Testowanie aplikacji

Nr przypadku testowego	Liczba użytkowników	Spawn rate	Host
1	5	1	http://127.0.0.1:8000/
2	50	1	http://127.0.0.1:8000/
3	500	1	http://127.0.0.1:8000/
4	1000	1	http://127.0.0.1:8000/
5	5000	1	http://127.0.0.1:8000/

Testowana była aplikacja internetowa main.py, która wyświetla napis „Hello, world!” w oknie przeglądarki po połączeniu się z hostem. W ramach testów skupiono się na sprawdzeniu wydajności serwera aplikacji pod obciążeniem. Testy zostały przeprowadzone na lokalnym serwerze odpalonym z poziomu komputera z procesorem AMD Athlon 200GE 3.20Ghz, 8GB pamięci RAM oraz systemem operacyjnym Windows 10 Pro. Do testów wykorzystano Locust w wersji 2.15.1. Pod uwagę brane były: ilość użytkowników, czas odpowiedzi serwera i liczba żądań na sekundę.

1. Pierwszy przypadek testowy:

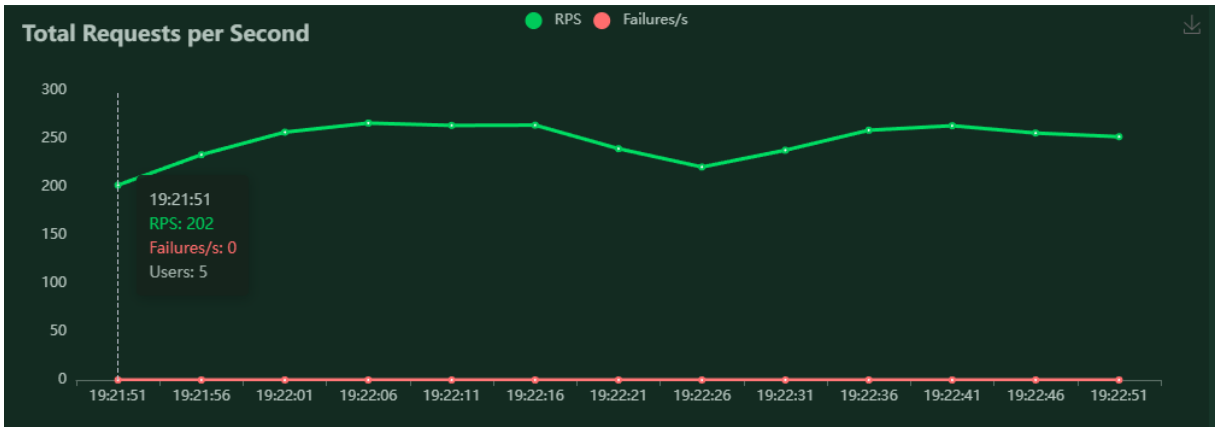
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 5. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 30ms a 40ms. Liczba żądań na sekundę wahała się między 200ms a 250ms.

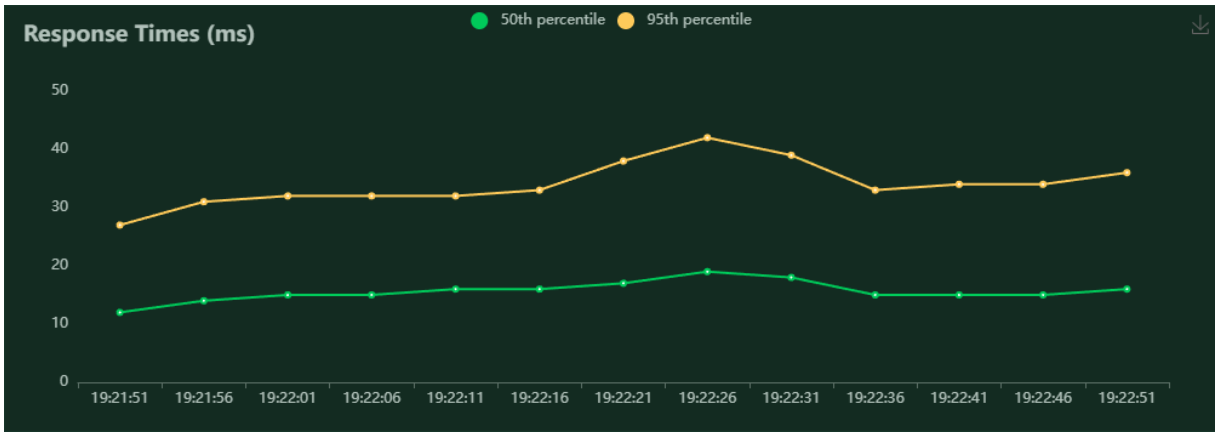
Request Statistics

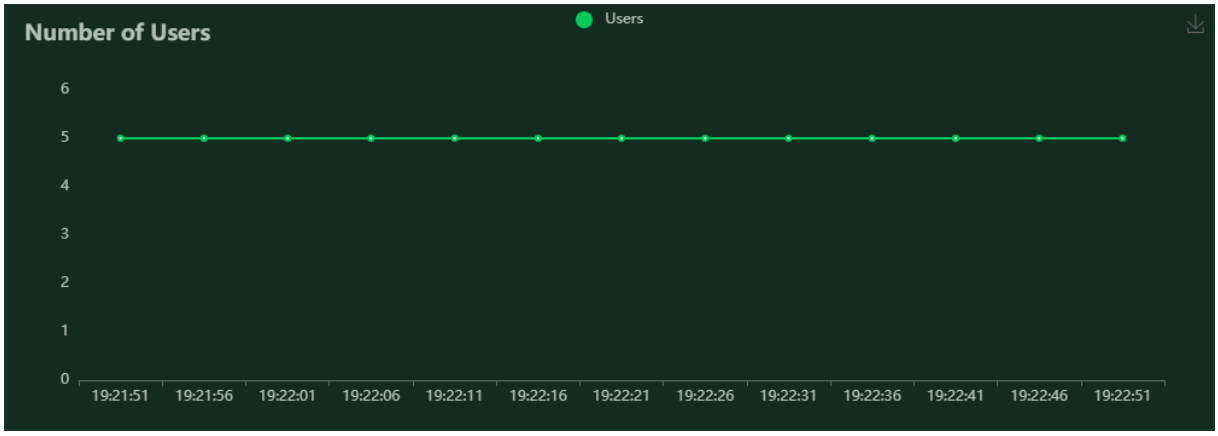
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	5464	0	27	6	109	13	81.7	0.0
GET	//hello	5686	0	14	3	82	269	85.1	0.0
POST	//hello	5685	0	14	3	61	14	85.0	0.0
	Aggregated	**16835**	**0**	**18**	**3**	**109**	**99**	**251.8**	**0.0**

Response Time 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	//	26	27	29	32	37	41	51	110
GET	//hello	14	14	15	17	19	21	28	83
POST	//hello	14	14	15	17	19	21	28	62
	Aggregated	**16**	**18**	**23**	**26**	**29**	**34**	**44**	**110**




Response Times (ms)
50th percentile
95th percentile
50
40
30
20
10
0
19:21:51
19:21:56
19:22:01
19:22:06
19:22:11
19:22:16
19:22:21
19:22:26
19:22:31
19:22:36
19:22:41
19:22:46
19:22:51


Number of Users
Users
6
5
4
3
2
1
0
19:21:51
19:21:56
19:22:01
19:22:06
19:22:11
19:22:16
19:22:21
19:22:26
19:22:31
19:22:36
19:22:41
19:22:46
19:22:51

2. Drugi przypadek testowy:

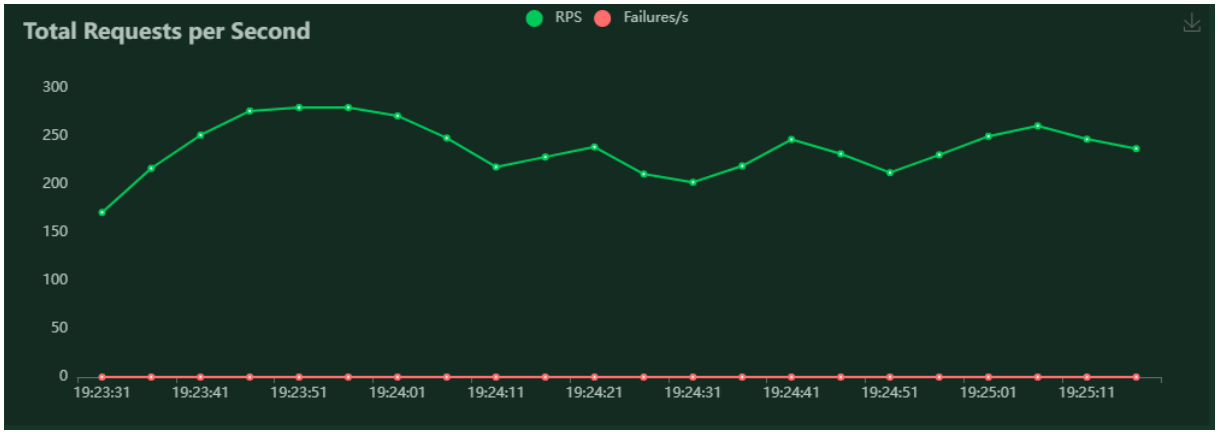
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 50. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 340ms a 430ms. Liczba żądań na sekundę wahała się między 171ms a 280ms.

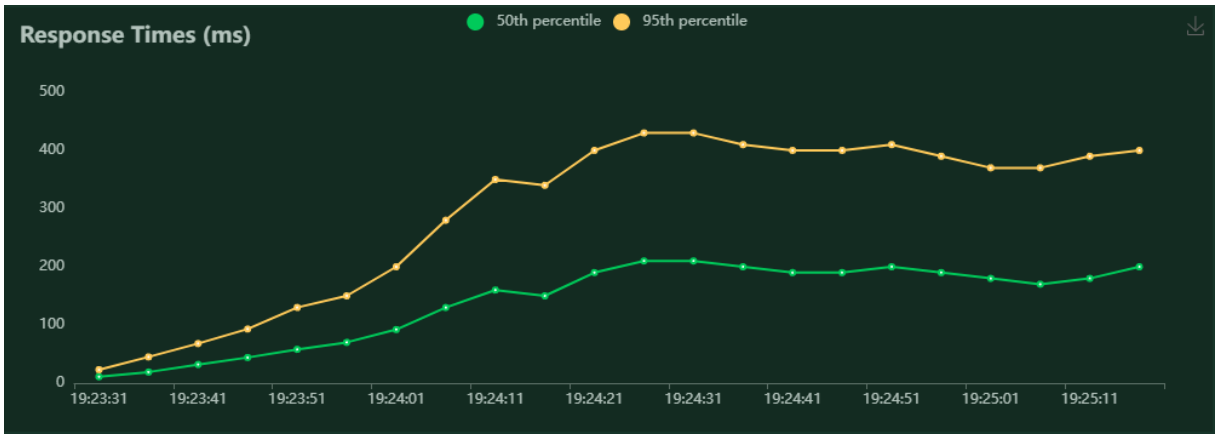
Request Statistics

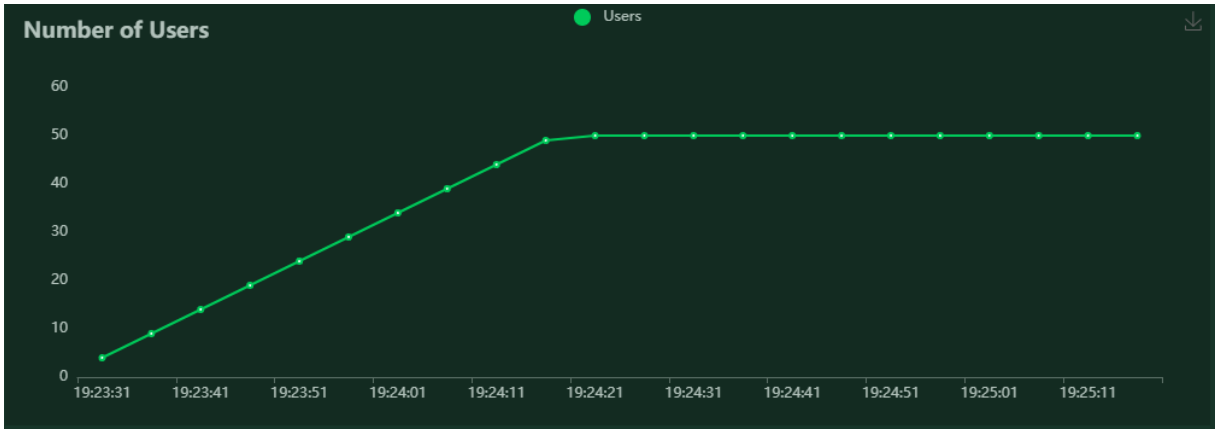
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	9001	0	234	6	662	13	81.1	0.0
GET	//hello	8803	0	123	3	431	269	79.3	0.0
POST	//hello	8789	0	113	3	453	14	79.2	0.0
	Aggregated	**26593**	**0**	**157**	**3**	**662**	**98**	**239.6**	**0.0**

Response Time 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	//	240	260	300	360	400	420	520	660
GET	//hello	130	130	150	190	220	230	280	430
POST	//hello	110	120	140	170	200	210	260	450
	Aggregated	**130**	**160**	**200**	**230**	**300**	**380**	**440**	**660**




Response Times (ms)
50th percentile
95th percentile
500
400
300
200
100
0
19:23:31
19:23:41
19:23:51
19:24:01
19:24:11
19:24:21
19:24:31
19:24:41
19:24:51
19:25:01
19:25:11


Number of Users
Users
60
50
40
30
20
10
0
19:23:31
19:23:41
19:23:51
19:24:01
19:24:11
19:24:21
19:24:31
19:24:41
19:24:51
19:25:01
19:25:11

3. Trzeci przypadek testowy:

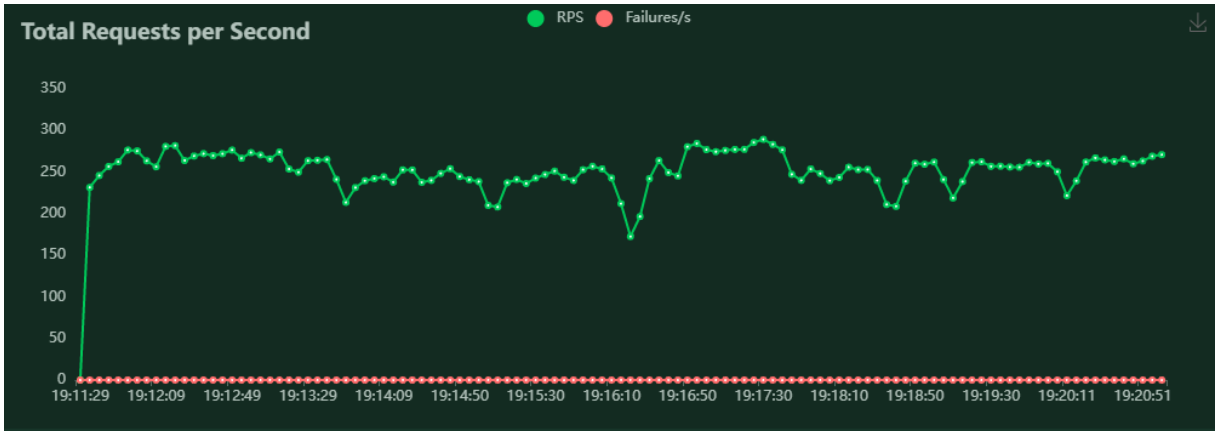
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 500. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 3000ms a 3700ms. Liczba żądań na sekundę wahała się między 172ms a 289ms.

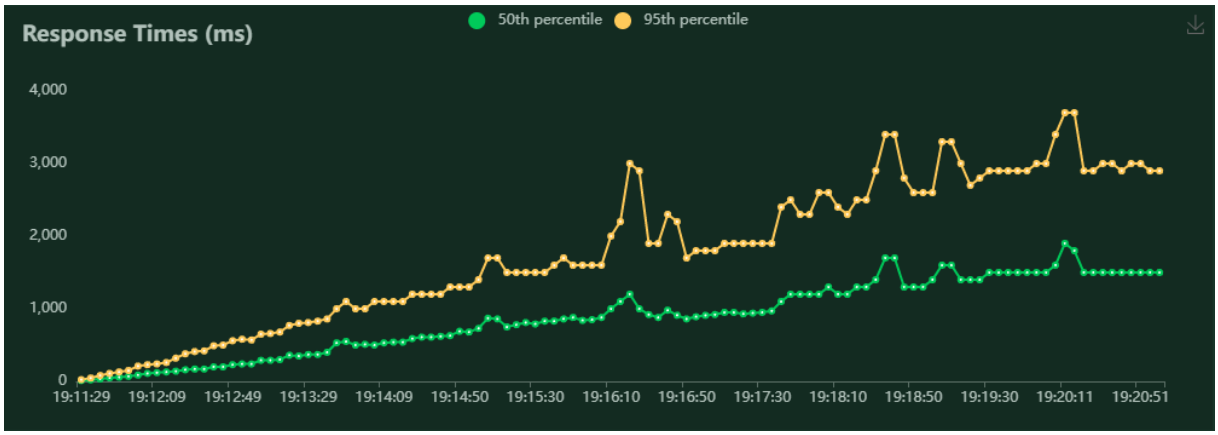
Request Statistics

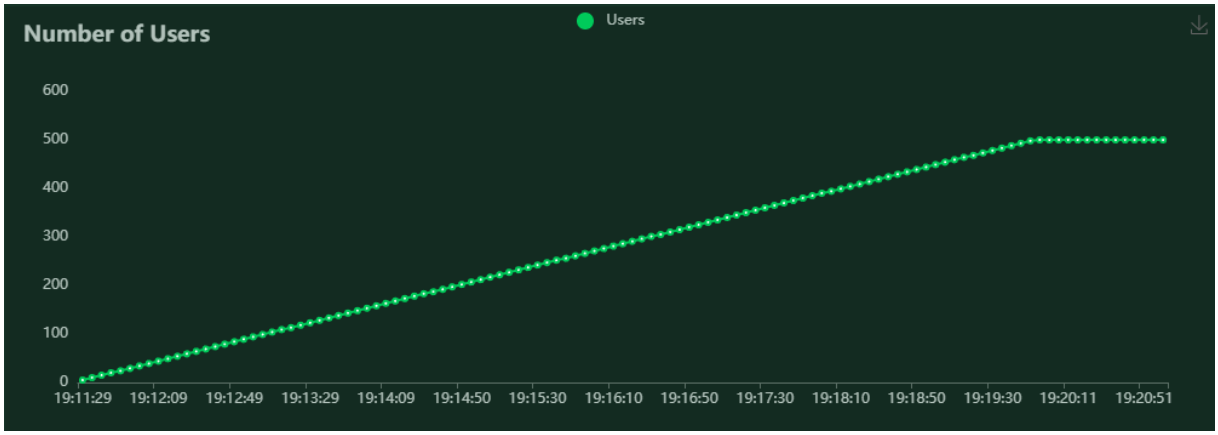
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	48970	0	1656	13	4311	13	84.7	0.0
GET	//hello	48789	0	832	5	3342	269	84.4	0.0
POST	//hello	48666	0	825	7	3194	14	84.2	0.0
	Aggregated	**146425**	**0**	**1105**	**5**	**4311**	**98**	**253.4**	**0.0**

Response Time 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	//	1700	1900	2400	2700	2800	3000	3400	4300
GET	//hello	860	950	1200	1300	1500	1500	1800	3300
POST	//hello	850	950	1200	1300	1400	1500	1800	3200
	Aggregated	**970**	**1200**	**1400**	**1600**	**2400**	**2800**	**3100**	**4300**




Response Times (ms)
50th percentile
95th percentile
4,000
3,000
2,000
1,000
0
19:11:29
19:12:09
19:12:49
19:13:29
19:14:09
19:14:50
19:15:30
19:16:10
19:16:50
19:17:30
19:18:10
19:18:50
19:19:30
19:20:11
19:20:51


Number of Users
Users
600
500
400
300
200
100
0
19:11:29
19:12:09
19:12:49
19:13:29
19:14:09
19:14:50
19:15:30
19:16:10
19:16:50
19:17:30
19:18:10
19:18:50
19:19:30
19:20:11
19:20:51

4. Czwarty przypadek testowy:

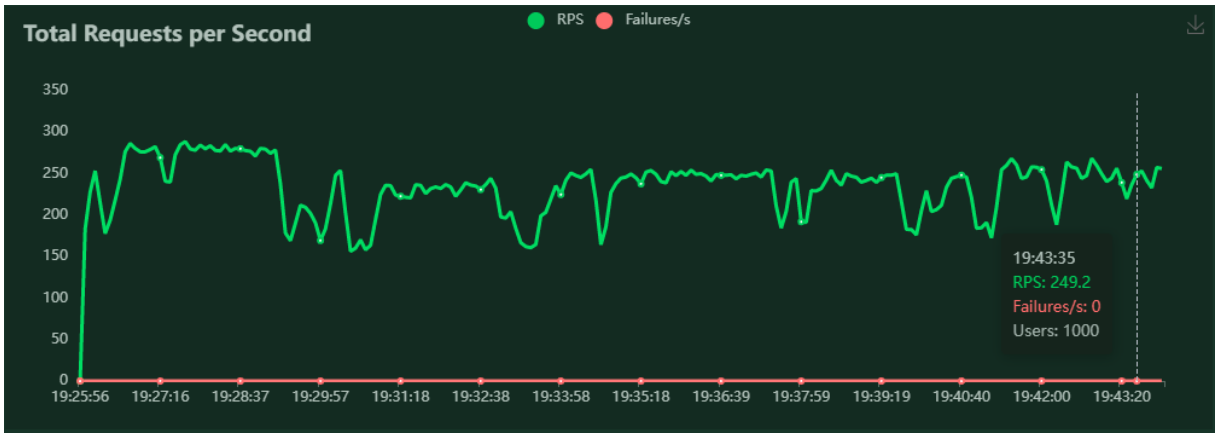
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 1000. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 5800ms a 6900ms. Liczba żądań na sekundę wahała się między 156ms a 286ms.

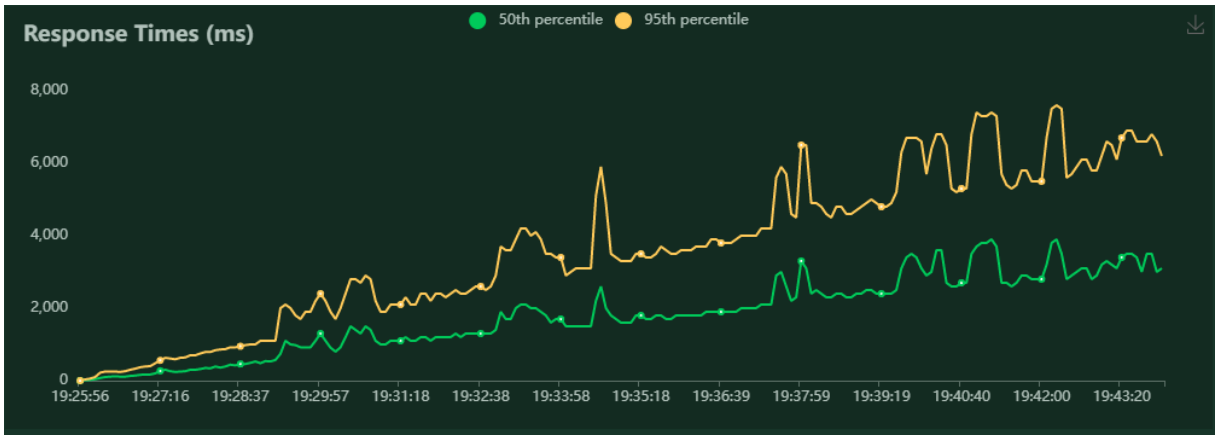
Request Statistics

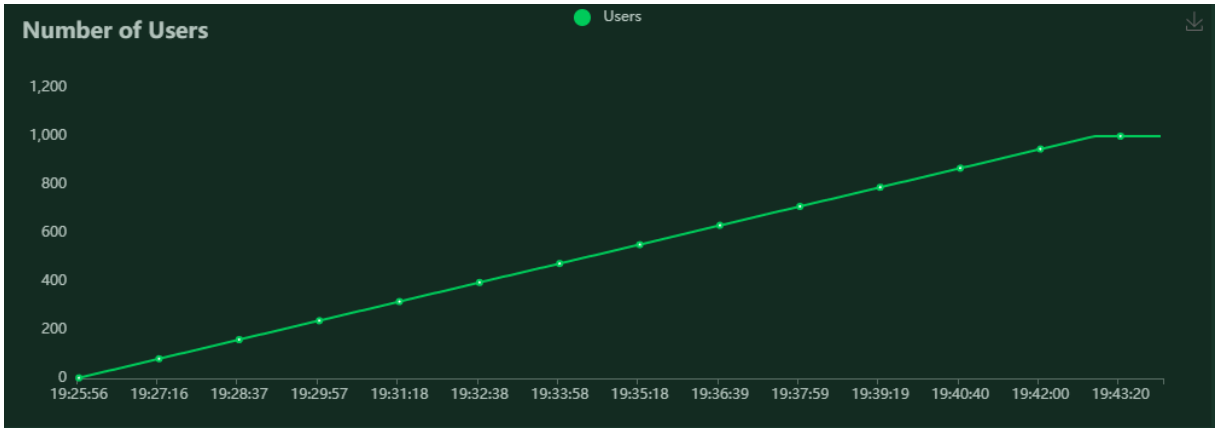
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	85214	0	3370	6	8519	13	78.3	0.0
GET	//hello	85209	0	1691	3	5635	269	78.3	0.0
POST	//hello	84970	0	1685	3	5533	14	78.1	0.0
	Aggregated	**255393**	**0**	**2249**	**3**	**8519**	**98**	**234.7**	**0.0**

Response Time 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	//	3400	3900	4700	5400	6000	6600	7300	8500
GET	//hello	1700	2000	2300	2700	3000	3300	3800	5600
POST	//hello	1700	2000	2300	2700	3000	3300	3800	5500
	Aggregated	**2000**	**2400**	**2800**	**3400**	**4700**	**5600**	**6800**	**8500**




Response Times (ms)
50th percentile
95th percentile
8,000
6,000
4,000
2,000
0
19:25:56
19:27:16
19:28:37
19:29:57
19:31:18
19:32:38
19:33:58
19:35:18
19:36:39
19:37:59
19:39:19
19:40:40
19:42:00
19:43:20


Number of Users
Users
1,200
1,000
800
600
400
200
0
19:25:56
19:27:16
19:28:37
19:29:57
19:31:18
19:32:38
19:33:58
19:35:18
19:36:39
19:37:59
19:39:19
19:40:40
19:42:00
19:43:20

5. Piąty przypadek:

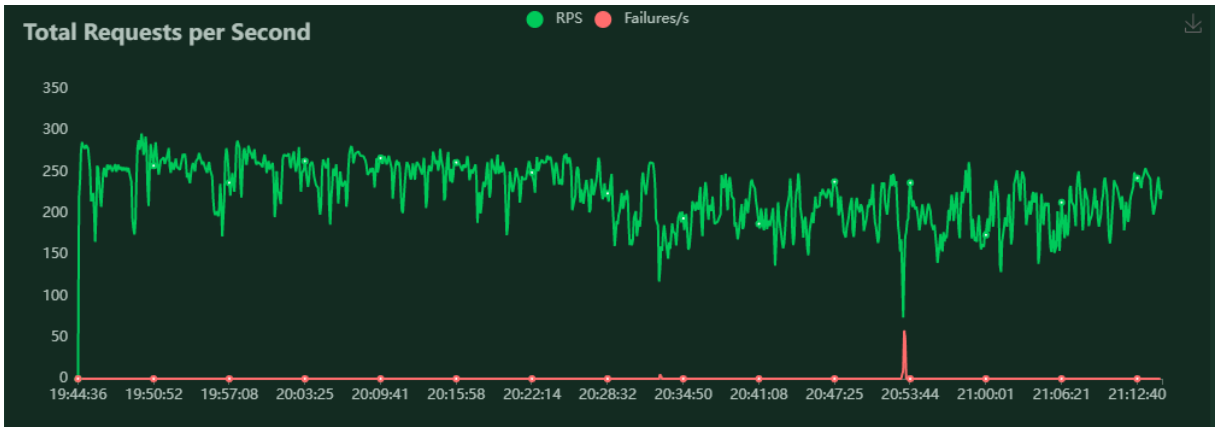
Testy trwały przez czas odpowiedni do osiągnięcia połączenia z wymaganą ilością użytkowników + 1 minuta, a maksymalna liczba użytkowników równocześnie korzystających z aplikacji wynosiła 5000. Wykresy dostępne na stronie Locust przedstawiają między innymi liczbę żądań na sekundę, czas odpowiedzi serwera, liczbę błędów oraz liczbę użytkowników. W trakcie testów nie wystąpiły żadne błędy, a czas odpowiedzi serwera po osiągnięciu maksymalnej ilości użytkowników wahał się między 30000ms a 39000ms. Liczba żądań na sekundę wahała się między 160ms (z pominięciem błędu, podczas którego najniższa wartość wyniosła 74ms) a 296ms.

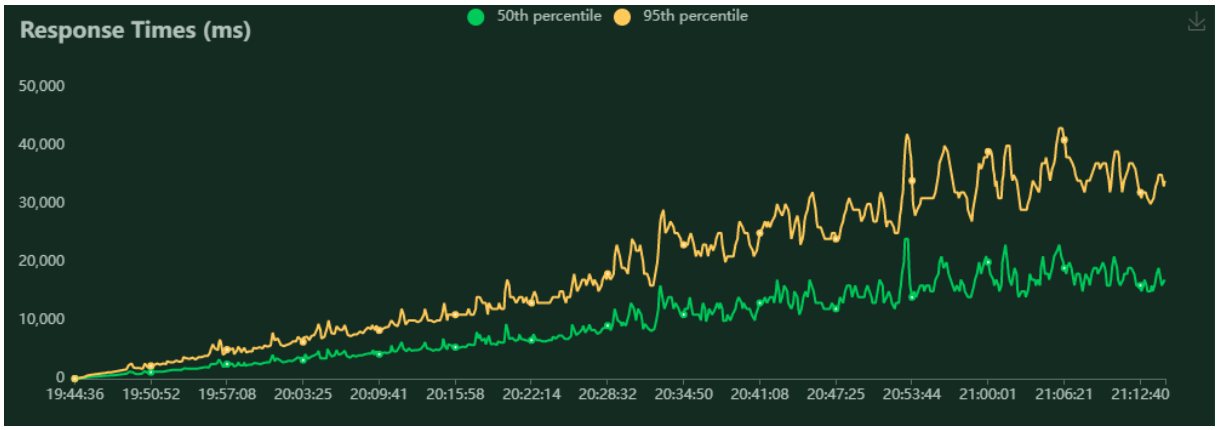
Request Statistics

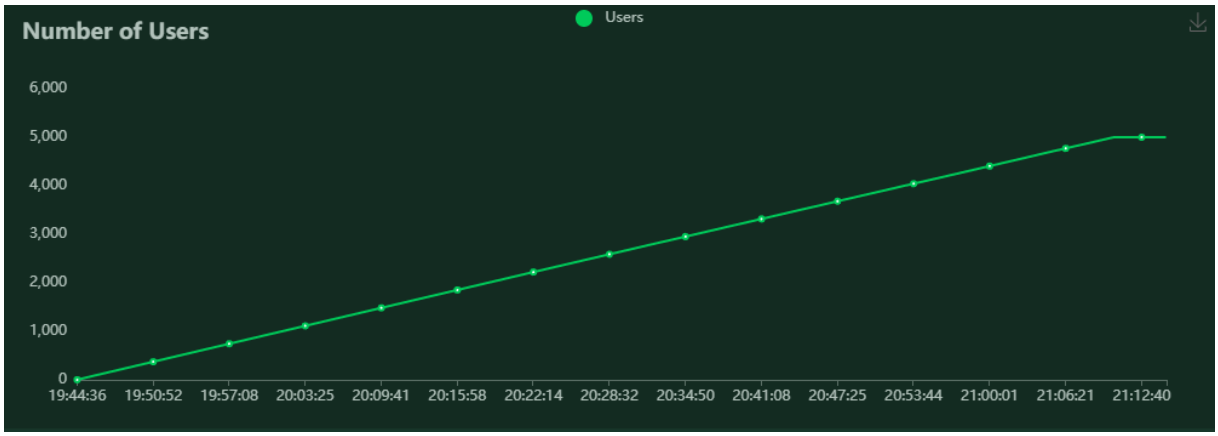
Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	//	406043	318	17411	6	43838	12	75.1	0.1
GET	//hello	406618	156	8729	3	26111	268	75.2	0.0
POST	//hello	405384	163	8718	3	26102	13	75.0	0.0
	Aggregated	**1218045**	**637**	**11620**	**3**	**43838**	**98**	**225.4**	**0.1**

Response Time 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	//	14000	22000	26000	30000	34000	37000	40000	44000
GET	//hello	7300	11000	13000	15000	17000	18000	21000	26000
POST	//hello	7300	11000	13000	15000	17000	18000	21000	26000
	Aggregated	**9900**	**13000**	**15000**	**18000**	**26000**	**32000**	**38000**	**44000**







Failures Statistics

Method	Name	Error	Occurrences
GET	//	[Errno 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	318
GET	//hello	[Errno 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	156
POST	//hello	[Errno 10061] [WinError 10061] Nie można nawiązać połączenia, ponieważ komputer docelowy aktywnie go odmawia.	163

Wnioski

Poszczególne testy różniły się ilością połączonych użytkowników, liczbą żądań na sekundę oraz czasem odpowiedzi serwera. Liczba żądań i czas odpowiedzi rósł proporcjonalnie do ilości użytkowników. Żaden test nie zgłosił błędu (wyjątkiem był piąty przypadek testowania, wydaje mi się, że błędy wyskoczyły przez chwilowy „lag” komputera). Aplikacja jest wydajna.