate and realistic load tests are the best indicator of how well any business system is able to service its customers in real-world scenarios.

However, a key point for businesses to recognize is that load testing web applications doesn't guarantee success. Businesses that approach load testing haphazardly, without a plan and/or in reaction to a major business catastrophe, leveraging the practice as a one time event, rather than an ongoing process of performance measurement and improvement, will not achieve the success possible with this discipline. Load testing is an ongoing process that must take place on a regularly scheduled basis, in sync with the ongoing changes that happen throughout the lifecycle of an application and its supporting infrastructure. A solid load testing strategy must complement performance monitoring and analysis in a production environment and, in turn, production monitoring and analysis should be leveraged to improve the accuracy of load tests.

MAXIMIZING ROI FOR LOAD TESTING

The achievement and maintenance of optimal application and infrastructure performance is a complex task. In addition to selecting the appropriate technology solution in a crowded market, IT managers must contend with proving the value of these solutions to business management. The cost to a business for poor application and infrastructure performance is only evident when there is a problem. Thus, the overhead of tools, staff and/or services necessary to facilitate an ongoing performance management strategy becomes much like an insurance policy. Individual businesses need to define their own risk tolerance and design a strategy that mitigates that risk accordingly.

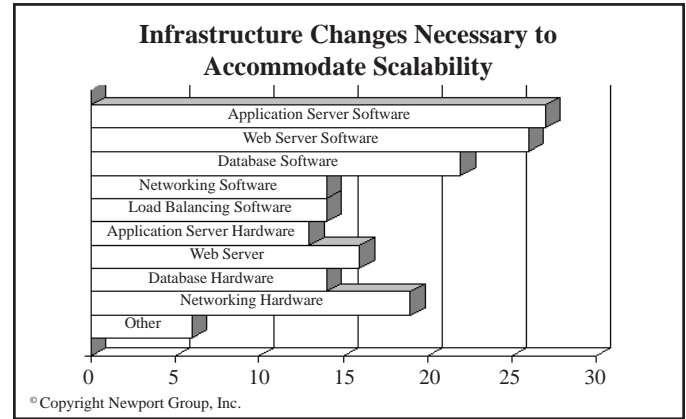
Any business leveraging technology-driven systems to gain revenue must implement a proactive strategy to manage the performance of their systems. However, the real challenge for a business lies in understanding their individual level of risk and then determining how many budget dollars they are willing to contribute to mitigate that risk. Poor performance equates to turning customers away at the door and negatively impacts revenue potential. Therefore, the cost/benefit argument must present the potential real-world impact in terms of the cost of slow time/downtime per hour, percentage of lost customers, the damage to business reputation, the cost of brand degradation, etc.



Newport Group research asserts that the cost of downtime for every million dollars of daily business revenue generated online can range between \$9,500 and \$27,500 per hour. However, this hourly downtime cost range is an average of nearly 250 businesses. Cost of downtime for large companies can exceed that significantly. For example, Intel reports that it earns about \$275,000 per hour from its online business channel. A company such as Delta Airlines reportedly earned approximately \$3.8 million per day in online ticket sales. All hours being considered equal, Delta has the potential to lose upwards of \$160,000 per hour of downtime. These costs are straightforward downtime costs that are measured based on average daily revenue intake via a web-based channel. They do not include subjective, intangible costs such as damaged business reputation, brand degradation and percentage of lost customers to competitors.

There has been a strong focus on return on investment (ROI) over the past 12 months as a result of current economic conditions and shrinking corporate profits. Businesses are finding that they must do more with the resources currently available to them as a result of receding IT budgets and staff layoffs. In 2001, the IT industry realized its highest unemployment rate ever, peaking at 5.5% in November of 2001 and climbing to 6.0% in January of 2002. Additionally, Merrill Lynch reports that IT budgets contracted by 2.4% in 2001 and the most recent data from Information Week's quarterly "Priorities" study concluded that an average of 7.7% of annual revenue was set aside for IT expenditures, down from 8% last quarter.

Under these circumstances, performance management solutions have a strong chance at securing budget dollars over new infrastructure purchases. Several years ago, many business climates supported the purchase of extra infrastructure in order to accommodate performance. However, today, these additional infrastructure purchases must be justified to upper level management and a business case must be proven. Adding hardware or software licenses is not the most efficient way to build in better performance and conduct capacity planning. It is a band-aid approach that does not necessarily address the actual cause of the performance problem. Newport Group research shows the most common reasons for performance problems to be software related – **meaning that better performance is most often gained through application and infrastructure tuning techniques.** Of course there will be times when additional hardware or software is required. Use of a load testing tool or service will aid in accurately predicting when capital expenditures are really necessary and provide clear results that justify the additional investments to upper level management and key decision makers within the enterprise.



By taking the application and infrastructure tuning approach, businesses achieve greater performance from existing systems. They gain critical information about the performance of their systems, which aids in more efficient problem resolution, and they are able to save money by accurately assessing and meeting capacity requirements. Also important to note is that adding more hardware to improve performance and gain redundancy is not a failsafe policy because the complex nature of Internet traffic has the potential to rely more heavily on specific resources. Thus, without the proper testing, tuning, and performance management practices, additional infrastructure that is not optimized appropriately has the potential to be underutilized. The bottom line is that relying on web application load testing as part of a solid performance management strategy will serve to protect revenue-generating business systems and control business risk.

Once the level of business risk is determined and a budget in support of the load testing effort is established, consideration must be given to finding a balance between identifying the right load testing tool and/or service for the enterprise and maximizing the business ROI for that choice.

IDENTIFYING THE RIGHT LOAD TESTING TOOL OR SERVICE FOR THE ENTERPRISE

The most widely adopted approaches to load testing include the purchase of load testing tools for in-house use and the newer, vendor-hosted load testing service approach. **Both approaches will demonstrate a business ROI, however, given the substantial up-front costs involved with the purchase of tools for in-house use, the outsourced services-based approach will provide a shorter-term ROI.** Since their introduction in early 2000, load testing services have become increasingly attractive for enterprises that are faced with maximizing their existing IT resources, reducing costs and achieving ROI in the short-term. In choosing either a tool and/or an outsourced load testing



service, beware that not all are alike. It is beneficial to select tools and/or services that are intuitive enough to test the dynamic and variable nature of web-based environments.

Packaged Testing Software: Typically, load testing tools sold as software packages require purchasers of the software to supply the hardware and staff necessary to run the load test software and drive tests. This approach involves up-front costs for procurement of the testing software and necessary hardware to conduct the tests, and IT resource costs to implement, develop, execute and manage the load testing process. Additional costs include annual software license fees for tool upgrades and vendor support, potential consulting and training costs and, often, the time and cost necessary to set-up a testing lab to mirror the production environment. This approach is costly, takes the longest time for a business to achieve its return on investment in comparison to other approaches, but provides complete in-house control and flexibility over load testing activities.

Vendor Hosted Services: Early in 2000, vendor hosted load testing services emerged to provide an alternative for conducting load tests for web application environments. Load testing services leverage the Internet to conduct load tests over the wire. With this model, load tests can be driven from one to many geographic locations around the world. The pricing for load testing services is inclusive of the utilization of the provider's hardware resources for test execution. In this respect, customers pay to have load tests designed and executed against their web application infrastructure and the service vendor leverages its own technology and network of distributed hardware resources to execute tests and deliver results. This approach provides a quick return on investment and is attractive for conducting tests with high volumes of concurrent users. Traditionally, services have been thought of as less flexible than in-house tools, however, there are a variety of different services vendors whose unique solutions can provide an equivalent level of control and flexibility.

So, when is it appropriate to use a load testing service versus an in-house tool? The answer entirely depends on the testing requirements of an individual business and the amount of resources available in terms of staff, expertise and hardware. The service-based approach adds value for those new to load testing because it can quickly build a test foundation and return test results quickly. Load testing services, such as Keynote LoadPro™ and TestPerspective™ (see sidebar, page 12) are attractive to businesses looking to maximize ROI during tough economic times, are overburdened with testing tasks and require additional support or simply desire the objectivity and the expertise of a service provider.

Perhaps the greatest value in leveraging load testing services lies in the expertise of the provider as well as the capabilities of providers to execute and drive tests from outside the firewall. Designing, executing and interpreting load testing results for widely distributed web applications is complex because environments are dynamic and impacted by constant change. There is a great deal of planning and analysis that must go into creating and executing realistic load tests for web applications and infrastructures. Load testing services are conducted by teams of experts that conduct a wide variety of load tests on a daily basis. From the expertise they gather through engagements with multiple customers, providers gain unique insight into common performance problems and can be highly resourceful in solving common problems quickly. Further, a good service provider will have valuable experiences that enable them to quickly emulate realistic user audiences for various vertical industries, such as retail, financial, telecommunications, etc. Keynote System, for instance, offers a testing service uniquely tailored to the retail vertical.

A recent Newport Group study found that 51% of businesses prefer to use hosted load testing services as a complement to in-house testing activities for select launches and/or upgrades - adopted on a more sporadic, as needed basis. The marriage of the two approaches just discussed is gaining popularity and offers compelling business value. In general, Newport Group recommends a mix of in-house tools and remote testing services. If possible, combining in-house testing efforts throughout the development lifecycle with hosted service expertise will provide the most balanced approach to load testing.

OPERATING IN A WEB ENVIRONMENT:

*The Challenge of Constant Change
Drives the Need for Load Test Execution*

Businesses intent on deriving revenue and efficiently servicing customers via a web-based business channel must be continuously aware of how well they are adjusting to rapidly changing conditions and unknown variables. Applications that were once relied upon to support a known number of users, accessing systems with known technologies, growing at known intervals and internal to a particular business, are now expected to support exponentially higher volumes of unpredictable users accessing systems with varying technologies and network structures, growing at relatively unknown intervals and external to the business.

Consider that industry estimates reveal that the number of worldwide Internet users was just under 200 million in 2000, and that number is expected to more than double by 2003. Even at 400 million, the number of Internet users will only represent 5.7% of the world's total population, leaving room for significant population expansion of Internet users. As the population of the user audience expands, that audience gains experience and familiarity with the convenience of transacting business online and will ultimately become less tolerant of poor performance. At the same time that audiences of Internet users expand, web applications continue to grow in size, complexity, features, and scope of integration with other applications and systems, both internal and external to a business. Simultaneously, new technologies continue to be introduced, adding more complexity to already complex environments.

Web application environments must be open and flexible enough to support the multitude of technology combinations relied upon by their heterogeneous mix of users and load test solutions must be robust enough to handle accurate testing of these complex environments. Configuration changes to an application infrastructure, the addition of clusters of servers or any new infrastructure devices, the partnership with new service providers, the integration with new applications or the requirement for a newly released application to share the computing resources of an infrastructure are several very strong drivers for the execution of load tests. The addition of application features and functions or changes to existing features and functions are also strong drivers for the execution of load tests.

In addition businesses must stay in sync with business growth, changes in Internet traffic patterns and customer profiles, seasonal usage patterns, special promotions and marketing campaigns. All of these various types of changes drive the need for load testing on a regular basis. In an effort to ensure that the profile of a user audience isn't negatively impacting performance, schedule a monthly load test that realistically emulates the current customer profile (use web logs to extract appropriate information). For example, maybe the number of site visitors that are becoming customers is increasing, and if so, how is the increased number of online purchase transactions affecting site performance? Is the application able to deliver responses within the realm of business expectations? In another scenario, maybe the business marketing manager is planning a special campaign for a busy holiday season, like Christmas or Valentine's day. If that promotion is going to reach millions of potential buyers, is the site prepared to handle volumes of traffic that may spike and exceed daily traffic volumes by a factor of two or higher?

All of these evolving variables have the potential to drive heavier volumes of customers to a particular business system at any given time. Thus, applications and their supporting infrastructures must be able to handle steady volumes of sustained traffic as it relates to known business patterns as well as unanticipated spikes of user traffic. This is an important point to keep in mind as load tests are developed because not only should a system be able to handle normal, anticipated traffic patterns, but it must also be able to survive unanticipated spikes of traffic and potential denial of service attacks.

THE LIFECYCLE APPROACH TO LOAD TESTING

To derive the greatest value and business ROI from the incorporation of web application load testing as part of the overall performance achievement and maintenance strategy, research indicates that IT teams must develop a lifecycle approach to testing throughout the application development phases. The load testing strategy with the greatest track record of success supports testing at critical stages of the application lifecycle, including design and development, staging and quality assurance, production and post-production stages.

During the design and development stages, system engineers and architects must work to gain an understanding of the infrastructure as a coordinated system. For example, how is performance affected when clusters of web servers or load balancers are added? As it relates to the design of the application itself, design engineers and developers should consider conducting walkthroughs to identify where bottlenecks could potentially reside. Resolving issues at this stage is much less costly than if design issues are uncovered prior to an application being pushed into production.

During development, simple load tests can be conducted on individual components of an application to understand if those components scale individually. Typically developers will test their own code components for functional correctness and to be sure they are in-line with application design specifications. Testing at the integration stage (staging and quality assurance) will tend to go smoother and faster when coding fixes can be eliminated as the cause of problems.

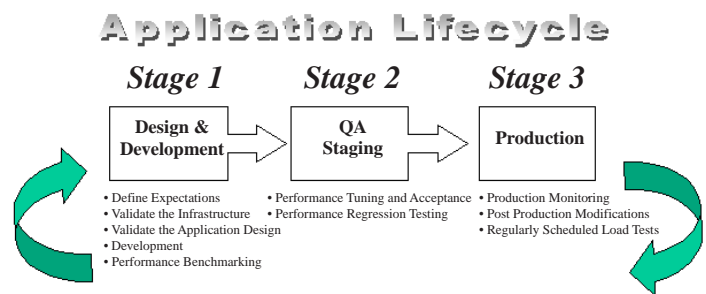


The most rigorous load tests are conducted at the staging and quality assurance stage of the lifecycle. QA staff must design meaningful load tests that simulate realistic web application users. QA staff will want to use the results of load tests to understand the capacity of the system and the number of users the site can expect to handle. As tests are run and results are viewed in real-time, QA and operations staff should work together to understand how various scenarios would impact performance. For example, how will the system operate if one and/or several servers go down under varying loads? As the database is populated with more and more registered users, is performance negatively impacted? Has my ISP allocated an appropriate amount of bandwidth to support anticipated user loads?

Once an application has survived rigorous testing in the staging and QA stage of the lifecycle and the application is deployed to production and post-production stages, load tests will remain a critical IT function. During the production stage, QA staff and operations staff will use load tests to combat ongoing changes that can impact a web application and supporting infrastructure. Additionally, special promotions and marketing campaigns will drive the need to load test as will the continuous evolution of a web site's user audience.

A load testing strategy for today's web-enabled application must consider performance factors that are both inside and outside of the firewall. Businesses will want to ensure that all of the infrastructure components within their own firewall – such as web, application and database servers, load balancers and internal network components – are fully optimized to function as a coordinated system. Once the application and infrastructure is opened to the public Internet domain, they will also want to test the performance of their coordinated system with variables outside of the firewall, such as the firewall itself, ISP bandwidth, routers, switches, external network components and peering relationships. Load tests inclusive of these external variables provide a holistic and realistic understanding of application performance, as real web users will experience it. Also, although these variables are outside of the control of the enterprise, they can and often are responsible for a fair amount of performance-robbing problems. For example, additional bandwidth can be added via an ISP or faulty routers or peering relationships can be identified and brought to the attention of the appropriate party for improvement. Using load tests will provide the factual information necessary to prove a case to a third party provider, eliminating finger-pointing and giving the performance issue a greater chance at being resolved correctly.

As it pertains to strengthening the lifecycle approach to load testing, businesses will best be served by involving the entire IT team, or at least representatives from multiple IT disciplines (dependent on IT team/company size). The involvement of a wide array of IT team members, representing various disciplines will typically be a significant organizational change for most businesses because within most IT departments, team members tend to work in a vacuum, focusing only on their specific discipline (i.e., the database, the application, the network, etc.). One suggestion for businesses working to facilitate a testing effort that encompasses greater collaboration among IT team members is to centralize the management of the testing process. For example, involve developers, test engineers, network specialists, database engineers, system administrators, design architects, business analysts and webmasters. Load testing is a practice that spans pre- and post-production environments and the more involved and informed various members of an IT team are, the greater the success of the load test strategy. It is important to promote collaboration between pre- and post-deployment IT groups, as load testing results are valuable metrics to exchange and share.



The lifecycle of any application and infrastructure is cyclical and load testing at each stage will evolve and be repeated as both business and technology changes are encountered. In general, consider how performance can be measured at each of the following stages.

**Lifecycle Stage 1:
Design & Development of a Web
Application (Pre-production)**

(Key Participants: CTO, design engineers, business analysts, developers, QA engineers, operations)

Step 1: Define Expectations: The first step to a load testing strategy should be to set goals for application performance. Before an application and its supporting infrastructure is designed and developed, IT teams must project the number of concurrent users, estimate growth pattern rates and consider application integration issues (both internally and via third party applications).

Step 2: Validate the Infrastructure: In architecting the infrastructure of a web-enabled business application, use load tests to validate that the software architecture will deliver the necessary performance and will scale linearly as hardware is added to accommodate growth. A thorough review of the appropriate configuration settings of various servers can ward off many performance-robbing problems early. For example, a check of CPU utilization on the database server will help correlate SQL updates or transactions requests relative to application performance. Metrics such as CPU, disk and memory usage of various servers (web, application, database, etc.) can serve as performance indicators. Web site architects should have a clear understanding about the configuration of the infrastructure. Decisions should be made about the number, types and layers of infrastructure components including web servers, load balancers and firewalls needed to support the application. In addition, information pertaining to third party infrastructure of an ISP and/or other third party service providers should be well understood at this early stage of the lifecycle to determine the performance impact on the supported applications.

Testing the scalability of the infrastructure must happen at the start of development, presumably after a prototype of the application has been created and is able to generate transactions to touch all tiers of the application infrastructure. This allows design engineers to determine the viability of the infrastructure as a coordinated system and correct design flaws that will result in bottlenecks, as well as address incompatibilities between technologies before the application is built.

Step 3: Validate the Application Design: Analyze the design of the application from a user's perspective. Is the application user-friendly and does it meet business requirements? For example, if a site is designed to sell a product, it is prudent to make sure that the application does not sidetrack potential customers with pages not relevant to the purchase. These basic checks will help create an efficient and appropriate application that meets the needs of its users. IT staff must validate whether or not the web application is meeting original design specifications and/or whether or not there are any unnecessary features and functions that add avoidable complexity. Try to uncover whether or not there are any code sequences that may degrade performance.

Step 4: Development: During development, simple load tests can be conducted on individual components of an application to understand if those components scale individually. Typically developers will test their own code components for functional correctness and validate their compliance with application design specifications. It is suggested that while testing for functional correctness,

they also test the same component with a small number of users (no more than 25, typically only 5 or 10) to do an early check on the scalability of the component.

Step 5: Performance Benchmarking: Carefully defining the types of load tests necessary and then executing the first round of tests against the initial version of an application provides a set of metrics, commonly referred to as performance benchmarks. Benchmarks can be established for individual components as well as for all of the integrated components once all code is stable. Benchmarks provide a point of reference against which future performance tests will be compared in order to quantify the capabilities of the application within its unique infrastructure and understand true performance improvements or degradations. Based on the changing performance requirements of individual businesses, the benchmark level will either be maintained or improved upon through subsequent milestones.

Lifecycle Stage 2: Staging and Quality Assurance of a Web Application (Pre-production)

(Key Participants: CTO, QA engineers, operations)

Step 1: Performance Tuning and Acceptance: This is the final load testing phase prior to the official application launch where all the different pieces of the application and its supporting infrastructure are integrated and performance is validated. Different transaction scenarios of real life usage are emulated and the performance capabilities of the final configuration are validated. Multiple varying scenarios are used to configure the hardware and software components to yield optimal performance. For example, testers will want to validate aspects such as the maintenance and resets of IP persistence after each log-off and that log-on response times are consistent with company performance goals or service level agreements. They will want to ensure server-clustering techniques provide adequate disbursement of incoming requests, fail over strategies and whether or not the application will handle the level of concurrent users anticipated as well as unanticipated user spikes. Newport Group research reveals that on average, Global 2000 businesses test with twice the amount of users they anticipate will be encountered in the live environment. We in turn would recommend this strategy because it has helped sites to cushion themselves against traffic spikes and it provides room for business growth.

Typically, testing the performance of an application running over the coordinated infrastructure components internal to a business (inside the firewall) is first done within a staging environment because test variables are easier to control. The staging environment is an exact replication of the production environment – including



all networking configurations, hardware, software, load balancers, etc., and it is our recommendation to test in a mirrored staging environment to have the greatest confidence in test results. Some businesses may opt to run tests on their production system and this strategy will work as long as businesses are equipped with a load testing tool or service that provides real-time results and gives the tester flexibility and control so that they can stop the test once they see certain statistics reaching a risky level.

Once tuning and acceptance is achieved for the internal computing environment, businesses must then test the complete application infrastructure including those variables that are outside of the firewall, such as the firewall itself, ISP bandwidth, routers, switches, peering relationships and the external networking components. For example, the nature of Internet traffic may result in networking packet loss, which typically results in increased connection times, server load and network loads. Businesses need to test from outside of their own firewall to understand how packet loss will affect the performance of their system. Other performance issues commonly uncovered with load tests driven from outside the firewall are insufficient bandwidth allocated by an ISP or faulty routers.

Step 2: Performance Regression Testing: Most people think of functional testing when they hear the term regression testing, but it can also be applied to load/performance testing. Performance regression testing is the process by which the web application is continuously re-tested in pre-production and measured against the established benchmarks to ensure that any pre-production application changes and/or infrastructure modifications do not result in degradation of performance. The benefit of performance regression testing is repeatability. Once load tests are executed and performance issues are uncovered, the application must be re-tested to validate that appropriate changes made in an effort to enhance performance, do in fact resolve the issues. If performance improves significantly during an iteration of regression tests, it is common that the benchmark metrics be augmented with newer metrics for future tests in order to reflect the improvements made to the application. This technique raises the bar for achieving higher performance goals and ensures a path of continual improvement.

Lifecycle Stage 3: Production and Post- Deployment Modifications (Post-production)

*(Key Participants: operations, developers,
CTO, QA engineers)*

Step 1: Production Monitoring: Taking performance measurements does not end once an application has been moved into production. However, instead of emulating heavy volumes of business users to test the site, the application will now be stressed by real business users. It is essential to monitor the performance of the site continuously to spot problems before customers. Using scripts built for load testing, single user scripts executing multiple types of transactions can be periodically run while the environment is live to validate response times and correct application behavior. Keynote's Test Perspective, Snapshots Edition, provides this type of functionality (see sidebar for more information). This is especially useful just after deployment when the site may experience problems that were missed in the pre-production testing phases. By leveraging production monitoring techniques, data pertaining to real life usage may be collected to help refine the load tests for future releases of the application to more accurately reflect the conditions the site will encounter. Also, consider multiple perspectives for performance monitoring. Businesses will want to monitor from outside of the firewall to gain an understanding of how various users are experiencing their business system and will also want to monitor how the components within their own environment handle daily fluctuations of transaction volumes.

Using load testing results from the pre-production phase, define performance thresholds to monitor against in production. If performance thresholds fall out of bounds (i.e., transaction response times exceeding business-defined limit), action needs to be taken (i.e., server resets, notification of appropriate parties via pager, email etc.). Ongoing monitoring proves to be a proactive performance validation practice for ensuring correct handling of daily transaction volumes. Performance monitoring is a preventative measure leveraged to minimize and/or eliminate downtime and slow time instances. It allows businesses to keep a watchful eye on their enterprise business systems and is necessary given the unpredictable nature of Internet traffic and user behavior.



Step 2: Post Production Modifications: Load tests must be repeated as applications are upgraded or changed with new features and functions, and as the infrastructure assimilates new hardware, software or configuration changes. Depending upon the change that occurs – whether to the application or the infrastructure – the testing lifecycle for load testing would loop back around to earlier stages. Will the upgrade of a new feature fit into the application design? If so, testers will want to check that the feature meets design specifications once it is developed and update load tests to adequately test the new feature, validating that it does not negatively impact performance.

Step 3: Regularly Scheduled Load Tests: In production real end users are firing transactions against the web application and load testing must continue. We suggest scheduling load tests on a regular basis (i.e., monthly) to ensure that changes in the profiles of users are not beginning to negatively impact performance. Regularly scheduled load test also work to keep businesses in step with business growth.

DEVELOPING MEANINGFUL LOAD TESTS TO MEASURE AND ASSESS PERFORMANCES

In order to develop meaningful tests to measure and assess application and infrastructure performance, there are many variables that must be considered as tests are designed and built. The goal is to develop load tests that have a high degree of realism; meaning that tests should be modeled as closely as possible to the user behavior that is expected in the live environment.

Begin by defining how the web application will be used. For businesses that have applications running in a production environment, analyze log files to understand traffic patterns and user behaviors. Variables such as the number of hits per hour, per day or per week, hit size and session duration will provide insight into the level of stress an application will endure.

Next, define all of the possible types of paths and transactions that your user audience can initiate. For example, users may be categorized as *information gathering visitors or transaction visitors*. Depending on the type of business, a more detailed and granular breakdown of users and the corresponding transactional paths they can traverse in each of those general categories is required.

It is suggested that the highest priority paths, such as paths that lead to purchases and/or critical customer service tasks, be identified and tested as fixed paths to ensure correct behavior and acceptable performance. In addition, tests should build in users that traverse through randomly generated user paths. This mix will provide a balance because highly prioritized paths will be well tested under varying load levels to evaluate capacity, endurance and the ability to handle large influxes of traffic spikes, while the randomization of paths will provide more combinations of paths and greater test coverage under the varying loads necessary to evaluate capacity, endurance and traffic spikes. This strategy has proven successful because of the size and number of links on a typical enterprise-scale web site. Balancing how tests are built to validate high priority fixed paths and lower priority randomized paths has worked to balance adequate test coverage with time-to-market deadlines. It is suggested that businesses identify a load testing product or service that has the ability to accommodate randomly generated user paths without the need for a large number of hard coded scripts, which prove very time-consuming and expensive to create and maintain.

Upon understanding how a web application will be used, profile an audience of users (user behavior modeling) with variations of characteristics to develop useful and realistic scenarios for performance tests. Remember that all web users are not alike. Web users have different levels of experience, familiarity, patience and leverage unique combinations of technologies to access and use a web site. Therefore, load tests make the mistake of testing with only one type of user behavior profile will not be representative of real-world visitor populations and the accuracy of load test result will be severely jeopardized.

Variables to Consider for Profiling Users and Creating Scenarios to Emulate Real-World Traffic

Connection Speed: Define connection speeds that are representative of the anticipated population of users. For example, a percentage of users will connect with dial-up modems, a percentage will use broadband connections such as ISDN, cable and/or DSL, and other percentages will connect via a T1 or T3 connection, etc.

Think Times: Vary think times for each virtual user. Think times are segments of time that a user is inactive with the application during an active session. It is best to rely on randomly generated think times in order to mimic real-world usage.



Data Transfer: The amount of data requested by individual users. Vary the amount of data and set up the source where that data will come from for the load tests. For load tests that run over extended periods of time to test the endurance of a site, set up virtual users that will request more data over time.

Browser Variability: Consider which types and corresponding versions of browsers will be used to interact with the application under test. Consider that different browsers will load images and frames differently, interpret JavaScript and various web services differently, allocate cache space and use cookies differently. Build this information into user profiles accordingly.

Communication Protocol: HTTP/HTTPS are the most common protocols users will rely on to access a site. It may be an important consideration for some sites to want to test with users that rely on other protocols such as FTP, POP3, SMTP and IMAP.

SSL Support: To protect the privacy of users and their specific information, SSL encrypts data as it is being sent to a specific user. A user typically verifies identity via use of a password to gain access to sensitive information. Encrypted data requires more computing resources and can be a drain on performance, therefore if an application uses SSL, load tests should reflect the percentage of users that will be accessing the system in a secure fashion.

Streaming Media: If your site features streaming media, be sure to build users into load tests that utilize this site feature to understand how performance will be impacted. Note that this feature tends to lead to a higher amount of user aborts/dropouts because this technology requires significant CPU and bandwidth computing resources, affecting user tolerance.

User Aborts: Configure tests that account for a percentage of user aborts. Impatient users or those that have begun to process an incorrect request will hit the STOP button on their browser to cancel the request. This ties up resources that can impact performance and therefore it is important to validate that your system is handling user aborts properly, dropping the requests and saving computing resources.

IP Address: Real users will connect to a site with unique IP addresses that must be maintained and tracked throughout an open session. When emulating traffic for load tests, assigning unique IP addresses to each virtual user will elevate realism. If the application infrastructure is reliant on load balancing technology, test to be sure that requests from a particular IP address

repeatedly get sent to the same server. Also test to validate an appropriate distribution of load to various servers to ensure that no single computing resource is relied upon more heavily than another with like capabilities.

Once all specific paths and transaction types are defined, prioritized and built into load tests that rely on realistic user profiles using the variables outlined above, the load test set up can begin. Complete the set up of load tests by accounting for variables such as ramp up speed/increments and ramp down speed/increments. Define the duration of the complete test as well as the duration of time peak loads will be sustained.

Another consideration when driving load tests from outside of the firewall is the geographic location from which those tests will be driven. Determine if the web application has a specific geographic dependency. For instance, an automotive B2B exchange may have the majority of its users located in Detroit or the State of Michigan. If this is the case, it is more important to conduct load tests from across the state rather than across the world. By comparison, on-line retail applications that are typically open to all users should consider load testing from several global locations. The advantage of conducting test in this manner is to fully understand and measure the users experience from the several geographic locations to see if there are any performance variations. Using comparison results from load tests conducted via multiple geographic locations can help business and technology executives make key decisions about performance. For example, should they consider relying on a third party infrastructure company, such as Akamai, to increase their infrastructure reliability? Newport Group research reveals that on average, most businesses choose to drive load tests from no more than three key geographic locations. Once all appropriate parameters are configured, begin execution of the load test.

One approach to executing load tests is to do so independent of any live traffic having access to the web site. Another approach is to conduct tests within the live environment, during non-peak business hours. The risk involved in testing in the live environment is that the site is open to both the simulated traffic generated by the load test as well as some level of real Internet traffic. It is recommended that load test execution take place during non-peak hours to minimize potential impact to actual users. It is important to have a load testing tool or service that can be easily controlled as test results are viewed in real-time. This gives quality assurance engineers the ability to stop the tests as site performance reaches a risky level.



Load Test Metrics and Their Meaning

As tests are executed, metrics such as round trip response times for transactions, HTTP requests per second and transactions per second should be collected. A random selection of path traversals and transaction types should be fired against the application to understand how it will perform as concurrent user volumes are ramped up and down. Note that as volumes of active users on the site are ramped down, it is important to check that computing resources are freed accordingly. Watch statistics such as CPU utilization, memory and disk space on individual web, application and database servers and make sure those numbers recede as load decreases.

The following are key metrics used to measure web application performance.

Response Time: The amount of time required for a transaction to complete. This metric is very important to understanding how an end user will experience an application. Response times can reveal how long it takes to load pages as volumes of concurrent users increase.

Throughput: Measured in Megabits per second (Mbps), throughput is a measurement of the number of requests processed by a web application over a given period of time. It is based on the amount of bandwidth being sent by a site over a given period of time.

HTTP Transactions per Second and Sessions per Second: The number of GET or POST requests arriving at a site. Session per second is closely related. It is the amount of users that arrive at a site per second.

Concurrent Users: The amount of users with open sessions on a particular site at a given time. As the number of concurrent users increases, system resource utilization increases.

Processor Utilization, Memory and Disk Space: CPU utilization, memory and disk space should be monitored for each of the server machines (web, application, database) comprising the complete application infrastructure. As load tests are executed and concurrent user levels increase, monitor server statistics to ensure that servers are not busy or overloaded. These statistics are especially useful in tuning various servers and operating systems for optimal performance efficiency.

Network Statistics: Monitor network statistics as load is increased to ensure appropriate bandwidth. Typically if the percentage of network bandwidth in use exceeds 40%, the network is approaching a level where it can be the bottleneck for the application.

By correlating metrics gained from various infrastructure components (servers, networks, load balancers, firewalls, etc.), with the specific behavior patterns of an application, the optimization of performance is achieved for the coordinated business system. There are unique metrics to look at based on a businesses choice of infrastructure (i.e., UNIX or Windows). Couple this with the unique user behavior patterns for an individual business and the challenge really lies in knowing how to interpret load test results and turn that information into the actionable tasks that will lead to performance gains. Keeping track of performance test results is recommended so that successive load tests and performance metrics can be compared, providing a historical perspective of performance over time.

Businesses must continuously anticipate problems. This is why performance testing and monitoring has become such an important discipline as it relates to transacting business over the web. Proper planning, identification of appropriate metrics, execution of relevant load tests as both technology and business changes occur and the careful analysis of the resulting data is the best strategy for making critical business decisions.

CONCLUSION

Web-enabled applications and infrastructures must be able to execute evolving business processes with speed and precision while sustaining high volumes of changing and unpredictable user audiences. Load testing gives the greatest line of defense against poor performance and accommodates complementary strategies for performance management and monitoring of a production environment. The discipline helps businesses succeed in leveraging Web technologies to their best advantage æ enabling new business opportunity, lowering transaction costs and strengthening profitability.

Fortunately, robust and viable solutions exist to help fend off disasters that result from poor performance. Automated load testing tools and services are available to meet the critical need of measuring and optimizing complex and dynamic application and infrastructure performance. Once these solutions are properly adopted and utilized, leveraging an ongoing, lifecycle-focused approach, businesses can begin to take charge and leverage information technology assets to their competitive advantage. By continuously testing and monitoring the performance of critical software applications, business can confidently and proactively execute strategic corporate initiatives for the benefit of shareholders and customers alike.



About Keynote Systems

Based in San Mateo California, Keynote Systems offers both full and self-service-based load testing solutions. Specifically, the two core load testing service offerings are Test Perspective™ and LoadPro™. A brief description of each follows.

Test Perspective: Test Perspective is a self-service, on-demand subscription based testing solution. It provides two levels of functionality within its Load Test Edition and Snapshots Edition. The Load Test Edition of Test Perspective offers the capability to test web applications under multi-user load. User actions are defined by a set of scripted scenarios and customized data for path variability. The Snapshots Edition of Test Perspective offers the capability to run single-user, single scenario tests from one or more geographic locations simultaneously.

Test Perspective is capable of executing load tests and snapshots of transaction performance using the same scripts. By downloading a script recorder from the Test Perspective portal, business users have the flexibility to define, create and run test scripts on their own schedule. Once scripts of any length have been recorded, they can be run either locally, or uploaded to the Test Perspective Portal. By running scripts through the Test Perspective portal, tests can be run and viewed immediately or scheduled for a specific time. Whether tests are run prior to an application going live, or after changes or modifications have been made, Test Perspective provides a flexible, low cost load testing solution to validate performance in real time from one or more geographic locations.

LoadPro: LoadPro is a full-service, outsourced testing service capable of meeting enterprise testing needs. LoadPro provides an accurate representation of real world Internet conditions by incorporating user behavior modeling, arrival rates and dynamic visitor patterns into tests. In addition, LoadPro test scenarios utilize randomly generated paths and searches to hit URL's in various combinations to accurately simulate real world Internet conditions. LoadPro test results are combined with analysis from Keynote testing professionals to provide performance-enhancing recommendations. A unique feature of the LoadPro service is that it can provide revenue projections versus load projections to estimate the amount of lost revenue that will result from varying levels of performance degradations, a feature that is very useful to business managers mindful of measuring their load testing ROI.

Both Test Perspective and LoadPro are solutions that eliminate the requirement to purchase hardware to conduct tests. These solutions also eliminate the cost associated with upgrading in-house testing software and corresponding maintenance fees. With enterprise-scale service-based solutions, Keynote has positioned itself as a provider of robust load testing solutions and consulting expertise. With a solid base of satisfied customers, Keynote solutions can be implemented quickly, provide enough functionality, accuracy and flexibility to run and analyze tests at any point during the development lifecycle and deliver a quick return on investment. Additional information about these services can be found at www.keynote.com.



About Newport Group

Newport Group is an independent information technology research firm. Founded in 1997, Newport Group was created with the intent and interest to provide detailed research services to major corporations and software vendors that share a vital interest in information technology. Drawing on years of IT research experience, Newport Group concentrates on selective IT research topics and trends. The benefits of this exclusive research approach are passed on to the client base with demonstrated expertise and insight.

The information contained in any and all final written materials by Newport Group will be based on accurate reliable information. However, because Newport Group information is gathered through numerous sources, including third parties, we cannot warrant that the final publication is complete and error-free. Newport Group disclaims all implied warranties, including, without limitation, warranties of merchantability of fitness for a particular purpose. Newport Group or its affiliates shall have no liability for any direct, incidental, special or consequential damages or lost profits.

© Copyright 2002 Newport Group, Inc. Reproduction is forbidden unless authorized
Additional copies may be obtained by contacting 1-508-362-9799

3230 Main Street, P.O. Box 909, Barnstable, Massachusetts 02630
Telephone: 508-760-3072 www.newport-group-inc.com Fax: 508-760-3705