Load Test Report

Date: 5/28/2015

Test from: virginia

Query URL: reviewsignal3.wpengine.com

Started at: Thu May 28 2015, 12:05:34 -04:00

Finished at: Thu May 28 2015, 12:06:34 -04:00

Test link: https://www.blitz.io/to#/play/input/virginia:50174c781f6ce4ff8e7a2f7af8e92f12

Analysis

This rush generated 56,277 successful hits in 60 seconds and we transferred 639.06 MB of data in and out of your app. The average hit rate of 938/second translates to about 81,038,880 hits/day.

The average response time was 27 ms.

You've got bigger problems, though: 1.45% of the users during this rush experienced timeouts or errors!

Response Times

Fastest: 21 ms
Slowest: 70 ms
Average: 27 ms

Test Configuration

Region: virginia
Duration: 60 seconds
Load: 1-2000 users

Other Stats

Avg. Hits: 938/sec
Transferred: 8.01 MB
Received: 631.06 MB

Hits

This rush generated 56,277 successful hits. The number of hits includes all the responses listed below. For example, if you only want HTTP 200 OK responses to count as Hits, then you can specify --status 200 in your rush.

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>HTTP</td>
<td>OK</td>
<td>56277</td>
</tr>
</tbody>
</table>

Errors

The first error happened at 32.5 seconds into the test when the number of concurrent users was at 1082. Errors are usually caused by resource exhaustion issues, like running out of file descriptors or the connection pool size being too small (for SQL databases).

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>TCP</td>
<td>Connection timeout</td>
<td>112</td>
</tr>
<tr>
<td>500</td>
<td>HTTP</td>
<td>Internal Server Error</td>
<td>715</td>
</tr>
</tbody>
</table>

Timeouts

The first timeout happened at 60 seconds into the test when the number of concurrent users was at 2000. Looks like you've been rushing with a timeout of 1000 ms. Timeouts tend to increase with concurrency if you have lock contention of sorts. You might want to think about in-memory caching using redis, memcached or varnish to return stale data for a period of time and asynchronously refresh this data.

https://www.blitz.io/to/#/play/input/virginia:50174c781f6ce4ff8e7a2f7af8e92f12
Response Times

The max response time was: **70 ms @ 1916 users**

Hit Rate

The max hit rate was: **1,825 hits per second**