White Paper | **Load Testing Meta-Analysis**

Authors:

Joel Weierman & Anu Krishnan

Improving Performance and Identifying Common Application Bottlenecks

Introduction

In order to better understand the most common types of performance and capacity issues that large-scale web sites experience Webmetrics conducted a detailed review of 375 recent load tests, selected from a larger battery of tests that we have managed over the past decade.

Analysis Scope

We specifically looked at the types of errors that were encountered as well as the overall percentage performance improvement achieved as part of the testing efforts. We also investigated the different types of bottlenecks (web server, database, etc.) an external load test is likely to uncover for a given application.

Analysis Parameters

The key parameters that were considered during the analysis are:

- Throughput (total transfer and page views)
- Errors (number and type)
- Average page load performance (seconds)
- Industry (travel, e-commerce, education)
- Improvement across the engagement (comparing metrics between the 1st and subsequent tests)

Conclusion

In addition to verifying system load capabilities and determining scalability and reliability, *we found that 76% of the time, load testing also aided in identification of bottlenecks in websites, the resolution of which resulted in significant improvement in performance as more tests were conducted.*

Results Summary

A detailed analysis of approximately 375 load tests was done to understand the common types of bottlenecks and errors observed by websites and quantify the improvements achieved due to load testing.

Key Parameters Analyzed:

- Throughput
- Errors
- Average page load performance (seconds)
- Industry (travel, e-commerce, education)
- Improvement across the engagement

Key Observations:

- 76% of the time, in addition to capacity verification, load testing engagements aided in identification of bottlenecks and overall improvements in website performance.
- The common bottlenecks observed were:
  - Broken web response
  - Application
  - High page load times
  - Low throughput / bandwidth limitations
  - Server issues
  - Load balancing
  - CPU limitations
  - Software versioning / software defects
- The common types of errors observed during the load tests were:
  - Internal server errors
  - Timeout errors
  - Content errors
  - Unable to connect
  - Others

Contents

Analysis Synopsis

The tests conducted were sourced from over 20 different industries including Travel, IT, Health, Education, Food, E-commerce, Insurance and News. The following table summarizes the results obtained:

Total Customer Engagements	83
Total Number of Tests Performed	375
Median Number of Test Iterations	3
Average Number of Concurrent Users	2000
Number of Customers with Performance Improvement	63
Average Page View Increase across Load Testing Engagement	335,737
Average Throughput Increase (Mbps)	94
Average Percentage Decrease in Page Load Times	63%

As the above table indicates, not only did a majority of customers show improvement, on average, the overall throughput of the application(s) increased by over 300,000 page views. In addition, many customers also were able to improve their page load times by more than 60% over the course of the engagement. Finally, many of these engagements were sized very moderately below the 2,000 concurrent user range, showing that a wide range of applications and industries can experience significant performance benefits just by going through the process of load testing and tuning their applications.

Common Bottlenecks Uncovered

Summary

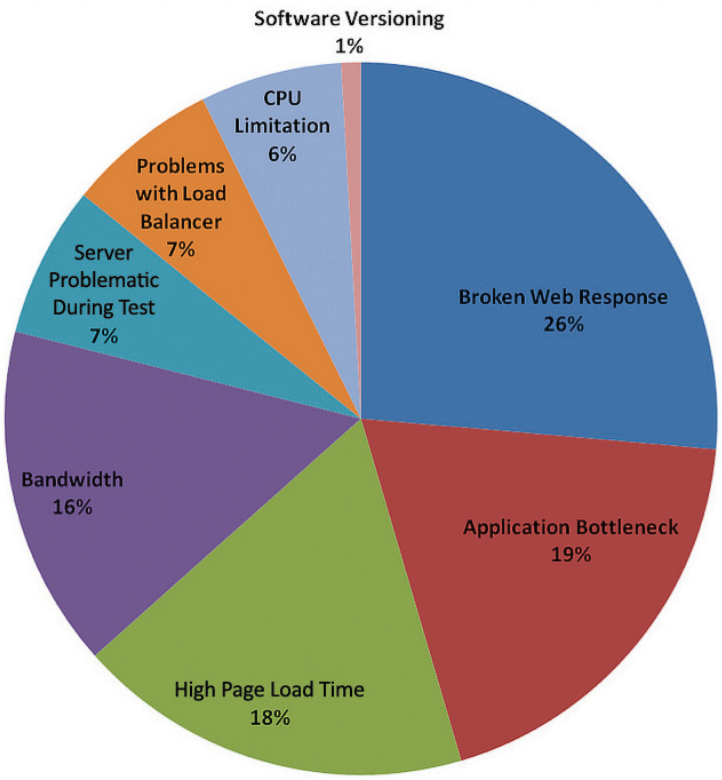
Analysis was done to find the common bottlenecks that were repeatedly uncovered as part of the load testing execution and subsequent analysis. The most common types of bottlenecks discovered are listed and explained below in order of most to least frequently observed:

Types of Bottlenecks:

1. **Broken Web Response:** This bottleneck constitutes page errors due to unexpected content (404 page returned), unexpected behavior (failed log in), and 500 internal server errors.
2. **Application:** There were cases where the website experienced high page load times and application errors due to a poorly written or inappropriately tuned application.
3. **High Page Load Times:** High page load times are either caused by high 1st packet times or transfer times or both. High 1st packet times are usually the result of server or CPU related issues whereas high transfer times are usually the result of network transfer or capacity problems.
4. **Low Throughput / Bandwidth Limitations:** The expected behavior for any website is a gradual rise in throughput with increasing load. Network saturation, router limitations and other factors often lead to early peak or large fluctuations in throughput.
5. **Server Issues (various):** Server related issues usually lead to high times to 1st packet. This bottleneck is also usually the reason behind errors such as timeouts, service unavailable and closed connection errors.
6. **Load Balancer:** A load balancer plays an important role in evenly distributing the received requests among servers. Load balancing issues could lead to network latency issues and increased server response times.
7. **CPU Limitations:** CPU limitations are typically caused by an overworked web or database server.
8. **Software Versioning / Software Defects:** Sometimes customers make changes after a load test, which lead to worse performance during the next test. In these cases the application was rolled back to a previous version.

Distribution of Bottlenecks

During analysis of the load tests, the various types of bottlenecks were recorded and their frequencies of occurrence were plotted to generate the following pie chart. The predominant types of bottlenecks noticed were page errors, high page load times and hardware/application bottlenecks.



Analysis of Error Types

Summary

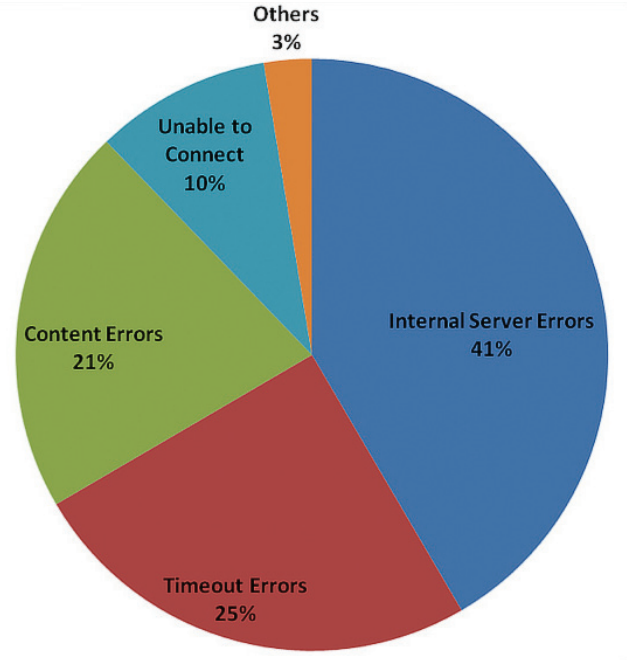
The main types of errors encountered during a typical load test can be categorized as follows. Errors are listed from the most frequently observed to least frequent.

Types of Errors:

1. **Timeout Errors:** This error occurs when time taken to process the request sent to the web server takes longer than the maximum wait-time set on the server.
2. **Internal Server Errors:** This error is usually caused when a web server encounters an unexpected condition causing a failed web request. This error includes 500 errors and unable to post errors.
3. **Content Errors:** This corresponds to the error that is reported when the web page loaded does not have the expected content.
4. **Unable to Connect:** This error manifests when the number of connections attempted to a server exceeds the set limit of the number of connections the server can process at a time.
5. **Others:** These errors include 400 and unable to resolve errors.

Distribution of Errors:

The following pie chart shows the frequency of the different types of errors encountered across all the tests:



From the chart, it can be seen that the most frequent types of error witnessed during the tests were internal server errors, timeout errors and content errors.

Top 10 Observed Improvements

Customer	Increase in Successful Page Views	% Reduction in Errors	% Decrease in page load times	Increase in Peak Throughput (Mbps)
1	449,444	-43.3	95	195
2	171,492	-0.24	98	2
3	166,194	-96.969	57	31
4	136,491	-5.74	77	29
5	120,884	-29.13	74	30
6	88,868	-0.92	46	7
7	71,836	-77.9	91	7
8	56,010	-90.5	60	8
9	39,119	-73.77	99	18
10	5,175	-40.74	10	8

The table summarizes the top 10 load testing engagements where significant improvement was recorded, in terms of one or more of the following factors: 1) successful page views, 2) reduction in errors, 3) percent decrease in page load times and 4) increase in peak throughput. Details on how these metrics were calculated are elaborated upon in the Appendix to this report, under the “Data Compilation” section (page 10).

Common Performance Optimizations

In general, customers use a variety of techniques to improve performance of their web applications, according to the baseline measurements that external load testing provides. During the process of analyzing the various customer engagements for this white paper, we found that these activities generally can be broken down into three categories, in order of most frequently used and most effective as measured by the test results:

1. Application performance tuning (code and database optimizations)
2. Adjustments to load balancer/router/firewall configurations
3. Addition of additional infrastructure such as memory, servers or bandwidth

Conclusion

As the analysis from this white paper indicates, over 75% of customers who conducted a load test experienced significant and quantifiable improvements in the performance of their web application(s). These improvements were often achieved by simple improvements to application or database code and generally not by the purchase of costly hardware and additional infrastructure. In conclusion, external performance monitoring and load testing continue to be the most reliable and consistent method of achieving significant gains in application performance with a minimal amount of investment.

Appendix

Additional Information on Data Compilation

Approximately 375 load tests were analyzed and organized into a database comprising a number of key performance indicators from each of the tests:

Aggregate Metrics for Analysis

The following performance metrics were used to determine if there was an improvement between the start and the end of a load testing engagement:

1. **Number of Successful Page Views:** The improvement was analyzed by looking at the increase in the number of page views across the load testing engagement.
2. **Errors (number and type):** The different types of errors and their frequency of occurrence were tabulated to find the distribution of the common errors that occur during load tests. Also, the decrease in error was used as a metric to determine improvement in performance of a website over the course of a load testing engagement.
3. **Average Page Load Performance (seconds):** In each load test, the pages that took the maximum load time were tabulated along with their corresponding load times and this was compared to the same page's load time with the same number of concurrent users at the last iteration to check for improvement in page load times. The percentage decrease in these times from the first to the last iteration was used as another metric for measuring improvement.
4. **Increase in Peak Throughput (Mbps):** The increase in peak throughput from the first to the last iteration of a load testing engagement was also used to quantify improvement in website performance.

After the data for the individual load testing engagements was tabulated, the average of all the above mentioned metrics was computed for the load testing engagements that showed improvement. These computed values are displayed in the "Analysis Synopsis" section (page 4).

Load Testing Methodology and Test Parameters

Summary

During each load test, one or more scenarios or user behaviors are simulated in parallel. The distribution of load between the scenarios is set based on customer requirements. The typical load test duration is 60 minutes. The main test parameters are:

1. **Number of Intervals:** The load test is split into a number of intervals. Users can be ramped up at the start of an interval or the load can be maintained at the previous level. The number of intervals is customizable. Typically, tests are configured with 10 intervals.
2. **Length of Interval:** This is the duration of an interval. It is usually set to 6 minutes for most test configurations.
3. **Page Think Time:** This is the time paused between each step within a scenario. We have observed from the past that users usually take 3 to 7 seconds to think before performing an action such as clicking on a link in a website. So this parameter is randomized between 3 to 7 seconds to closely simulate real user behavior. This can also be customized on a per customer basis.
4. **Ramp in Time:** A sudden ramp-up of users at the start of an interval could cause the servers to overload. To prevent this, the users are generally ramped up over a duration specified for the ramp-up time parameter at the start of an interval. This value is usually set to 2 minutes.
5. **Concurrent Users and Split:** This parameter specifies the total number of concurrent users accessing the site at the end of the last interval. Most tests are configured to achieve the maximum load in the last interval of the test. This can be altered per each customer's individual requirements. The load is also typically then sub-divided amongst two or more unique user behaviors, or scenarios.

About Neustar® Inc.

Neustar, Inc. (NYSE: NSR) solves complex communications challenges and provides market-leading, innovative solutions and directory services that enable trusted communication across networks, applications, and enterprises around the world. Neustar Webmetrics services provide website monitoring and testing for companies that want to ensure online performance, competitive advantage, and a positive end-user experience. Visit us at www.neustar.biz and www.webmetrics.com.