Load Test Report

Date: 5/27/2015

Test from: virginia

Query URL: wpnew.reviewsignal.com

Started at: Wed May 27 2015, 04:08:37 -04:00

Finished at: Wed May 27 2015, 04:09:37 -04:00

Test link: https://www.blitz.io/to#/play/input/virginia:67573881efd0185ee644dbad7a6f67fc

Analysis

This rush generated 29,649 successful hits in 60 seconds and we transferred 378.83 MB of data in and out of your app. The average hit rate of 494/second translates to about 42,694,560 hits/day.

The average response time was 162 ms.

You’ve got bigger problems, though: 46.22% of the users during this rush experienced timeouts or errors!

Response Times

<table>
<thead>
<tr>
<th>Fastest</th>
<th>Slowest</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>104 ms</td>
<td>1,103 ms</td>
<td>162 ms</td>
</tr>
</tbody>
</table>

Test Configuration

Region: virginia

Duration: 60 seconds

Load: 1-2000 users

Other Stats

Avg. Hits: 494 /sec

Transferred: 4.06 MB

Received: 374.77 MB

Hits

This rush generated 29,649 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>29649</td>
</tr>
</tbody>
</table>

Errors

The first error happened at 20 seconds into the test when the number of concurrent users was at 664. 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>25356</td>
</tr>
</tbody>
</table>

Timeouts

The first timeout happened at 5 seconds into the test when the number of concurrent users was at 164. 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.
The max response time was: **1102 ms @ 1916 users**

The max hit rate was: **1,268 hits per second**