Load Test Report

**Date:** 6/4/2015

**Test from:** virginia

**Query URL:** http://live-reviewsignal3.pantheon.io/

**Started at:** Thu Jun 4 2015, 09:26:10 -04:00

**Finished at:** Thu Jun 4 2015, 09:27:10 -04:00

**Test link:** https://www.blitz.io/to#/play/input/virginia:e39245ea3f5bad46534b642ff509b95a

**Analysis**

This rush generated **55,814** successful hits in **60 seconds** and we transferred **767.36 MB** of data in and out of your app. The average hit rate of **930/second** translates to about **80,372,160** hits/day.

The average response time was **52 ms**.

You've got bigger problems, though: **0.22%** of the users during this rush experienced timeouts or errors!

**Response Times**

| Fastest: 52 ms | Region: virginia | Avg. Hits: 930 /sec |
| Slowest: 54 ms | Duration: 60 seconds | Transferred: 8.04 MB |
| Average: 52 ms | Load: 1-2000 users | Received: 759.32 MB |

**Hits**

This rush generated **55,814** 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>55814</td>
</tr>
</tbody>
</table>

**Errors**

The first error happened at **25 seconds** into the test when the number of concurrent users was at **831**. 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>
</tbody>
</table>

**Timeouts**

The first timeout happened at **40 seconds** into the test when the number of concurrent users was at **1332**. 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

![Response Times graph]

The max response time was: **53 ms @ 1332 users**

Hit Rate

![Hit Rate graph]

The max hit rate was: **1,827 hits per second**

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