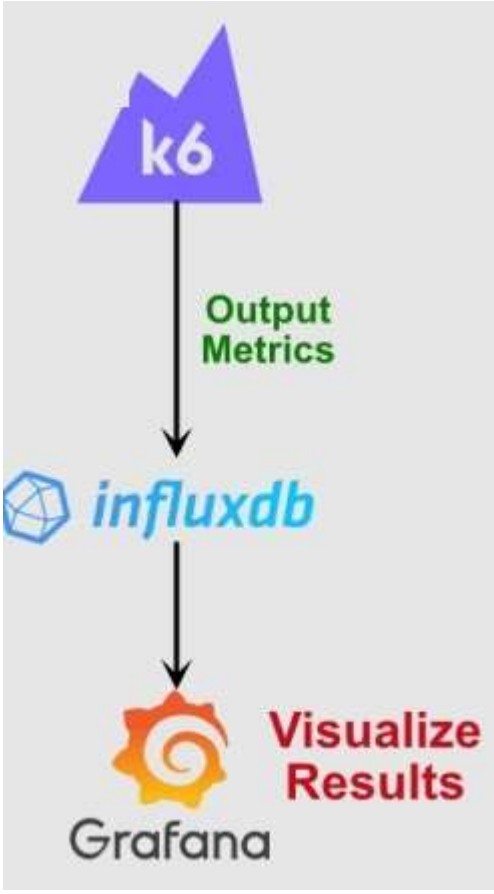
1. Explain the solution.

The target site was a search engine, which consists of 2 pages. the main page loaded first and accept an input parameter (country name in my test) and the second page show the parameter search result. For more accurate analysis, I removed the static components. (Because these components can be cached or can be handled with a static HTTP server like Apache).

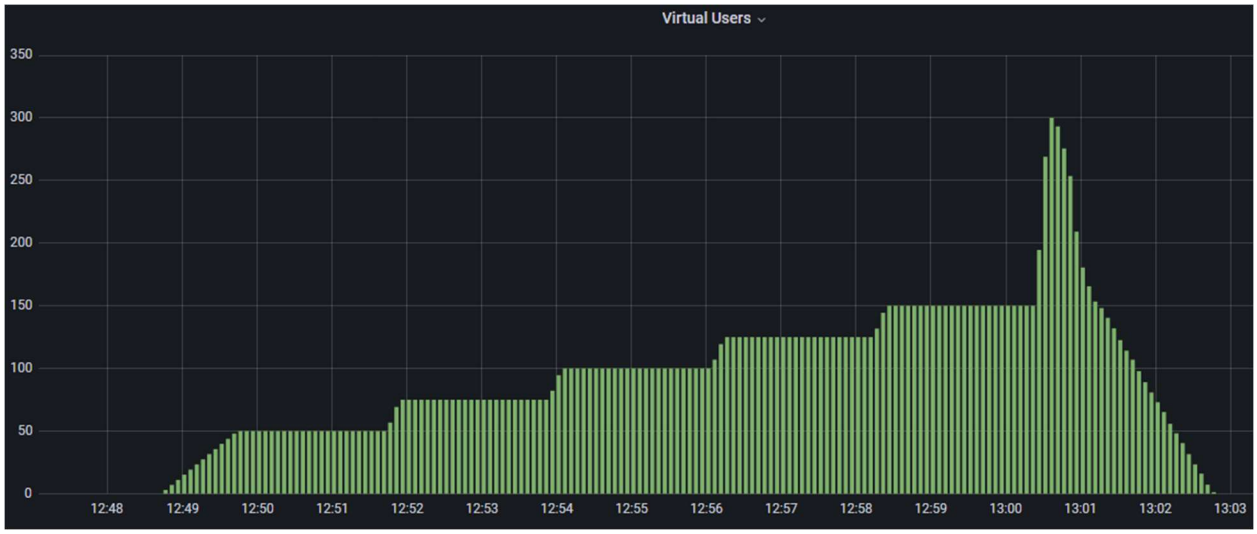
Due to the first Baseline tests, the search engine can't handle the 1000 requests so the goals (the threshold in K6) failed because of the high error rate and page failure (about 50% failure rate).

Another important point in designing the test case was that the site did not serve many concurrent requests from one IP. So, after a specific user level, all requests failed due to "closed socket" errors. So, I have to design another test plan for overcoming this issue. Finally for reducing the failure rate, I increased the response time threshold check of pages to 2000 MS.

The goal of the test is to find problems in loading and searching system. I executed the test for the suggesting subsystem in JMeter. You can find it in Github.
I used Grafana & InfluxDB tools for monitoring and presenting results.



The most important thing that made it difficult to analyze the tests was the lack of resource consumption charts.
Finally the last main VU chart designed like this.

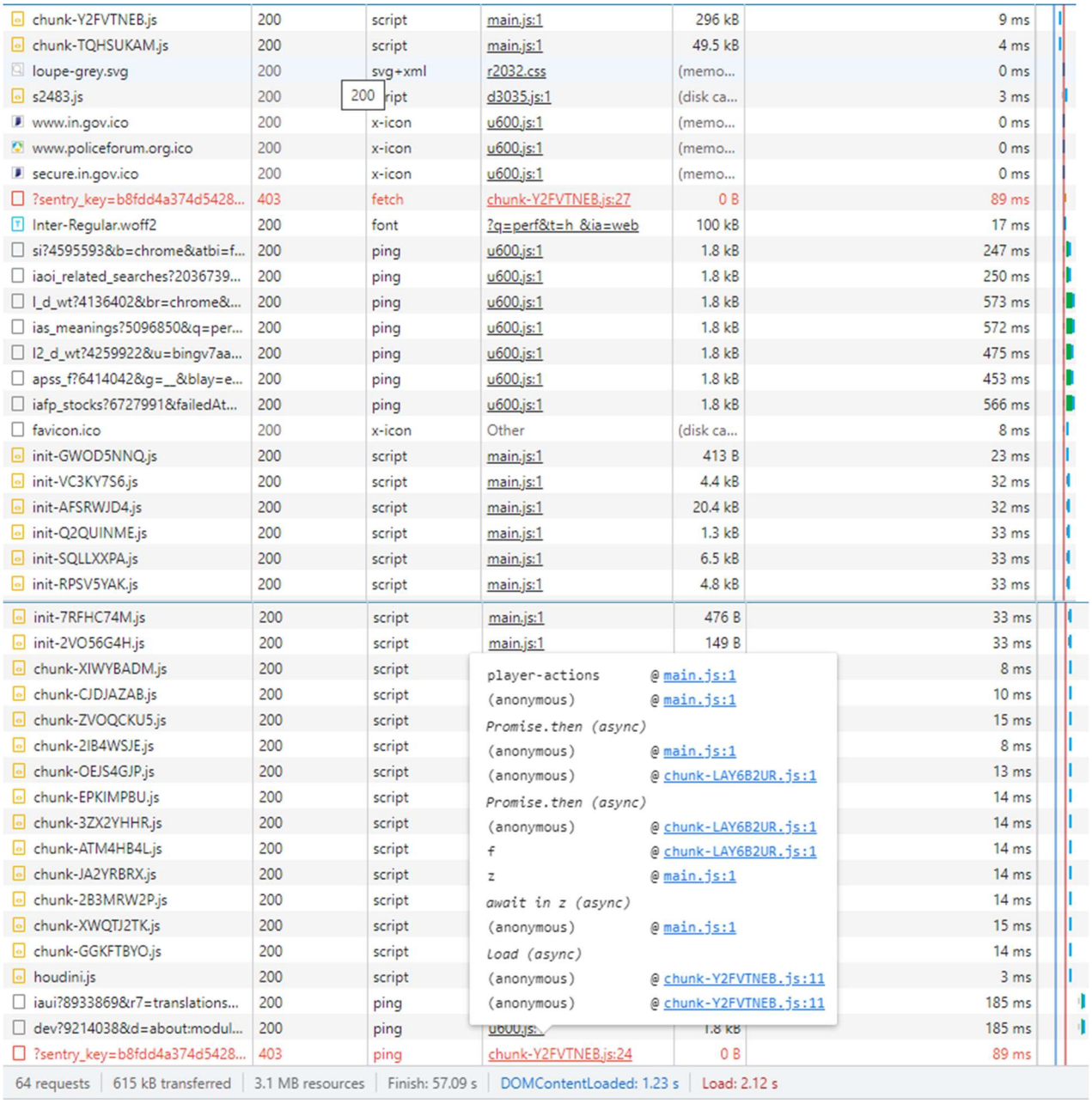


2. Analyze few HTTP/S responses in details.

The best way to analyze a response is to use web performance tools like Lighthouse. According to the reports for Loading the whole request (search page), users need to download 3.1 MB of data. there are 64 requests needed to load the page. the first response time of the page is 2.1 S. (the first user impression time).

The main request protocol is Get. So the parameter (country name) pass in URL for searching.

In the scripts, I define a check based on status code. So, if the status code is 200 , I consider the response successful.


chunk-Y2FVTNEB.js 200 script main.js:1 296 kB 9 ms
chunk-TQHSUKAM.js 200 script main.js:1 49.5 kB 4 ms
loupe-grey.svg 200 svg+xml r2032.css (memo... 0 ms
s2483.js 200 ript d3035.js:1 (disk ca... 3 ms
www.in.gov.ico 200 x-icon u600.js:1 (memo... 0 ms
www.policeforum.org.ico 200 x-icon u600.js:1 (memo... 0 ms
secure.in.gov.ico 200 x-icon u600.js:1 (memo... 0 ms
?sentry_key=b8fdd4a374d5428... 403 fetch chunk-Y2FVTNEB.js:27 0 B 89 ms
Inter-Regular.woff2 200 font ?q=perf&t=h_&ia=web 100 kB 17 ms
si?4595593&b=chrome&atbi=f... 200 ping u600.js:1 1.8 kB 247 ms
iaoi_related_searches?2036739... 200 ping u600.js:1 1.8 kB 250 ms
I_d_wt?4136402&br=chrome&... 200 ping u600.js:1 1.8 kB 573 ms
ias_meanings?5096850&q=per... 200 ping u600.js:1 1.8 kB 572 ms
I2_d_wt?4259922&u=bingv7aa... 200 ping u600.js:1 1.8 kB 475 ms
apss_f?6414042&g=__&blay=e... 200 ping u600.js:1 1.8 kB 453 ms
iafp_stocks?6727991&failedAt... 200 ping u600.js:1 1.8 kB 566 ms
favicon.ico 200 x-icon Other (disk ca... 8 ms
init-GWOD5NNQ.js 200 script main.js:1 413 B 23 ms
init-VC3KY7S6.js 200 script main.js:1 4.4 kB 32 ms
init-AFSRWJD4.js 200 script main.js:1 20.4 kB 32 ms
init-Q2QUINME.js 200 script main.js:1 1.3 kB 33 ms
init-SQLLXXPA.js 200 script main.js:1 6.5 kB 33 ms
init-RPSV5YAK.js 200 script main.js:1 4.8 kB 33 ms
init-7RFHC74M.js 200 script main.js:1 476 B 33 ms
init-2VO56G4H.js 200 script main.js:1 149 B 33 ms
chunk-XIWYBADM.js 200 script 8 ms
chunk-CJDJAZAB.js 200 script 10 ms
chunk-ZVOQCKU5.js 200 script 15 ms
chunk-2IB4WSJE.js 200 script 8 ms
chunk-OEJS4GJP.js 200 script 13 ms
chunk-EPKIMPBU.js 200 script 14 ms
chunk-3ZX2YHHR.js 200 script 14 ms
chunk-ATM4HB4L.js 200 script 14 ms
chunk-JA2YRBRX.js 200 script 14 ms
chunk-2B3MRW2P.js 200 script 14 ms
chunk-XWQTJ2TK.js 200 script 15 ms
chunk-GGKFTBYO.js 200 script 14 ms
houdini.js 200 script 3 ms
iaui?8933869&r7=translations... 200 ping 185 ms
dev?9214038&d=about:modul... 200 ping 185 ms
?sentry_key=b8fdd4a374d5428... 403 ping chunk-Y2FVTNEB.js:24 0 B 89 ms
player-actions @ main.js:1
(anonymous) @ main.js:1
Promise.then (async)
(anonymous) @ main.js:1
(anonymous) @ chunk-LAY6B2UR.js:1
Promise.then (async)
(anonymous) @ chunk-LAY6B2UR.js:1
f @ chunk-LAY6B2UR.js:1
z @ main.js:1
await in z (async)
(anonymous) @ main.js:1
Load (async)
(anonymous) @ chunk-Y2FVTNEB.js:11
(anonymous) @ chunk-Y2FVTNEB.js:11
64 requests 615 kB transferred 3.1 MB resources Finish: 57.09 s DOMContentLoaded: 1.23 s Load: 2.12 s

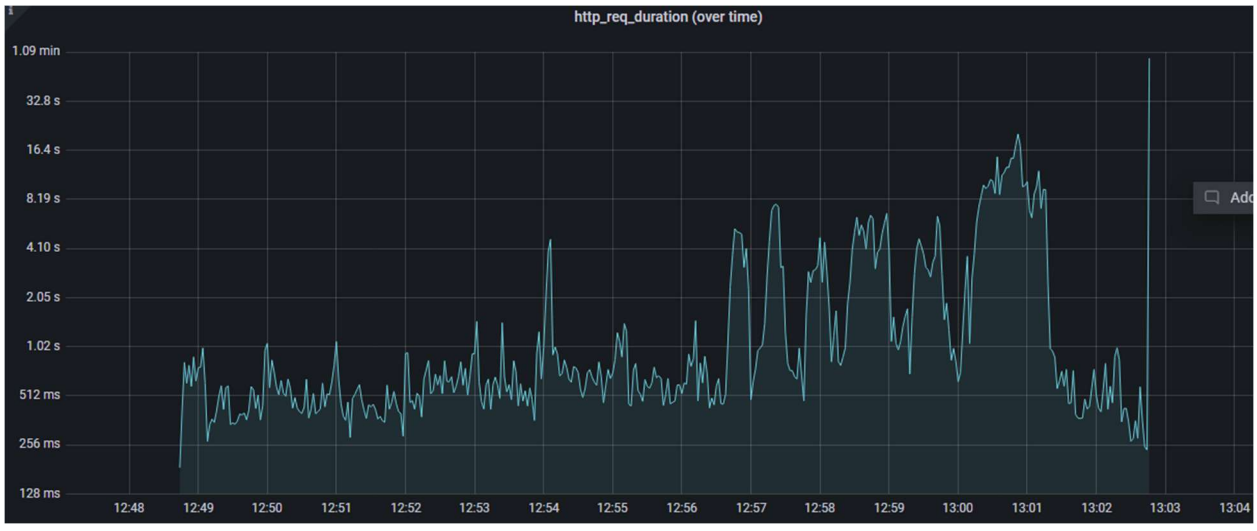
Request URL: https://duckduckgo.com/?q=perf&t=h_&ia=web
Request Method: GET
Status Code: 200
Remote Address: 40.81.94.43:443
Referrer Policy: origin

Response Headers

cache-control: max-age=1
content-encoding: br
content-security-policy: default-src 'none' ; connect-src https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; manifest-src https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; media-src https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; script-src blob: https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ 'unsafe-inline' 'unsafe-eval' ; font-src data: https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; img-src data: https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; style-src https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ 'unsafe-inline' ; object-src 'none' ; worker-src blob: ; child-src blob: https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; frame-src blob: https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; form-action https://duckduckgo.com https://*.duckduckgo.com https://3g2upl4pq6kufc4m.onion/ https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/ ; frame-ancestors 'self' ; base-uri 'self' ; block-all-mixed-content ;
content-type: text/html; charset=UTF-8
date: Thu, 20 Jan 2022 13:12:02 GMT
expect-ct: max-age=0
expires: Thu, 20 Jan 2022 13:12:03 GMT
permissions-policy: interest-cohort=()
referrer-policy: origin
server: nginx
server-timing: total;dur=28;desc="Backend Total"
strict-transport-security: max-age=31536000
vary: Accept-Encoding
x-content-type-options: nosniff
x-duckduckgo-locale: en_US
x-duckduckgo-results: 1
x-frame-options: SAMEORIGIN
x-xss-protection: 1;mode=block

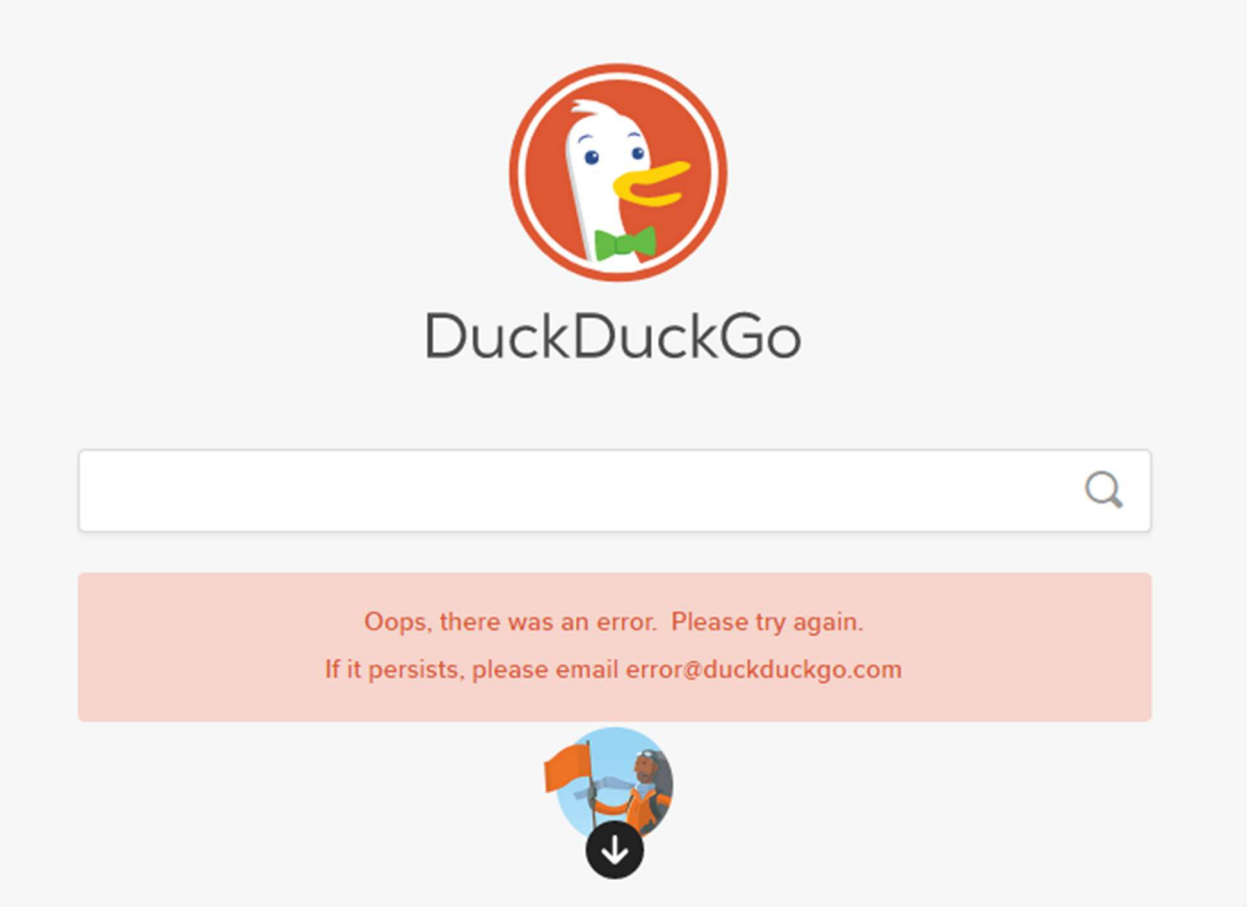
3. Did the load test have an impact on web application response time?

absolutely yes. Due to the charts, with increasing the load the response time (subsequently the error rate) increased too.



There was another problem with increasing the load. The site Banned my IP Address after increasing the load to 1000 users.

```
WARN[0718] Request Failed                                error="Get \"https://duckduckgo.com/\": read tcp 192.168.1.4:7227->52.142.124.215:443: wsarecv: An existing con
nection was forcibly closed by the remote host."
WARN[0727] Request Failed                                error="Get \"https://duckduckgo.com/\": dial tcp 52.142.124.215:443: connectex: A connection attempt failed bec
ause the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond."
WARN[0728] Request Failed                                error="read tcp 192.168.1.4:7255->52.142.124.215:443: wsarecv: An existing connection was forcibly closed by th
e remote host."
WARN[0729] Request Failed                                error="Get \"https://duckduckgo.com/?q=South Korea&t=h_\": read tcp 192.168.1.4:7506->52.142.124.215:443: wsare
cv: An existing connection was forcibly closed by the remote host."
WARN[0730] Request Failed                                error="Get \"https://duckduckgo.com/?q=Brazil&t=h_\": read tcp 192.168.1.4:7267->52.142.124.215:443: wsarecv: A
n existing connection was forcibly closed by the remote host."
WARN[0746] Request Failed                                error="Get \"https://duckduckgo.com/\": unexpected EOF"
WARN[0747] Request Failed                                error="Get \"https://duckduckgo.com/\": request timeout"
WARN[0749] Request Failed                                error="Get \"https://duckduckgo.com/?q=Ireland&t=h_\": read tcp 192.168.1.4:7274->52.142.124.215:443: wsarecv:
An existing connection was forcibly closed by the remote host."
```



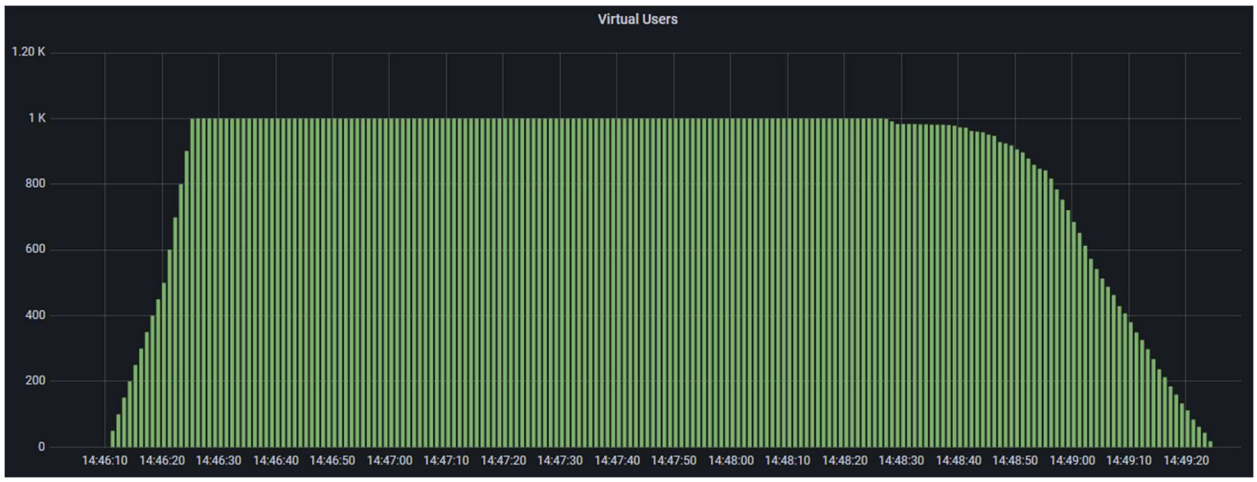
4. What is the optimal application response time for modern web applications?

I think the best answer to this question is "as soon as possible". In my opinion, the site response time should be as short as possible so the end-user doesn't sense the delay. The user won't accept a high response time or Blank & White screen phenomenon (High-Performance Web Sites by Steve Souders). The web designers have to provide progressive rendering by component parallel downloading techniques. (e.g. put JS file at the bottom of the page and put CSS files at the top of the page to prevent of Blank & White screen phenomenon).

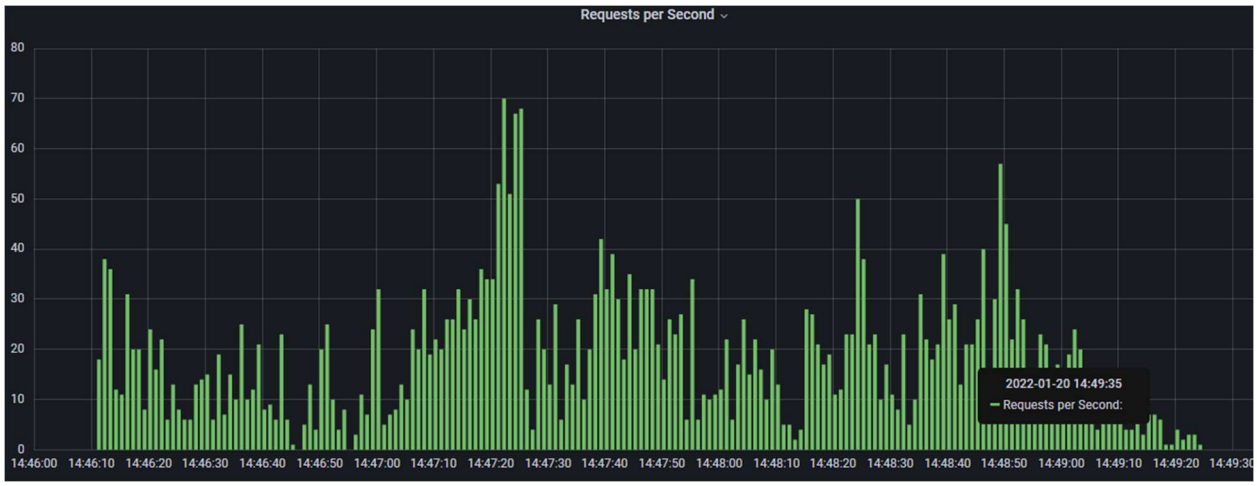
5. How would you define acceptable load for web applications?

A load is acceptable for a web application as far as the performance metrics (Response Time, Resource Usage, TPS,...) being stable during the periods of time. Of course, with capacity planning techniques, we can design our product and deployment model based on the end-users load. Also changing in the load will be acceptable if the infrastructure resource usage changes are reasonable and of course, the most important thing is the web failure rate shouldn't be ascending.

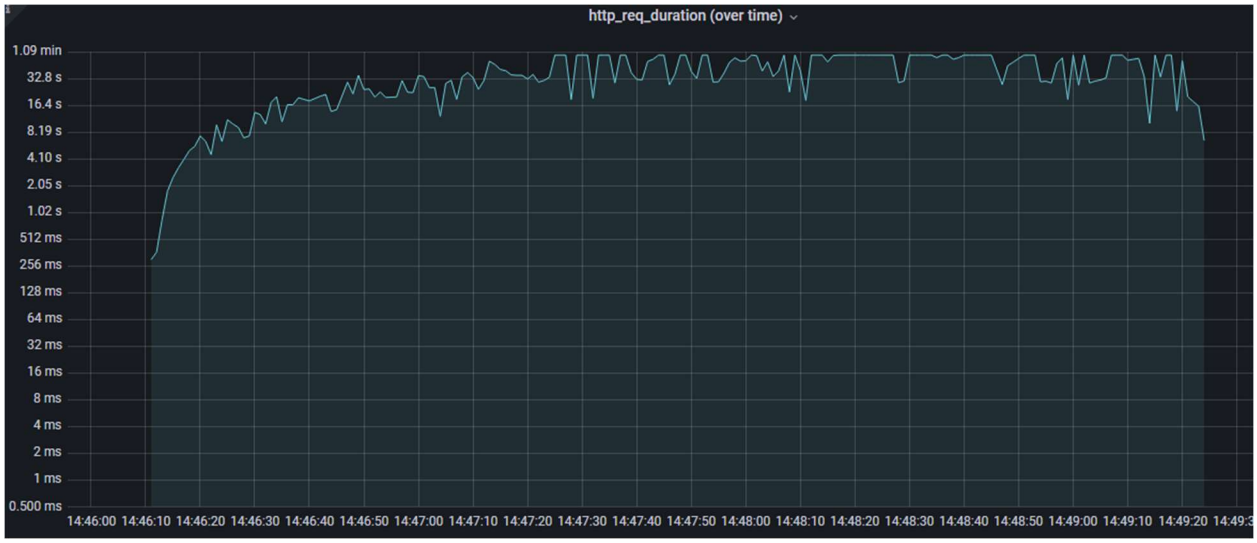
Requested Test Plan



The VU chart. Presenting user level during the test.



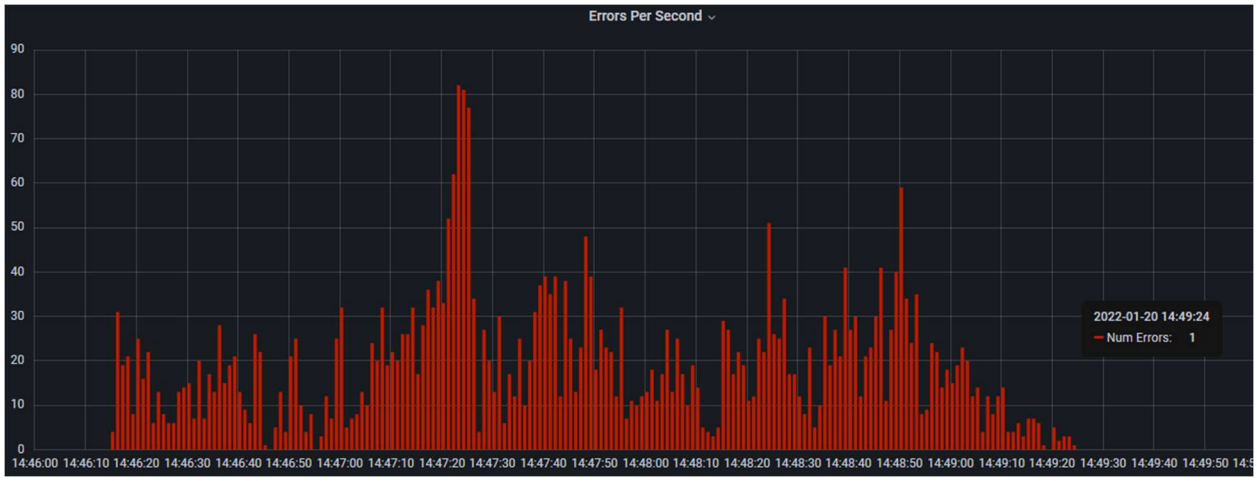
HTTP request/Sec. Presenting chaos TPS, because of banned IP, high RT, website intolerance



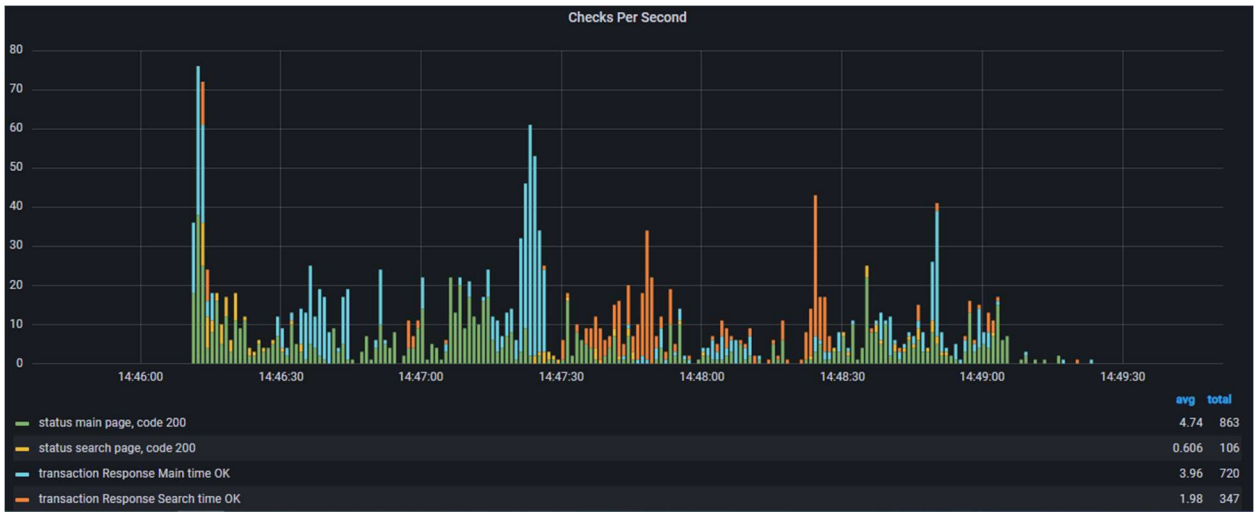
Response Time Chart. Presenting High RT. Most Of transaction get timeout.

Pass transaction chart

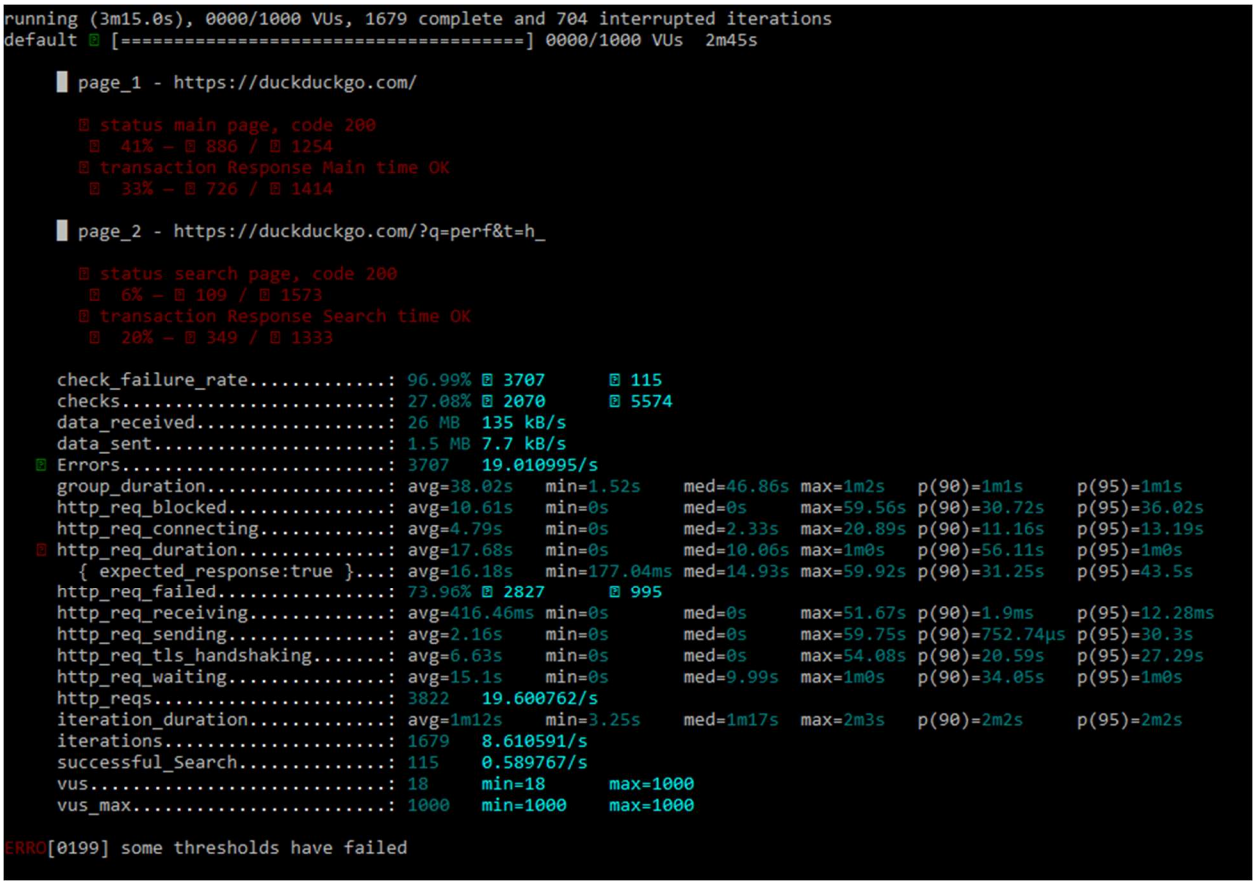
As you can see in this chart, Because of the high response time we didn’t catch pass- transactions. During the test, the site banned my IP, so the Error Rate increased during the test period.



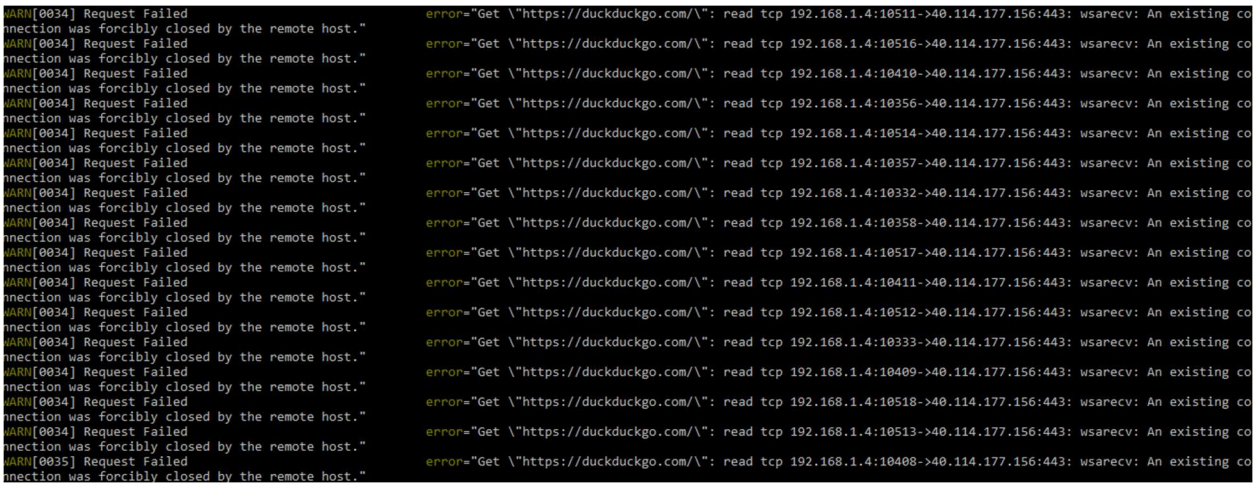
Error rate per Sec chart



Check per sec chart

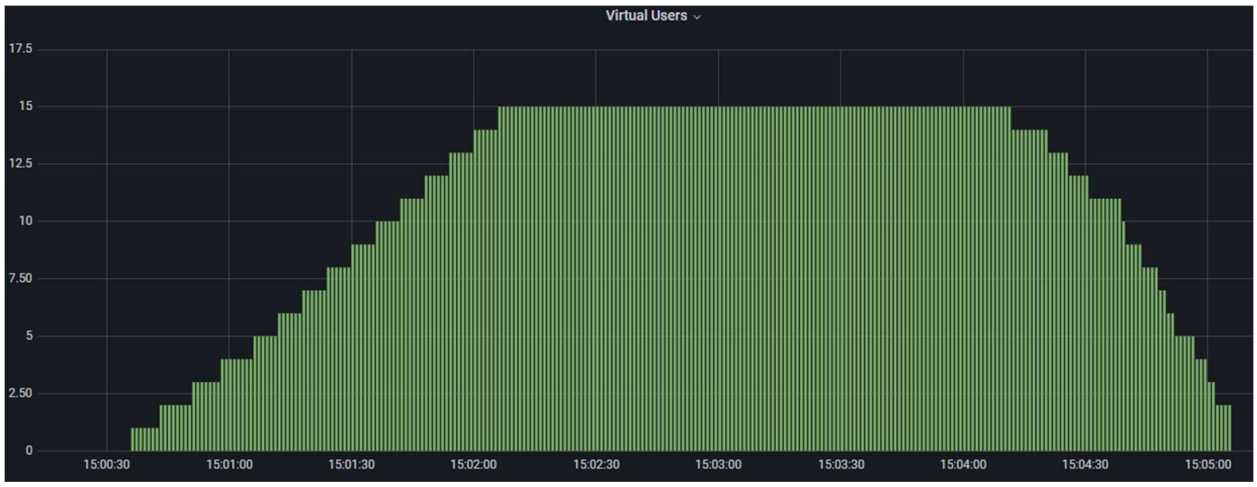


Final Report generated by k6

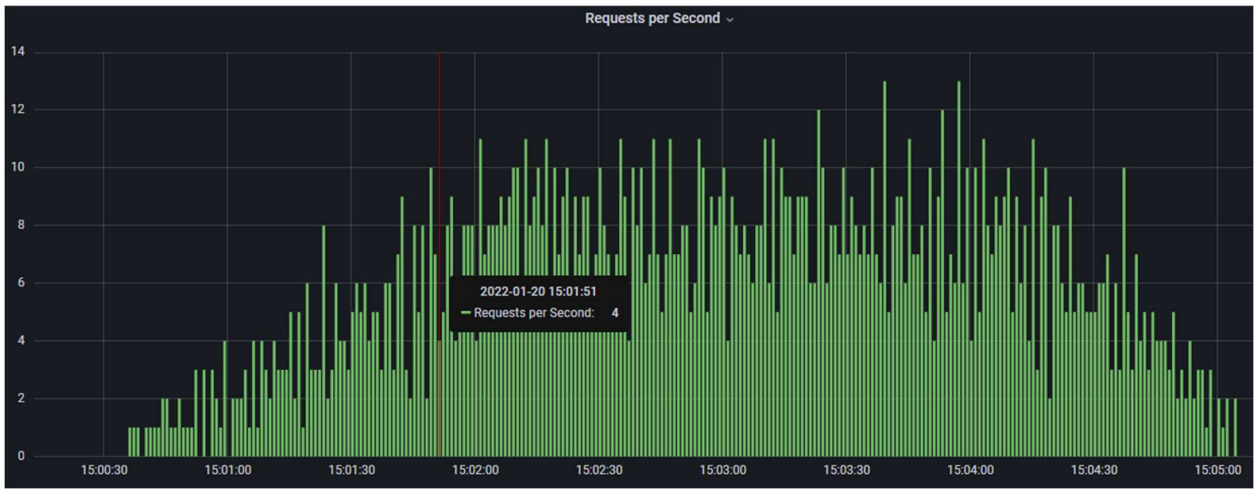


The connection closed by website host, because of high requested test level.

Baseline test



The VU chart. presenting user level during the test.

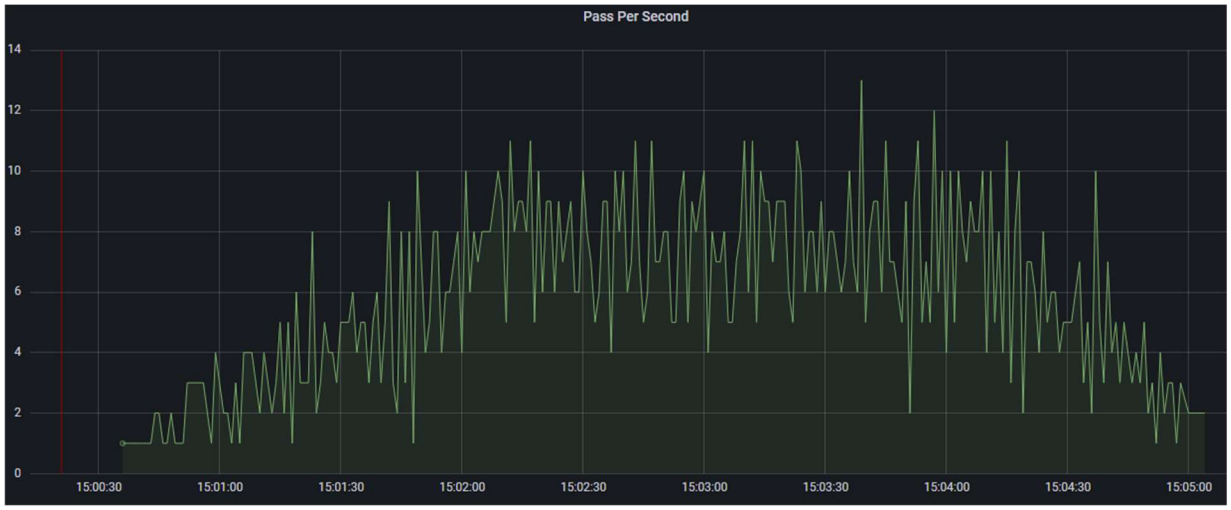


Request per sec. the baseline test up to 15 user has stable charts!

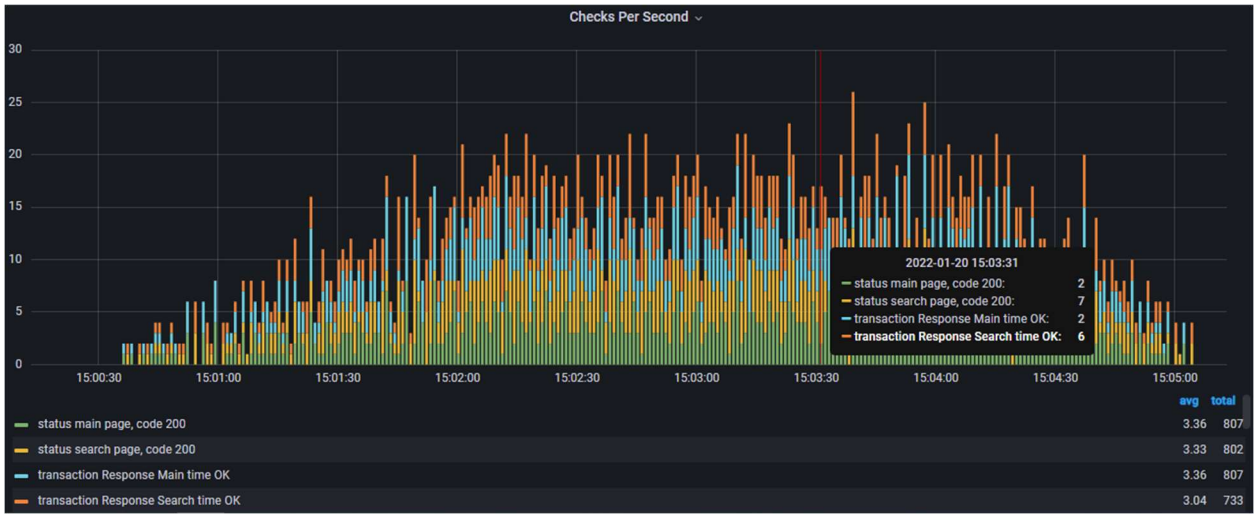
Baseline R.T. chart.

Due to former chart, I decided to increase default check response time to 2000.

Error per sec. it shows that baseline test (up to 15 users) has minimum error rate.



Pass per sec. based on former chart, most of trans pass successfully.

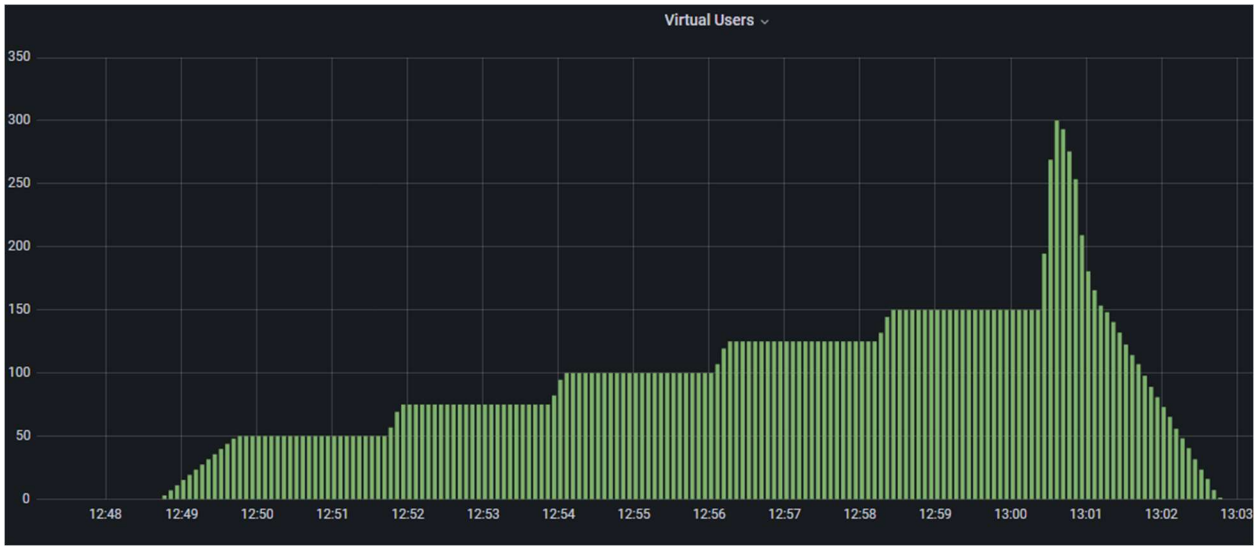


Baseline check per sec. most of the transaction get 200 status code.

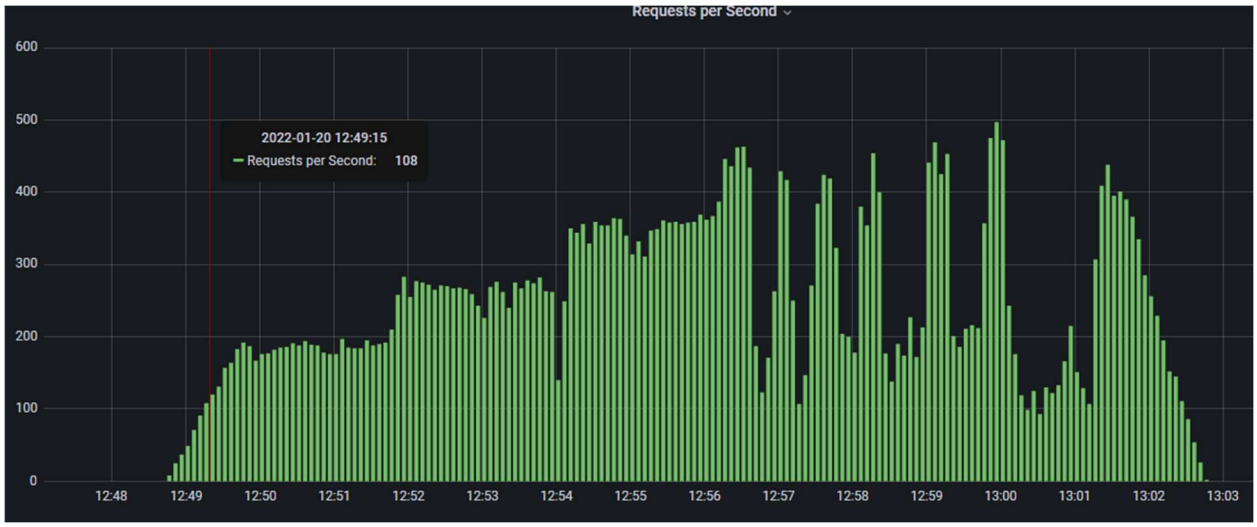
```
  scenarios: (100.00%) 1 scenario, 15 max VUs, 5m0s max duration (incl. graceful stop):
           * default: Up to 15 looping VUs for 4m30s over 6 stages (gracefulRampDown: 30s, gracefulStop: 30s)


running (4m30.9s), 00/15 VUs, 808 complete and 0 interrupted iterations
default ✓ [======================================] 00/15 VUs  4m30s

     ✓ page_1 - https://duckduckgo.com/

       ✓ status main page, code 200
       ✓ transaction Response Main time OK

     ✓ page_2 - https://duckduckgo.com/?q=perf&t=h_

       ✓ status search page, code 200
       ✗ transaction Response Search time OK
         ↳  91% — ✓ 738 / ✗ 70

     check_failure_rate.............: 4.33%  ✗ 70       ✓ 1546
     checks.........................: 97.83% ✓ 3162     ✗ 70
     data_received..................: 30 MB  110 kB/s
     data_sent......................: 167 kB 617 B/s
   ✗ Errors.........................: 70     0.258445/s
     group_duration.................: avg=1.87s    min=1.17s    med=1.76s    max=5.91s    p(90)=2.67s    p(95)=2.95s
     http_req_blocked...............: avg=5.58ms   min=0s       med=0s       max=1.61s    p(90)=0s       p(95)=0s
     http_req_connecting............: avg=1.73ms   min=0s       med=0s       max=261.59ms p(90)=0s       p(95)=0s
   ✓ http_req_duration..............: avg=865.27ms min=173.4ms  med=751.36ms max=4.89s    p(90)=1.65s    p(95)=1.94s
       { expected_response:true }...: avg=865.27ms min=173.4ms  med=751.36ms max=4.89s    p(90)=1.65s    p(95)=1.94s
     http_req_failed................: 0.00%  ✓ 0        ✗ 1616
     http_req_receiving.............: avg=247.34ms min=0s       med=10.59ms  max=4.27s    p(90)=747.09ms p(95)=1s
     http_req_sending...............: avg=469.44µs min=0s       med=531.29µs max=33.9ms   p(90)=721.25µs p(95)=843.22µs
     http_req_tls_handshaking.......: avg=3.78ms   min=0s       med=0s       max=1.35s    p(90)=0s       p(95)=0s
     http_req_waiting...............: avg=617.46ms min=164.57ms med=521.27ms max=2.85s    p(90)=1.11s    p(95)=1.43s
     http_reqs......................: 1616   5.96638/s
     iteration_duration.............: avg=3.75s    min=2.57s    med=3.64s    max=7.26s    p(90)=4.57s    p(95)=4.99s
     iterations.....................: 808    2.98319/s
     successful_Search..............: 1546   5.707935/s
     vus............................: 2      min=1      max=15
     vus_max........................: 15     min=15     max=15
```

Final Report generated by k6

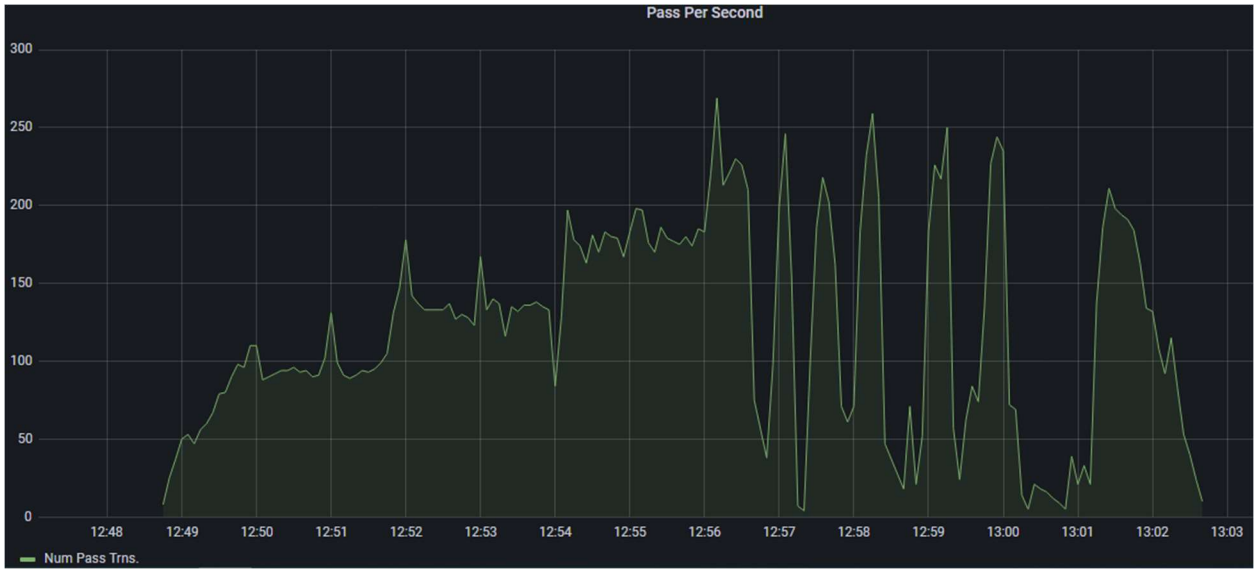
Main Test Plan



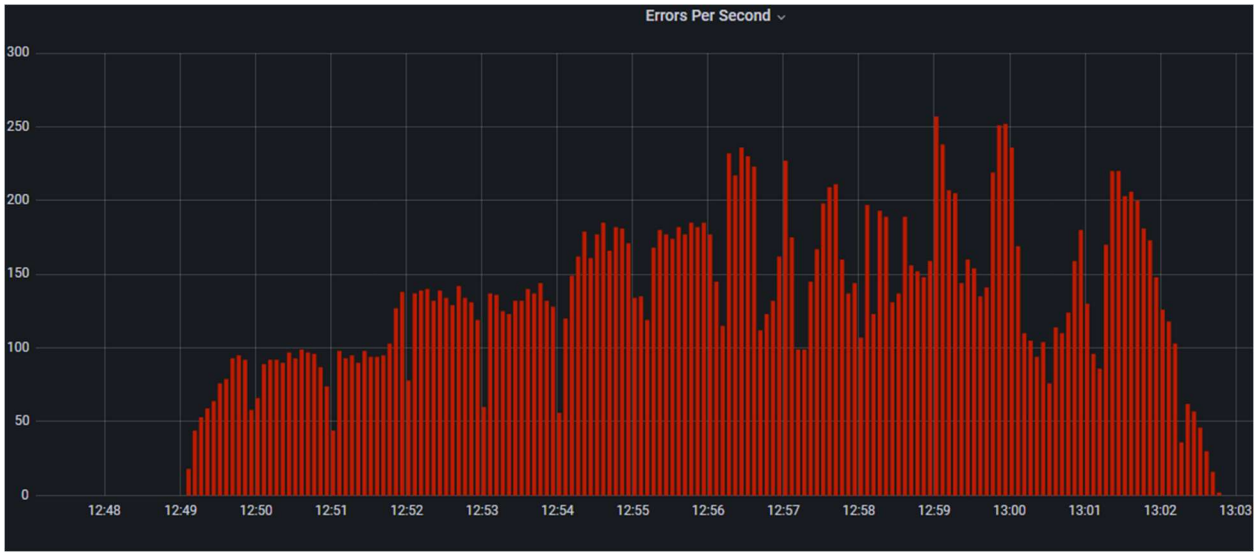
The VU chart. presenting user level during the test.



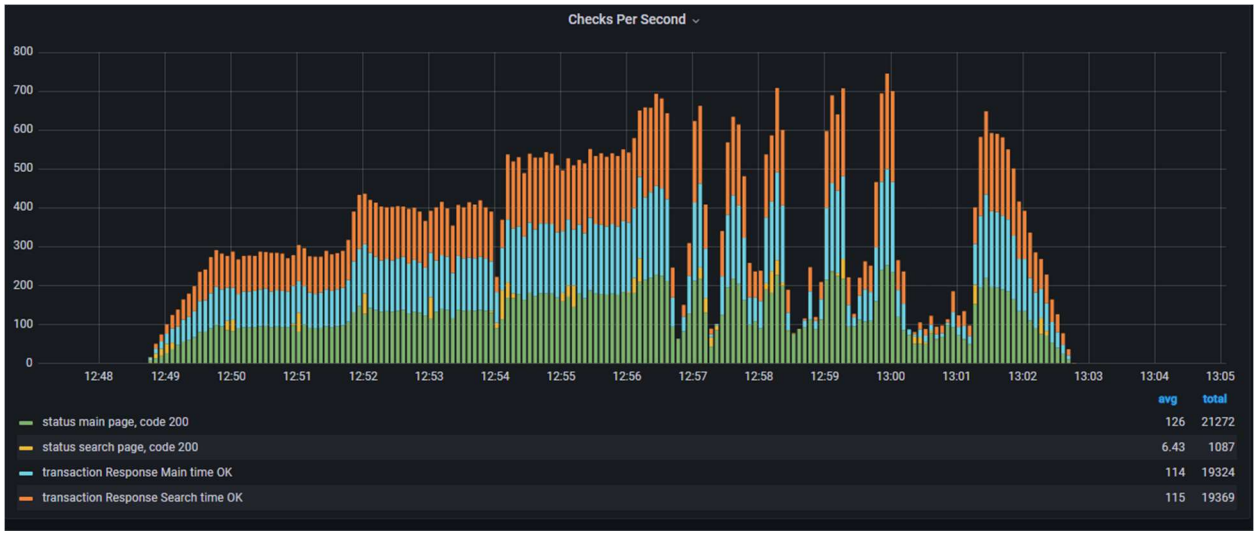
Requested R.T. it shows that after specific level the system will be un-stable.



Pass Transaction per sec. after 100 user pass tps is reduced.



Error per sec. At the end of the test (Ramp down) the test produced less error.



Check per sec

```
running (14m01.3s), 000/300 VUs, 19687 complete and 4 interrupted iterations
default ✓ [======================================] 000/300 VUs  14m0s

    ✓ page_1 - https://duckduckgo.com/

      ✓ status main page, code 200
      ✓ transaction Response Main time OK
        ✓  85% — ✓ 16770 / ✗ 2920

    ✓ page_2 - https://duckduckgo.com/?q=perf&t=h_

      ✗ status search page, code 200
        ✓  4% — ✓ 868 / ✗ 18819
      ✗ transaction Response Search time OK
        ✓  84% — ✓ 16574 / ✗ 3113

    check_failure_rate.............: 56.39% ✗ 22208     ✓ 17169
    checks.........................: 68.44% ✓ 53902     ✗ 24852
    data_received..................: 299 MB 356 kB/s
    data_sent......................: 3.1 MB 3.7 kB/s
  ✗ Errors.........................: 22208  26.396633/s
    group_duration.................: avg=2.13s    min=1.16s    med=1.4s     max=34.09s p(90)=3.69s    p(95)=5.07s
    http_req_blocked...............: avg=41.15ms  min=0s       med=0s       max=18.02s p(90)=0s       p(95)=0s
    http_req_connecting............: avg=13ms     min=0s       med=0s       max=6.52s  p(90)=0s       p(95)=0s
  ✗ http_req_duration..............: avg=1.08s    min=166.08ms med=399.67ms max=29.37s p(90)=2.67s    p(95)=4.05s
      { expected_response:true }...: avg=1.17s    min=166.08ms med=397.26ms max=28.97s p(90)=2.86s    p(95)=4.8s
    http_req_failed................: 47.79% ✗ 18819     ✓ 20558
    http_req_receiving.............: avg=34.31ms  min=0s       med=503.9µs  max=11.99s p(90)=998.2µs  p(95)=1.08ms
    http_req_sending...............: avg=297.56µs min=0s       med=0s       max=1.18s  p(90)=638.44µs p(95)=782.15µs
    http_req_tls_handshaking.......: avg=28.08ms  min=0s       med=0s       max=14.05s p(90)=0s       p(95)=0s
    http_req_waiting...............: avg=1.05s    min=165.09ms med=397.98ms max=29.37s p(90)=2.61s    p(95)=3.94s
    http_reqs......................: 39377  46.803865/s
    iteration_duration.............: avg=4.26s    min=2.36s    med=2.95s    max=50.39s p(90)=7.64s    p(95)=10.24s
    iterations.....................: 19687  23.400149/s
    successful_Search..............: 17169  20.407231/s
    vus............................: 1      min=1       max=300
    vus_max........................: 300    min=300     max=300

ERRO[0843] some thresholds have failed
```

Final Report generated by k6