Load Test Report

Date: 5/27/2015

Test from: virginia

Query URL: http://reviewsignal.onpressidium.com/

Started at: Wed May 27 2015, 10:07:04 -04:00

Finished at: Wed May 27 2015, 10:08:04 -04:00

Test link: https://www.blitz.io/to#/play/result/virginia:67573881efd0185ee644dbad7a6c885d

Analysis

This rush generated **47,567** successful hits in **60 seconds** and we transferred **769.79 MB** of data in and out of your app. The average hit rate of **793/second** translates to about **68,496,480** hits/day.

The average response time was **233 ms**.

You've got bigger problems, though: **0.02%** of the users during this rush experienced timeouts or errors!

**Response Times**

<table>
<thead>
<tr>
<th>Fastest</th>
<th>Slowerst</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>233 ms</td>
<td>234 ms</td>
<td>233 ms</td>
</tr>
</tbody>
</table>

**Test Configuration**

- Region: **virginia**
- Duration: **60 seconds**
- Load: **1-2000 users**
- Avg. Hits: **793/sec**
- Transferred: **6.84 MB**
- Received: **762.95 MB**

**Other Stats**

- Avg. Hits: **793/second**
- Transfered: **6.84 MB**
- Received: **762.95 MB**

**Analysis**

This rush generated **47,567** successful hits in **60 seconds** and we transferred **769.79 MB** of data in and out of your app. The average hit rate of **793/second** translates to about **68,496,480** hits/day.

The average response time was **233 ms**.

You've got bigger problems, though: **0.02%** of the users during this rush experienced timeouts or errors!

**Hits**

This rush generated **47,567** 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>47567</td>
</tr>
</tbody>
</table>

**Errors**

The first error happened at **45 seconds** into the test when the number of concurrent users was at **1499**. 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>1</td>
</tr>
</tbody>
</table>

**Timeouts**

The first timeout happened at **7.5 seconds** into the test when the number of concurrent users was at **247**. 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 was: 234 ms @ 1999 users

Hit Rate

The max hit rate was: 1,565 hits per second