Load Test Report

Date: 7/18/2016

Test from: virginia

Query URL: http://agency.reviewsignal.com/
Started at: Mon Jul 18 2016, 04:19:26 -04:00
Finished at: Mon Jul 18 2016, 04:20:26 -04:00
Test link: https://www.blitz.io/to#/play

Analysis

This rush generated 44,310 successful hits in 60 seconds and we transferred 739.65 MB of data in and out of your app. The average hit rate of 739/second translates to about 63,806.400 hits/day.

The average response time was 249 ms.

You've got bigger problems, though: 43.44% of the users during this rush experienced timeouts or errors!

Response Times

| Fastest: 189 ms | Slowest: 676 ms | Average: 249 ms |

Test Configuration

| Region: virginia | Duration: 60 seconds | Load: 1-3000 users |

Other Stats

| Avg. Hits: 739/sec | Transferred: 6.11MB | Received: 733.54MB |

Hits

This rush generated 44,310 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>44310</td>
</tr>
</tbody>
</table>

HTTP 200 OK 100% (44310)

Errors

The first error happened at 2.5 seconds into the test when the number of concurrent users was at 121. 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>33581</td>
</tr>
</tbody>
</table>

Connection timeout 100% (33581)

Timeouts

The first timeout happened at 12.5 seconds into the test when the number of concurrent users was at 621. 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.
**Response Times**

The max response time was: **675 ms @ 2874 users**

**Hit Rate**

The max hit rate was: **1,705 hits per second**

Powered by [www.blitz.io](http://www.blitz.io)