CYREX

Load Testing Report

Prepared for: **Roby Weir, COO, Jigstack**
Prepared by: **Tim De Wachter, CTO**, ***Cyrex Ltd***

06/04/2021

Table of Contents

Executive summary

Cyrex was contracted by JigStack to conduct a load test to determine the performance of the Lemonade web platform on a large scale. Specifically, this test validates whether the application is ready to handle a certain volume of end users. Cyrex developed load testing scripts that simulate real user behaviour within different parts of the platform:

- Register accounts
- Activate emails
- Login with accounts
- Campaign management
  - Create campaigns
  - Browse campaigns
  - Browse transactions
- Buy tokens

Afterwards, Cyrex conducted several iterations of load testing on JigStack's infrastructure. Lastly a live demo was performed in order to demonstrate the results.

What follows is a detailed explanation of all the different tests, discovered problems and potential remediations. Most of the logs are not visible to Cyrex, thus making it hard to find the root cause of a certain bottleneck when scaling the volume of users. Nevertheless, we are eager to perform another iteration of testing in case a certain issue cannot be identified.

We are confident that the load test and this report helps the customer to raise the platform's availability and scalability to a higher level.

Conclusion

During the various iterations in the load test, different scaling issues were identified on the AWS configuration and mail server implementation. All discovered issues have been clearly communicated to JigStack and together with their technical team, logs and metrics have been analysed in order to resolve the most critical issues.

We are confident that the platform is able to handle a load of 1000 active, concurrent users without any errors or blocking problems. The average response time during this amount of load is 2,5 seconds, which is an acceptable delay during short periods of high load.

Whenever a lot of users would register at the same time, load times start to increase, up till half a minute, during the combination test. Since registration is a one-time process, it shouldn't often happen concurrently and in this way these numbers can be treated as a worst-case scenario.

The response times of the login endpoint have been drastically improved over the various iterations of the load test and it should not pose any problems.

We want to thank JigStack for putting trust in our knowhow and expertise concerning load testing.

Load Test

This section lists all the load tests that were conducted by Cyrex on JigStack's infrastructure. Any issues that arose during the load test are described and whenever applicable, charts and statistics are provided.

Whenever something in the report is unclear, Cyrex is eager to help and supply additional details. Feel free to contact us at any time.

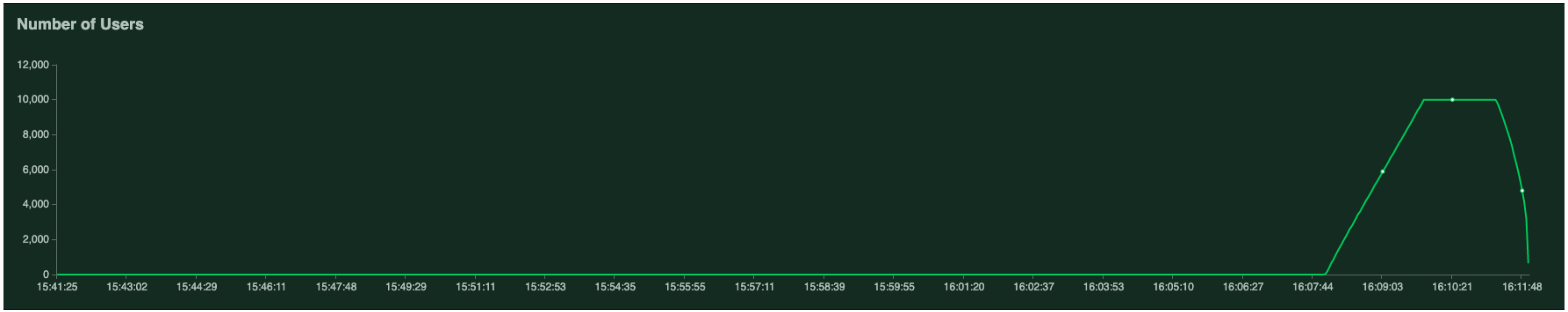
Website browsing

During this test, the bot farm sends multiple requests to fetch the static files of the website in order to test the load that the website can handle, this without performing any user-specific actions.

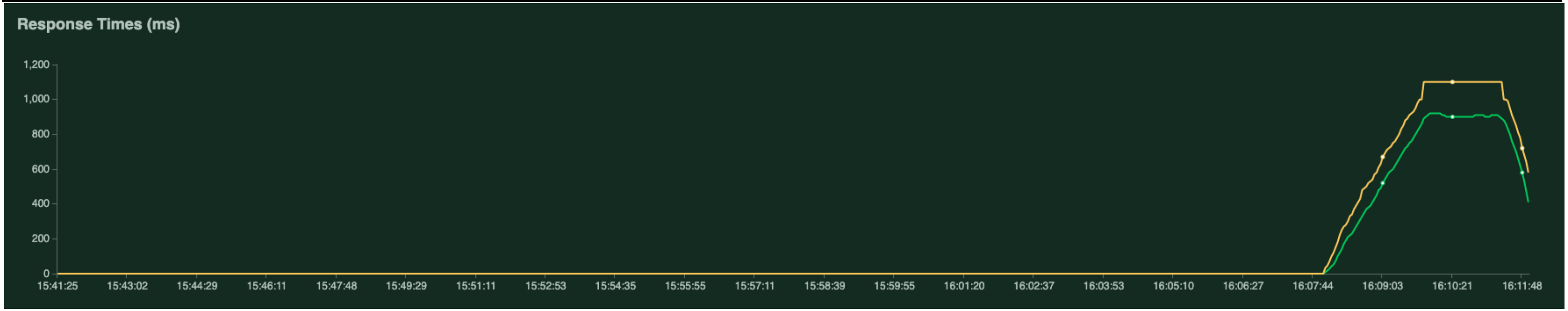
Interpreted results

No issues were discovered during this test. As you can see in the image below, the load times increase corresponding to the volume of users. Whenever the test ends, the load times stabilize. In case we have 10.000 concurrent users, we reach a response time of approximately 1 second.

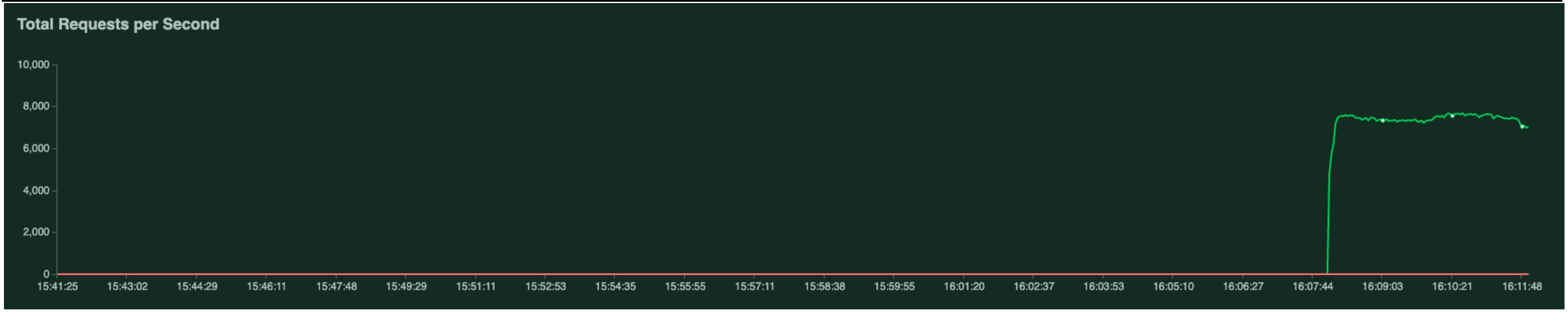
Results of test



Number of Users



Response Times (ms)



Total Requests per Second

Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	445563	0	668	5	6016	3086	1839.8	0.0
GET	/static/css/main.3bedb8ce.chunk.css	446345	0	667	5	6163	1503	1843.1	0.0
GET	/static/js/2.8d2c1ccb.chunk.js	445783	0	690	12	6646	2116495	1840.7	0.0
GET	/static/js/main.5cd86c43.chunk.js	446361	0	670	6	6659	291331	1843.1	0.0
	Aggregated	1784052	0	674	5	6659	602887	7366.8	0.0

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	740	820	870	910	970	1000	1100	6000
GET	/static/css/main.3bedb8ce.chunk.css	740	820	870	910	970	1000	1100	6200
GET	/static/js/2.8d2c1ccb.chunk.js	760	840	880	930	990	1000	1200	6600
GET	/static/js/main.5cd86c43.chunk.js	740	820	870	910	970	1000	1100	6700
	Aggregated	750	830	870	920	970	1000	1100	6700

API combination test

During this test, the bot farm acts as regular users would act, performing various actions like registration, login, requesting campaign information and buying tokens at random.

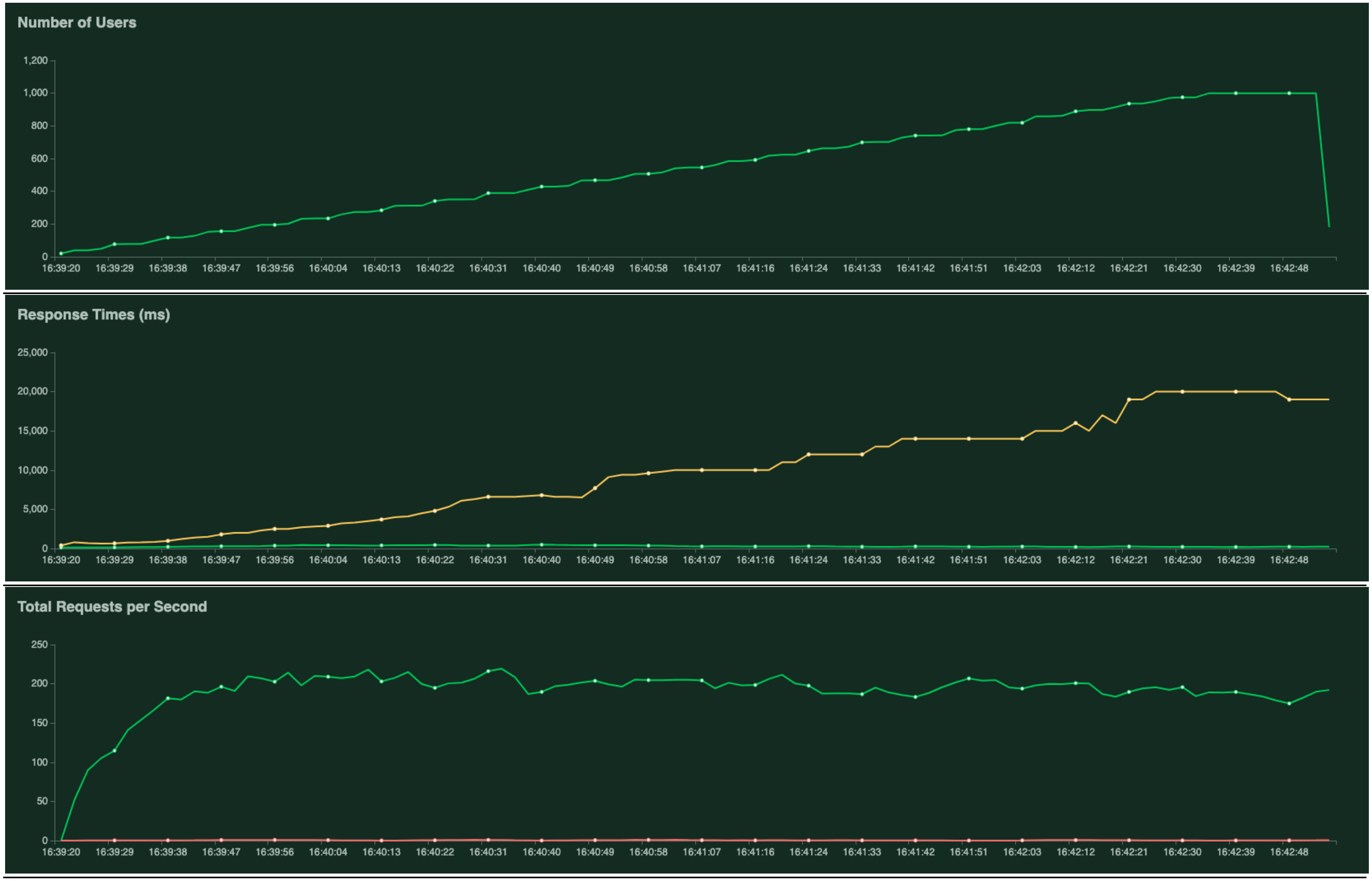
Interpreted results

No blocking issues were discovered during this test. The previously encountered 503 and 504 errors are no longer present at a load of 1000 users. A very low percentage of the requests still respond with an error (502), we did discover this as well in the individual load tests. It might be worth checking the nginx logs but could be hard to pinpoint this specific problem.

We can still clearly identify that registration is one of endpoints that take most resources and a longer time to respond, affecting the other endpoints in the long run. Since registration is a one-time action for users, the effects of this should not be drastic, but this should be taken into account upon release of the product and any time-based promotions where a lot of new users are expected.

Average response time during the test was 2,5 second, which we deem acceptable. While the maximum response time (on registration) peeked at around 30 seconds.

Results of test



Request statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/campaign-latest-active	1482	1	3863	82	20419	2019	6.9	0.0
POST	/public/check-wallet-address	34	0	2910	133	14015	126	0.2	0.0
POST	/public/jwt/verify	55	0	5041	138	19410	100	0.3	0.0
POST	/public/login	316	0	589	69	7372	297	1.5	0.0
POST	/public/register	6620	6	7453	72	29069	81	30.6	0.0
POST	/user/check-wallet-address	1447	0	2937	101	14328	125	6.7	0.0
POST	/user/jwt/verify	1446	1	4282	100	20802	100	6.7	0.0
POST	/user/login	8898	7	503	67	7541	304	41.2	0.0
POST	/user/register	2061	6	6625	81	28566	81	9.5	0.0
GET	GetAffiliateCampaign	1396	1	3750	71	20364	194	6.5	0.0
GET	GetCampaigns	9079	11	940	66	26821	1335	42.0	0.0
GET	GetTransactions	9259	10	955	66	26738	744	42.8	0.0
	Aggregated	42093	95	2537	66	29069	620	194.7	0.1

Response statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/campaign-latest-active	2100	3300	5000	7000	10000	13000	19000	20000
POST	/public/check-wallet-address	1700	2200	2700	3700	8200	14000	14000	14000
POST	/public/jwt/verify	4000	6000	7100	8400	12000	15000	19000	19000
POST	/public/login	140	170	240	460	1800	3500	5400	7400
POST	/public/register	5600	7400	9900	12000	16000	20000	27000	29000
POST	/user/check-wallet-address	1800	2700	3800	5100	7500	9200	13000	14000
POST	/user/jwt/verify	2500	3600	5500	7600	11000	14000	20000	21000
POST	/user/login	140	170	240	460	1400	2700	5100	7500
POST	/user/register	4500	6200	8400	11000	16000	19000	27000	29000
GET	GetAffiliateCampaign	1900	3100	4700	7100	10000	13000	19000	20000
GET	GetCampaigns	150	190	260	370	1500	5800	16000	27000
GET	GetTransactions	150	190	260	370	1500	6200	17000	27000
	Aggregated	290	600	1900	4100	8900	13000	20000	29000