LOAD TEST REPORT

DATE: 7/24/2014

TEST FROM: VIRGINIA

Query URL: http://wp-dreamhost.reviewsignal.com/

Started at: Thu Jul 24 2014, 12:24:10 -04:00

Finished at: Thu Jul 24 2014, 12:24:10 -04:00

ANALYSIS

This rush generated 10,237 successful hits in 60 seconds and we transferred 108.93 MB of data in and out of your app. The average hit rate of 171/second translates to about 14,741,280 hits/day.

The average response time was 201 ms.

You've got bigger problems, though: 71.43% of the users during this rush experienced timeouts or errors!

RESPONSE TIMES

FASTEST: 3 ms
SLOWEST: 768 ms
AVERAGE: 201 ms

TEST CONFIGURATION

REGION: VIRGINIA
DURATION: 60 seconds
LOAD: 1-2000 users

OTHER STATS

AVG. HITS: 171/sec
DATA TRANSFERED: 108.93 MB

HITS

This rush generated 10,237 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>10237</td>
</tr>
</tbody>
</table>

HITS

ERRORS 14.51% (5201)
TIMEOUTS 56.92% (20396)
**ERRORS**

The first error happened at **37.5 seconds** into the test when the number of concurrent users was at **1248**. 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>5201</td>
</tr>
</tbody>
</table>

**TIMEOUTS**

The first timeout happened at **2.5 seconds** into the test when the number of concurrent users was at **81**. 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: **768 ms @ 1582 users**

The max hit rate was: **264 hits per second**