

Testowanie aplikacji z Locust

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/

Został przeprowadzony test na aplikacji internetowej main.py, która wyświetla/wita użytkownika napisem „Hello World” w przeglądarce internetowej. Podczas testów była sprawdzana wydajność serwera aplikacji podczas jego obciążenia. Testy były przeprowadzane na lokalnym serwerze z komputera który posiadał procesor Intel Core i5 9th 2.40 GHz, 16Gb pamięci operacyjnej oraz zainstalowany system Windows 10 Pro. W testach został wykorzystany Locust w wersji 2.15.1. a pod uwagę były brane takie dane jak: czas odpowiedzi serwera, ilość użytkowników oraz liczba żądań na sekundę.

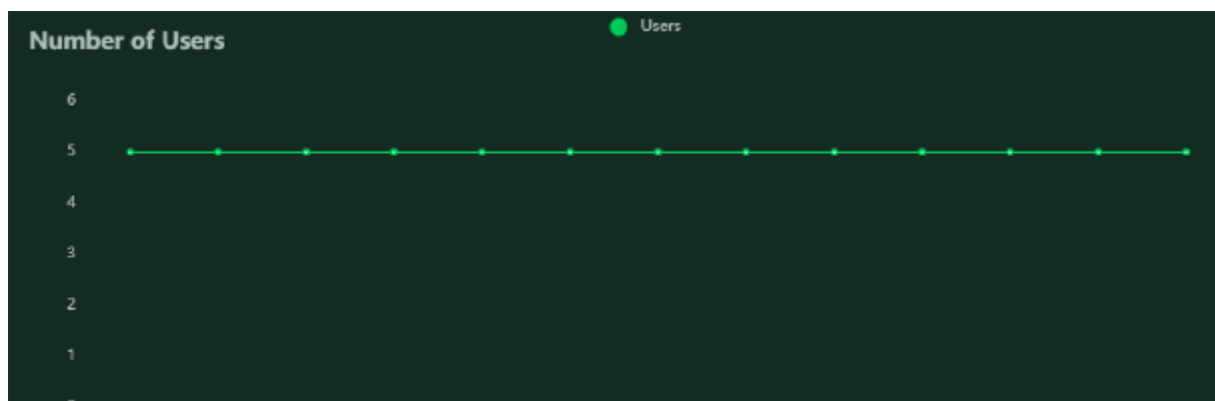
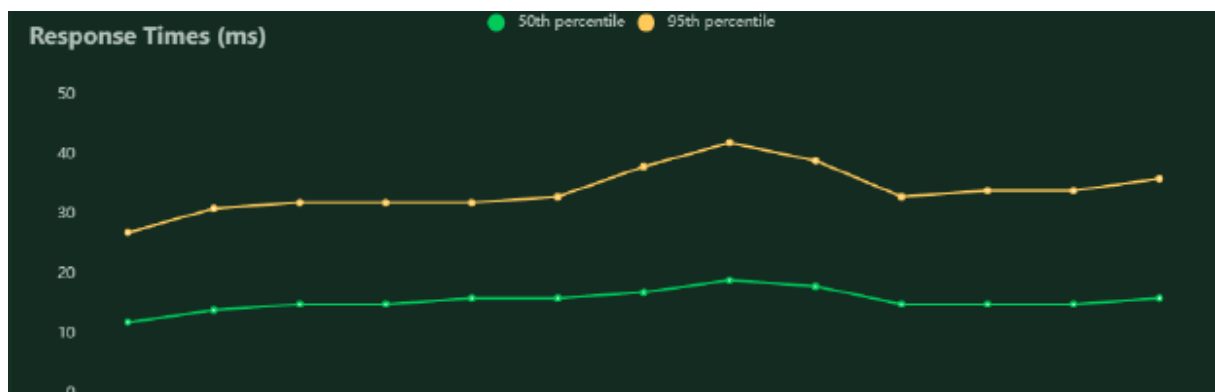
1. Pierwszy test

Pierwszy przypadek testowy trwał do osiągnięcia połączenia z wymaganą ilością użytkowników, maksymalna liczba użytkowników wynosiła pięć. Poniższe wykresy przedstawiają czas odpowiedzi serwera, liczbę żądań na sekundę oraz liczbę błędów.

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	5685	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



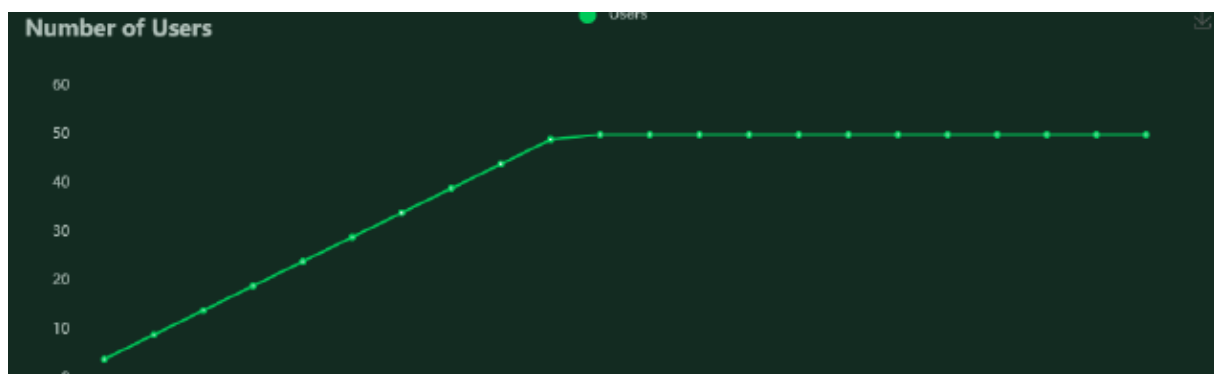
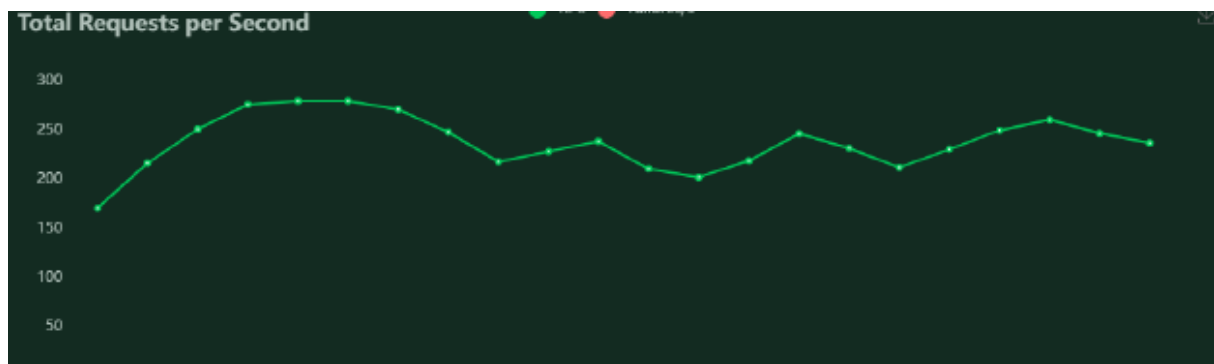
2. Drugi test

Drugi przypadek testowy trwał do osiągnięcia połączenia z wymaganą ilością użytkowników, maksymalna liczba użytkowników wynosiła 50. W trakcie testu nie wystąpiły żadne błędy, a wykresy przedstawiają takie parametry jak: czas odpowiedzi serwera, liczbę żądań na sekundę, liczbę błędów oraz liczbę użytkowników.

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

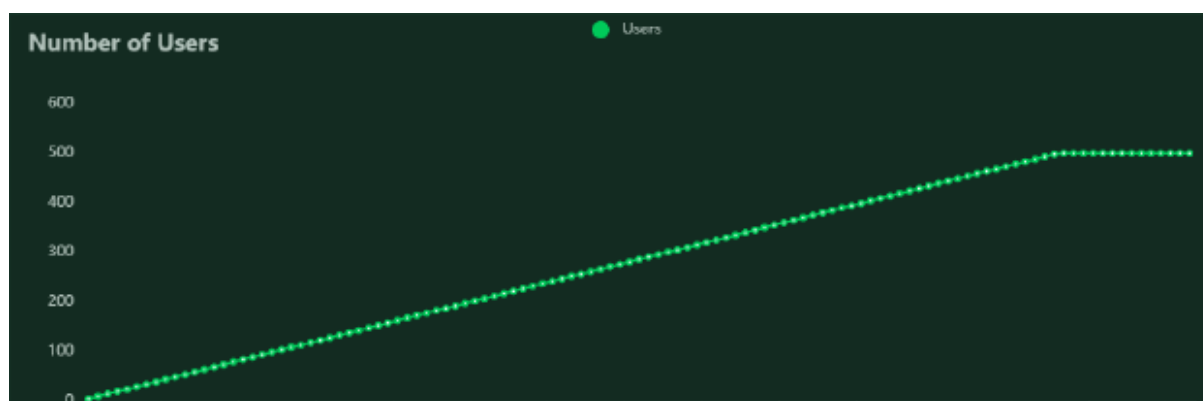
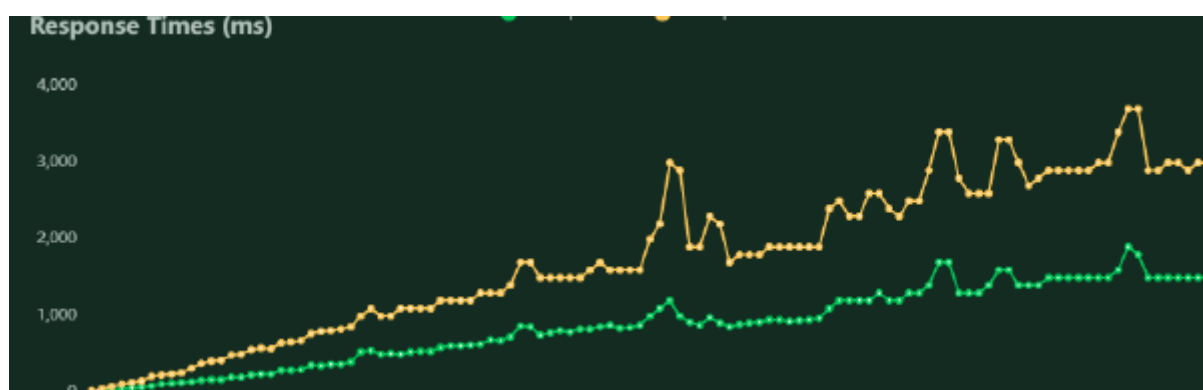


3. Trzeci test

Trzeci przypadek testowy trwał do osiągnięcia połączenia z wymaganą ilością użytkowników, maksymalna liczba użytkowników wynosiła 500. W trakcie testu nie wystąpiły żadne błędy, a wykresy przedstawiają takie parametry jak: czas odpowiedzi serwera, liczbę żądań na sekundę, liczbę błędów oraz liczbę użytkowników.

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	46666	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	850	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

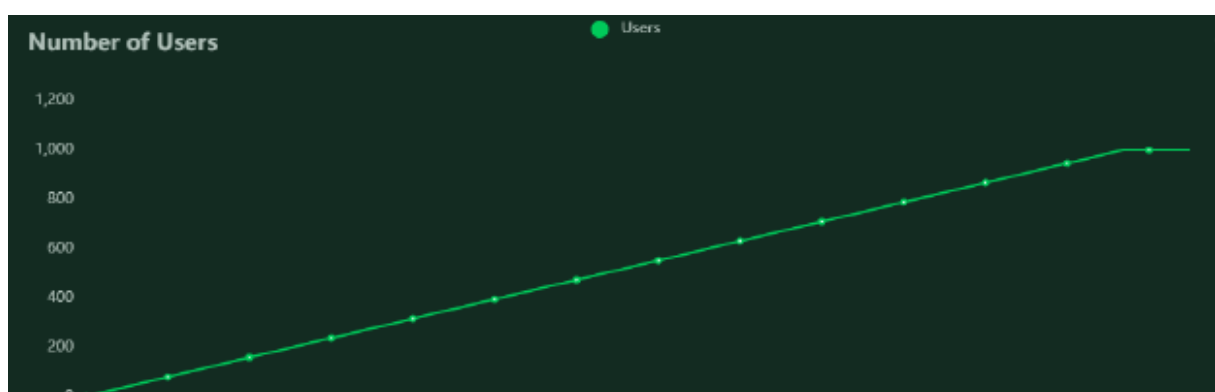
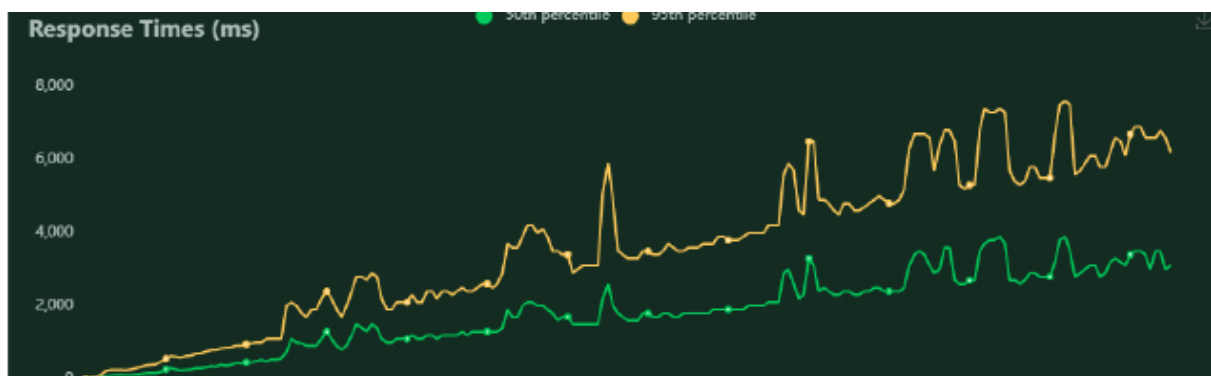


4. Czwarty test

Czwarty przypadek testowy trwał do osiągnięcia połączenia z wymaganą ilością użytkowników, maksymalna liczba użytkowników wynosiła 1000. W trakcie testu nie wystąpiły żadne błędy, a wykresy przedstawiają takie parametry jak: czas odpowiedzi serwera, liczbę żądań na sekundę, liczbę błędów oraz liczbę użytkowników.

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)	98%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

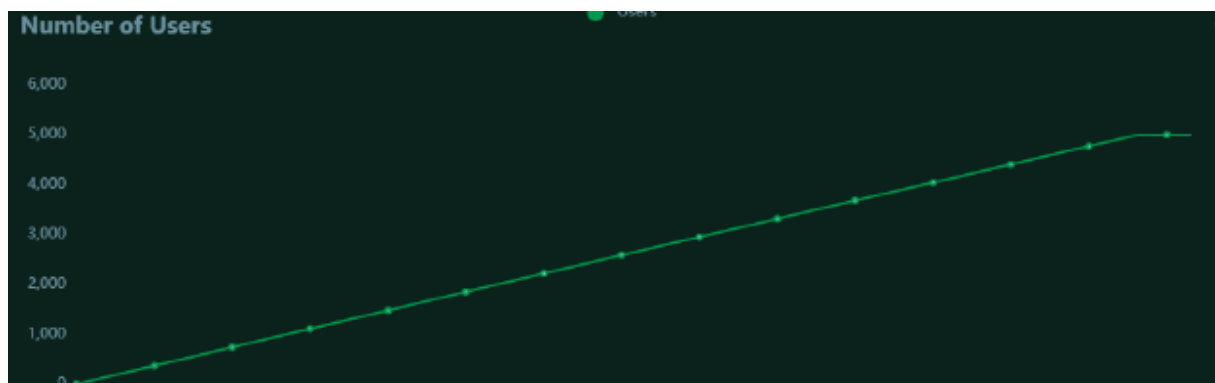
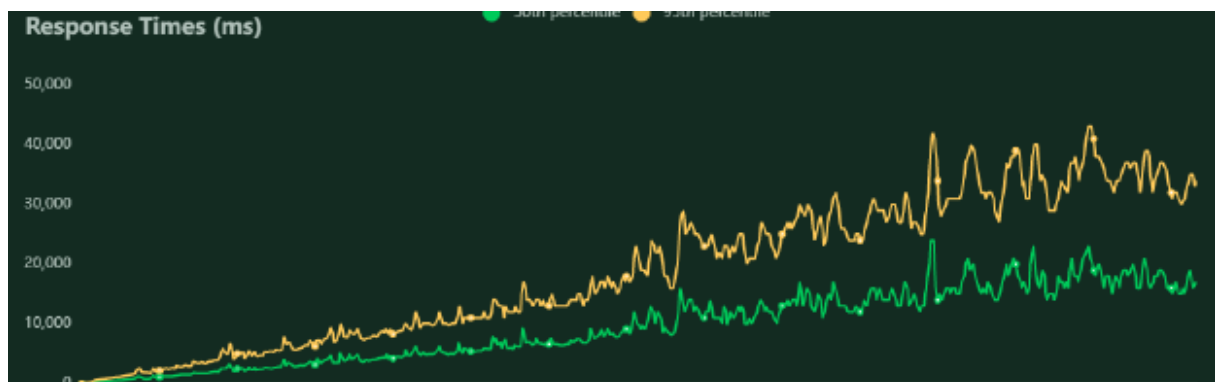


5. Piąty test

Piąty przypadek testowy trwał do osiągnięcia połączenia z wymaganą ilością użytkowników, maksymalna liczba użytkowników wynosiła 5000. W trakcie testu nie wystąpiły żadne błędy, a wykresy przedstawiają takie parametry jak: czas odpowiedzi serwera, liczbę żądań na sekundę, liczbę błędów oraz liczbę użytkowników.

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



Wnioski z przeprowadzonych testów

Przeprowadzone pięć testów różniły się od siebie liczbą żądań na sekundę, czasem odpowiedzi serwera oraz liczbą użytkowników, a czas odpowiedzi i żądania rosły proporcjonalnie do ilości użytkowników. Żaden z powyższych testów nie zgłosił błędu co oznacza że testowana aplikacja jest jak najbardziej wydajna.