Load Test Report

Date: 7/16/2016

Test from: virginia

Query URL: http://2016pressable30.reviewsignal.com/

Started at: Sat Jul 16 2016, 08:05:21 -04:00

Finished at: Sat Jul 16 2016, 08:06:21 -04:00

Test link: https://www.blitz.io/to#/play

Analysis

This rush generated 77,652 successful hits in 60 seconds and we transferred 1.38 GB of data in and out of your app. The average hit rate of 1,294/second translates to about 111,818,880 hits/day.

The average response time was 134 ms.

Response Times

Fastest: 133 ms
Slowest: 141 ms
Average: 134 ms

Test Configuration

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

Other Stats

Avg. Hits: 1,294 /sec
Transferred: 11.22 MB
Received: 1,398.23 MB

Hits

This rush generated 77,652 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>77652</td>
</tr>
</tbody>
</table>

HTTP 200 OK 100% (77652)

Timeouts

The first timeout happened at 10 seconds into the test when the number of concurrent users was at 496. 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: \textbf{140 ms @ 2874 users}

The max hit rate was: \textbf{2,537 hits per second}