

# JMeter - sprawozdanie

1 maja 2020

## Spis treści

<b>1</b>	<b>Plany testów</b>	<b>3</b>
1.1	Stan gry w pamięci podręcznej . . . . .	3
1.2	Stan gry w pliku tekstowym . . . . .	3
1.3	Sposób prezentacji danych . . . . .	4
<b>2</b>	<b>Testy</b>	<b>4</b>
2.1	Stan gry w pamięci podręcznej . . . . .	4
2.1.1	Test 1 . . . . .	5
2.1.1.1	GET - /game/view . . . . .	5
2.1.1.2	POST - /game/move . . . . .	6
2.1.2	Test 2 . . . . .	7
2.1.2.1	GET - /game/view . . . . .	8
2.1.2.2	POST - /game/move . . . . .	8
2.1.3	Test 3 . . . . .	9
2.1.3.1	GET - /game/view . . . . .	10
2.1.3.2	POST - /game/move . . . . .	10
2.1.4	Test 4 . . . . .	11
2.1.4.1	GET - /game/view . . . . .	12
2.1.4.2	POST - /game/move . . . . .	12
2.1.5	Test 5 . . . . .	13
2.1.5.1	GET - /game/view . . . . .	14
2.1.5.2	POST - /game/move . . . . .	14
2.1.6	Test 6 . . . . .	15
2.1.6.1	GET - /game/view . . . . .	16
2.1.6.2	POST - /game/move . . . . .	16
2.1.7	Dodatkowy test . . . . .	17
2.2	Stan gry w pliku tekstowym . . . . .	18
2.2.1	Test 1 . . . . .	18
2.2.1.1	GET - /game/view . . . . .	18
2.2.1.2	POST - /game/move . . . . .	19
2.2.2	Test 2 . . . . .	20
2.2.2.1	GET - /game/view . . . . .	21
2.2.2.2	POST - /game/move . . . . .	21
2.2.3	Test 3 . . . . .	22
2.2.3.1	GET - /game/view . . . . .	23
2.2.3.2	POST - /game/move . . . . .	23
2.2.4	Test 4 . . . . .	24
2.2.4.1	GET - /game/view . . . . .	25
2.2.4.2	POST - /game/move . . . . .	25
2.2.5	Test 5 . . . . .	26
2.2.5.1	GET - /game/view . . . . .	27
2.2.5.2	POST - /game/move . . . . .	27
2.2.6	Test 6 . . . . .	28
2.2.6.1	GET - /game/view . . . . .	29
2.2.6.2	POST - /game/move . . . . .	29

2.2.7	Dodatkowy test . . . . .	30
2.2.7.1	GET - <i>/game/view</i> . . . . .	31
2.2.7.2	POST - <i>/game/move</i> . . . . .	31
<b>3</b>	<b>Analiza wyników</b> . . . . .	<b>33</b>
3.1	Wnioski . . . . .	34

# 1 Plany testów

W aplikacji webowej zostały zdefiniowane dwa end-point'y odpowiadające za pobranie widoku stanu gry (*/game/view*, metoda GET) oraz za wysłanie na serwer żądania ruchu (*/game/move*, metoda POST).

Testy zostały zaplanowane w taki sposób, że zostaną równocześnie uruchomione dwie grupy wątków (z danymi parametrami). Dodatkowo dla end-point'u */game/move* miał zostać utworzony dodatkowy plik konfiguracyjny *csv* z przykładowymi ruchami, lecz niemającymi większego sensu ani nieoddającymi logiki gry, gdyż nie miało to znaczenia. Plik miał wyglądać w ten sposób:

```
1 4:5:z;14:1:t;5:13:b;2:1:x;2:1:m;13:10:y
2 0:7:a;11:1:a;2:13:h;5:6:u;6:5:r;12:11:c
3 7:4:g;12:5:e;5:14:h;9:6:l;3:5:g;11:12:i
4 1:0:g;12:5:w;9:10:x;5:7:n;9:9:r;10:13:q
5 9:1:t;14:0:y;1:13:n;2:3:y;0:7:o;11:10:p
6 ...
```

Sposób zdefiniowania ruchów został opisany w **README** poprzedniego projektu (tak jak zostało wspomniane ruchy z pliku nie muszą mieć większego sensu). Plik został automatycznie wygenerowany (po zdefiniowaniu odpowiedniego wyrażenia regularnego) na **tej** stronie i jest umieszczony pod ścieżką:

*rlis\_jmeter\scrabble\game\resources\example\_moves.csv*

Testy zostały podzielone na dwa główne etapy - testy aplikacji ze stanem gry przechowywanym w pamięci podręcznej oraz testy aplikacji ze stanem gry przechowywanym w pliku *txt*.

## 1.1 Stan gry w pamięci podręcznej

Zasadniczo dwie główne fazy testów nie mają przebiec tak samo w celu porównania obu sposobów, więc niemalże cały opis testów znajdzie się w tej podsekcji.

W pierwszej kolejności zostaną utworzone dwie grupy wątków (odpowiadające za wybrane end-point'y) wraz z wybranymi parametrami przypisanymi do danego testu (dla obu grup parametry będą te same). Dwie grupy wątków zostały utworzone w tym celu, żeby uzyskać osobne wykresy dla każdego z end-point'ów. Czyli w rezultacie działających wątków (użytkowników) będzie dwa razy więcej niż w wierszu **Liczba użytkowników**.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
<b>Liczba użytkowników</b>	5	15	30	5	15	30
<b>Uruchom w ciągu (sekund)</b>	2	2	2	4	4	4
<b>Liczba powtórzeń</b>	50	50	50	50	50	50

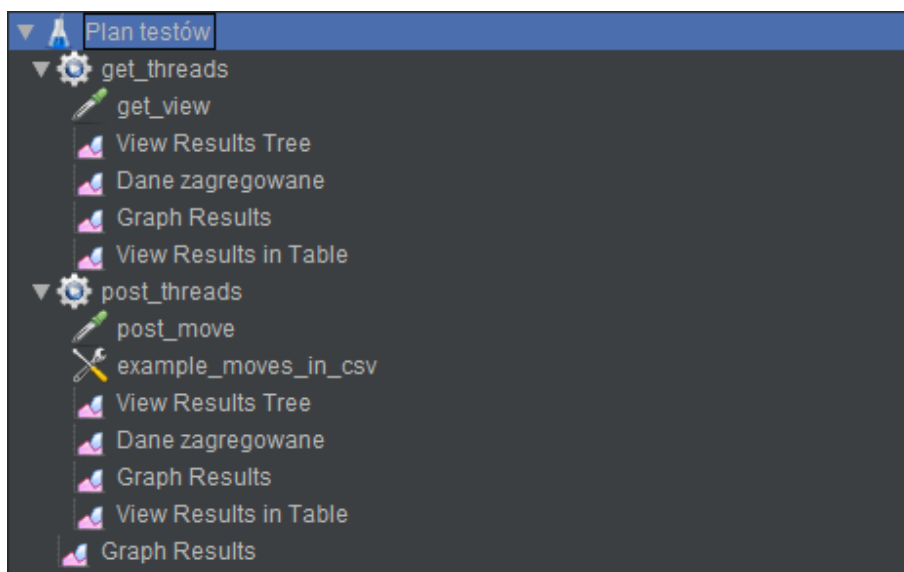
Liczba użytkowników reprezentuje liczbę wirtualnych użytkowników wykonujących skrypt. Drugi parametr oznacza ile czasu ma zabrać osiągnięcie pełnej liczby wątków (użytkowników), zaś trzeci to liczba powtórzeń skryptu dla każdego z wątków.

## 1.2 Stan gry w pliku tekstowym

Testy dla aplikacji przechowywującej stan gry w pliku tekstowym mają mieć taki sam przebieg jak w przypadku wcześniejszego sposobu.

### 1.3 Sposób prezentacji danych

W celu prezentacji danych miało zostać utworzonych kilku słuchaczy i struktura planu testów miała wyglądać w ten sposób:



Rysunek 1: Struktura planu testów

Osobne wyniki w różnych formach (tabela, drzewo, wykres, zagregowane dane) dla każdej grupy wątków oraz jeden wspólny wykres dla obu grup. Dodatkowo widać plik konfiguracyjny dla grupy *post\_threads* z przykładowymi ruchami.

## 2 Testy

Po wykonanej fazie planowania zostały rozpoczęte właściwe testy.

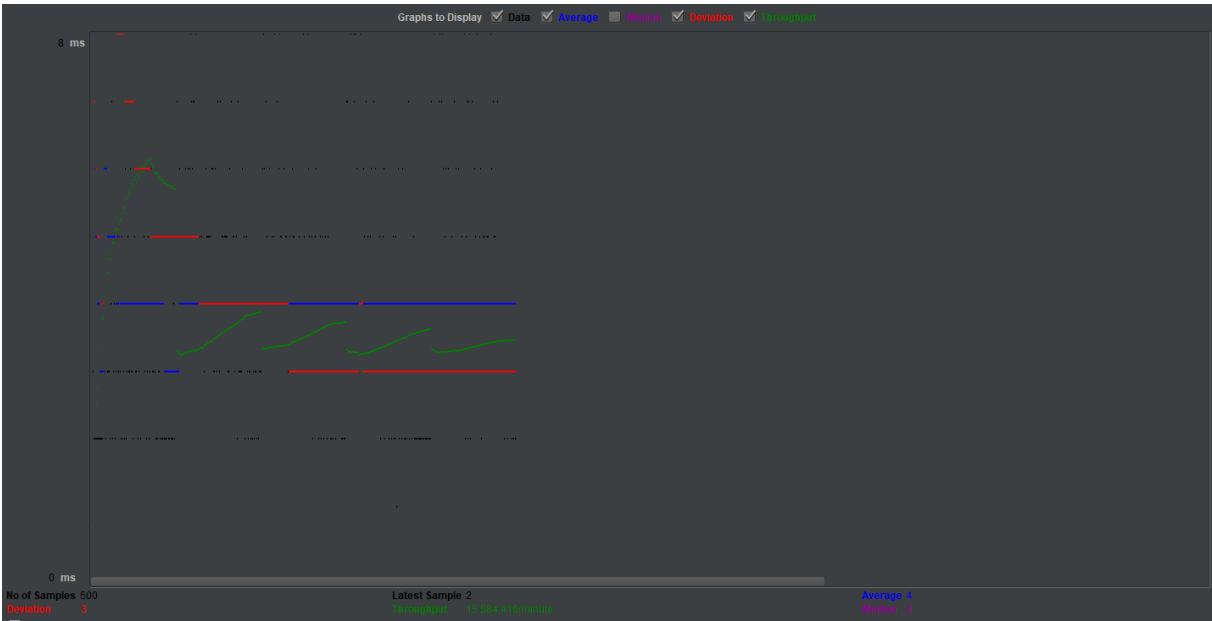
### 2.1 Stan gry w pamięci podręcznej

Aplikacja została wykorzystana w stanie niezmienionym od poprzedniego projektu i kod wraz z end-point'ami znajduje się pod ścieżką:

*rlis\_jmeter\scrabble\game\app\_cache.csv*

2.1.1 Test 1

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
5	2	50

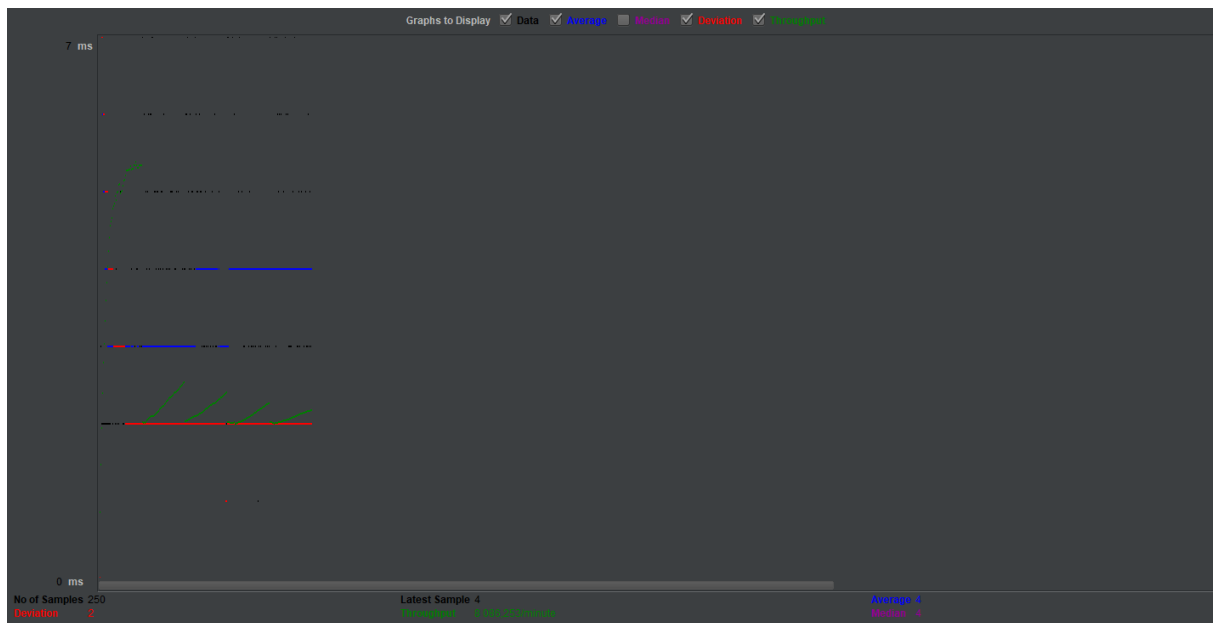


Rysunek 2: Wspólny wykres dla obu end-point’ów

2.1.1.1 GET - /game/view

Dane zagregowane												
Nazwa: Dane zagregowane												
Uwagi:												
Zapisuj/Czytaj wyniki z pliku												
Nazwa pliku	<div>Przeglądaj</div> <div>Zapisuj do loga</div> <div>Wyświetlaj tylko: <input type="checkbox"/> Błędy <input type="checkbox"/> Sukcesy</div> <div>Konfiguruj</div>											
Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	250	4	4	7	8	16	1	20	0,00%	134,8/sec	6858,81	16,45
RAZEM	250	4	4	7	8	16	1	20	0,00%	134,8/sec	6858,81	16,45

Rysunek 3: Zagregowane dane dla end-point’u /game/view



Rysunek 4: Wykres dla end-point'u */game/view*

#### 2.1.1.2 POST - */game/move*

Dane zagregowane

Nazwa: Dane zagregowane

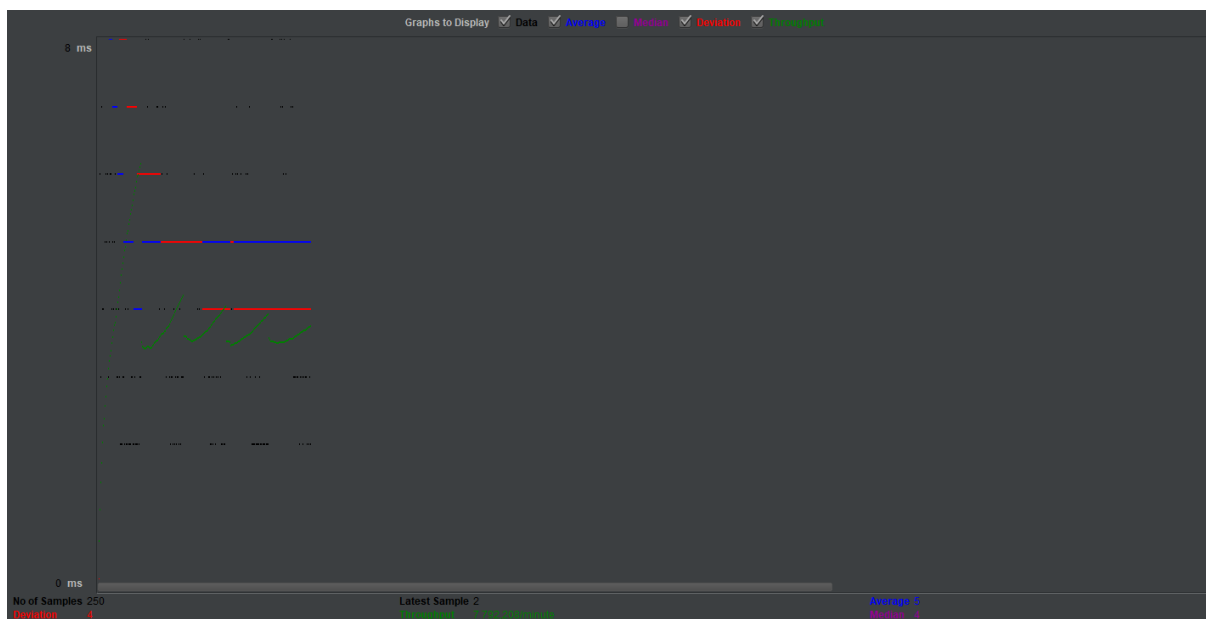
Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku  Przeglądaj... Zapisuj do loga Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy Konfiguruj

Etykieta	# Liczba próbek	Srednia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	250	5	4	10	10	22	2	50	0.00%	129.9/sec	53.14	59.89
RAZEM	250	5	4	10	10	22	2	50	0.00%	129.9/sec	53.14	59.89

Rysunek 5: Zagregowane dane dla end-point'u */game/move*



Rysunek 6: Wykres dla end-point'u */game/move*

## 2.1.2 Test 2

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
15	2	50



Rysunek 7: Wspólny wykres dla obu end-point'ów

2.1.2.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do logaWyświetlaj tylko: 

Błędy

Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	750	4	5	7	8	11	1	13	0.00%	357.3/sec	18281.24	43.62
RAZEM	750	4	5	7	8	11	1	13	0.00%	357.3/sec	18281.24	43.62

Rysunek 8: Zagregowane dane dla end-point'u /game/view



Rysunek 9: Wykres dla end-point'u /game/view

2.1.2.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do logaWyświetlaj tylko: 

Błędy

Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	750	15	14	28	32	41	2	55	0.00%	314.7/sec	128.78	145.50
RAZEM	750	15	14	28	32	41	2	55	0.00%	314.7/sec	128.78	145.50

Rysunek 10: Zagregowane dane dla end-point'u /game/move

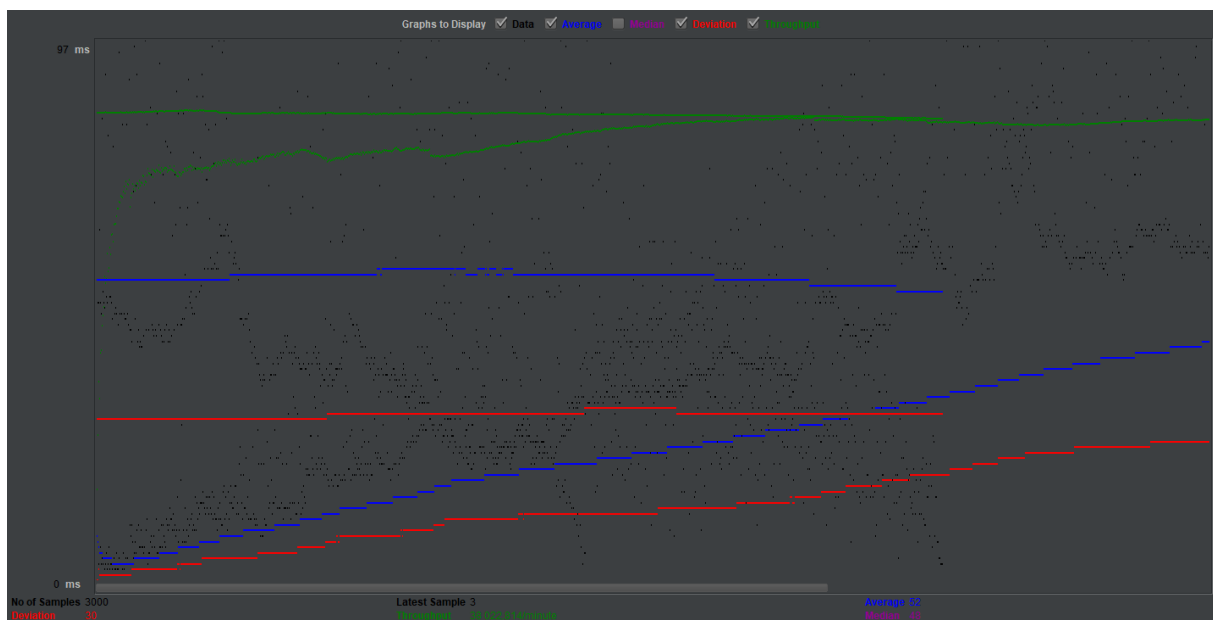




Rysunek 11: Wykres dla end-point'u /game/move

### 2.1.3 Test 3

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
30	2	50



Rysunek 12: Wspólny wykres dla obu end-point'ów

### 2.1.3.1 GET - /game/view

Dane zagregowane

Nazwa: 

Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: 

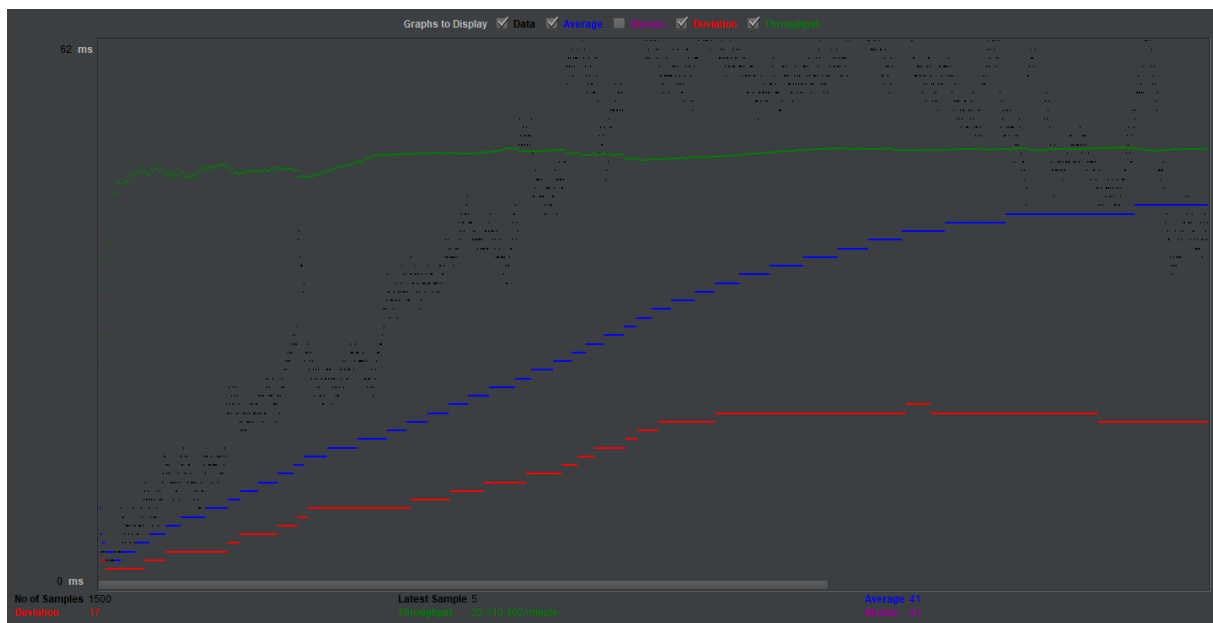
Błędy

Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	1500	41	44	62	64	71	2	78	0.00%	371.8/sec	19024.45	45.39
RAZEM	1500	41	44	62	64	71	2	78	0.00%	371.8/sec	19024.45	45.39

Rysunek 13: Zagregowane dane dla end-point'u /game/view



Rysunek 14: Wykres dla end-point'u /game/view

### 2.1.3.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj...

Zapisuj do loga

Wyświetlaj tylko: 

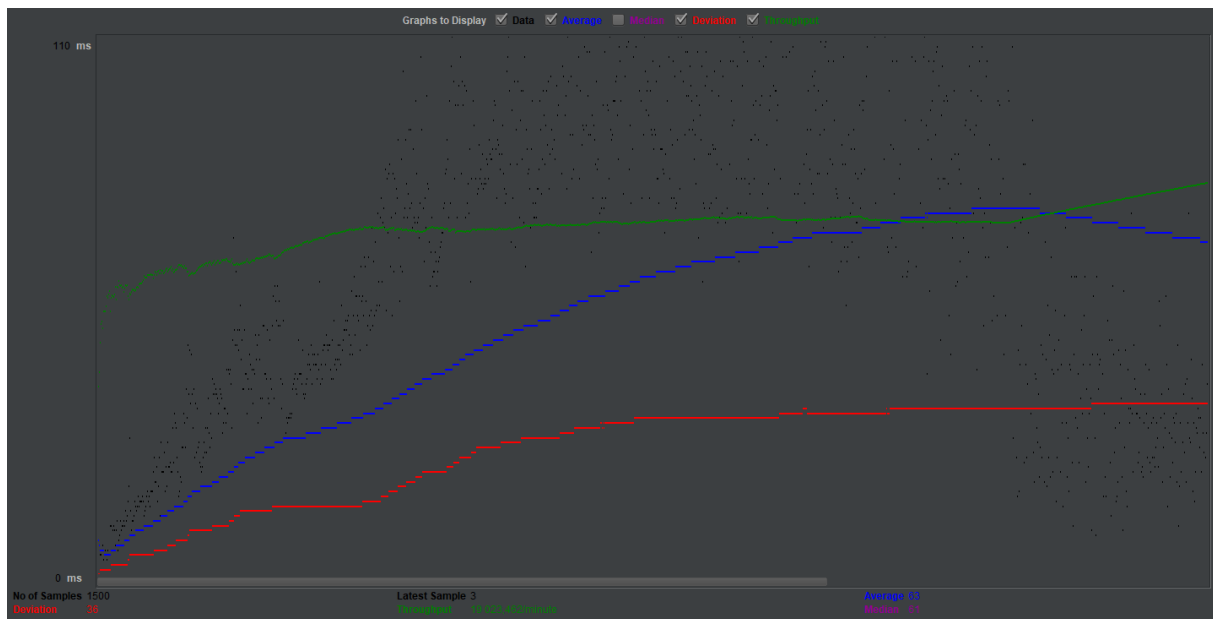
☐ Błędy

☐ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	1500	63	61	110	123	156	3	190	0.00%	317.1/sec	129.73	146.45
RAZEM	1500	63	61	110	123	156	3	190	0.00%	317.1/sec	129.73	146.45

Rysunek 15: Zagregowane dane dla end-point'u /game/move



Rysunek 16: Wykres dla end-point'u /game/move

#### 2.1.4 Test 4

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
5	4	50



Rysunek 17: Wspólny wykres dla obu end-point'ów

2.1.4.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: 

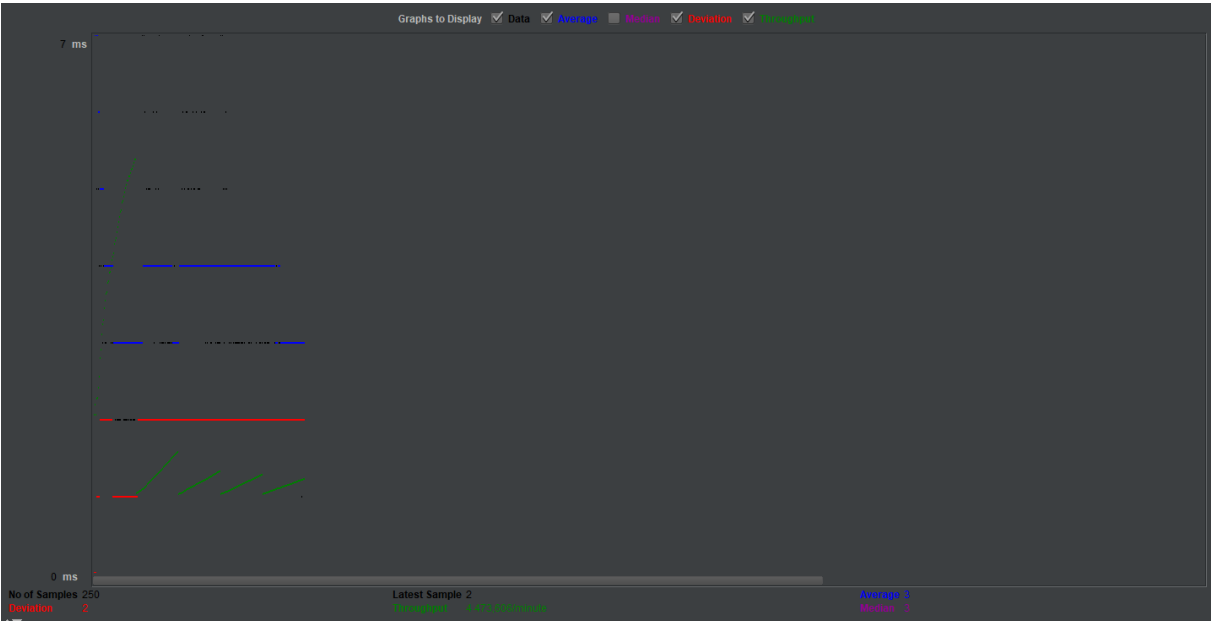
Błędy

Sukcesy

Konfiguruj

	Wykryta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view		250	3	3	7	9	12	1	13	0.00%	74.6/sec	3814.72	9.10
RAZEM		250	3	3	7	9	12	1	13	0.00%	74.6/sec	3814.72	9.10

Rysunek 18: Zagregowane dane dla end-point'u /game/view



Rysunek 19: Wykres dla end-point'u /game/view

2.1.4.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: 

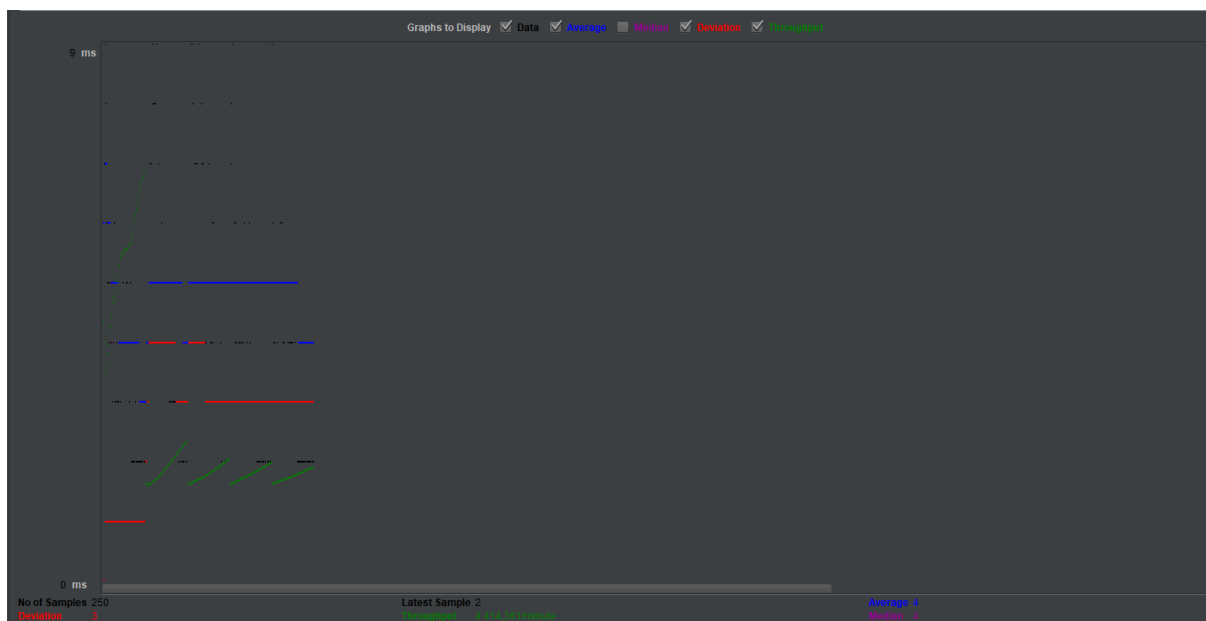
Błędy

Sukcesy

Konfiguruj

	Wykryta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move		250	4	4	9	11	18	2	27	0.00%	73.6/sec	30.10	34.05
RAZEM		250	4	4	9	11	18	2	27	0.00%	73.6/sec	30.10	34.05

Rysunek 20: Zagregowane dane dla end-point'u /game/move



Rysunek 21: Wykres dla end-point'u `/game/move`

### 2.1.5 Test 5

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
15	4	50



Rysunek 22: Wspólny wykres dla obu end-point'ów

2.1.5.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: 

Błędy

Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	750	3	3	5	7	11	1	16	0.00%	193.6/sec	9905.09	23.63
RAZEM	750	3	3	5	7	11	1	16	0.00%	193.6/sec	9905.09	23.63

Rysunek 23: Zagregowane dane dla end-point’u /game/view



Rysunek 24: Wykres dla end-point’u /game/view

2.1.5.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: 

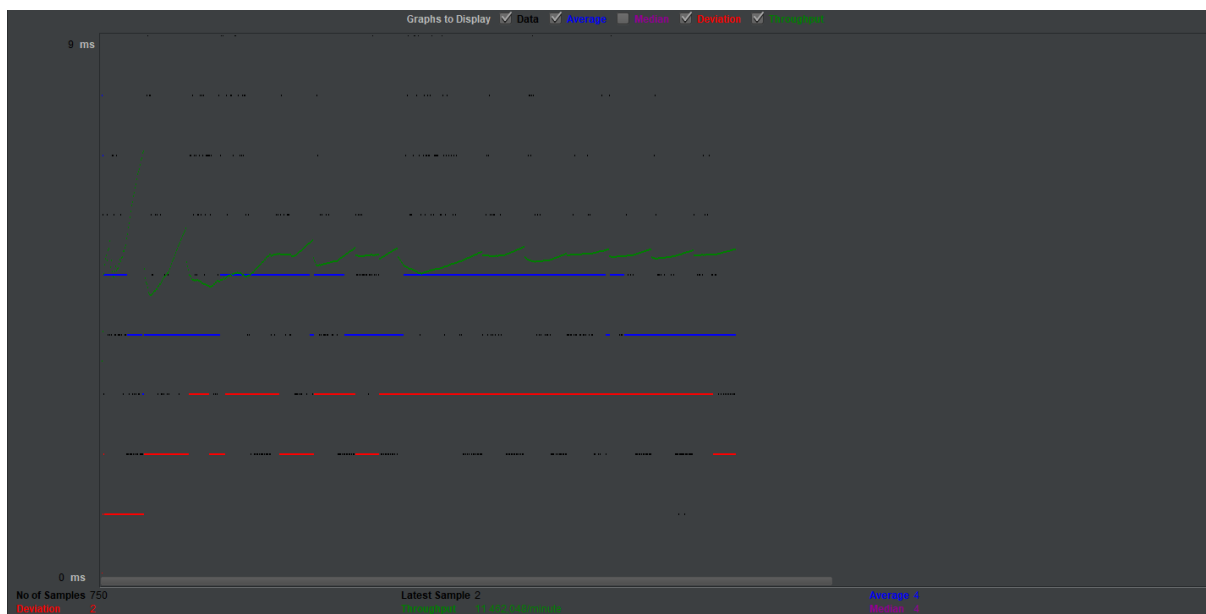
Błędy

Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	750	4	4	8	11	15	1	29	0.00%	191.0/sec	78.17	88.27
RAZEM	750	4	4	8	11	15	1	29	0.00%	191.0/sec	78.17	88.27

Rysunek 25: Zagregowane dane dla end-point’u /game/move



Rysunek 26: Wykres dla end-point'u */game/move*

## 2.1.6 Test 6

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
30	4	50



Rysunek 27: Wspólny wykres dla obu end-point'ów

2.1.6.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: ☐ Błędy ☒ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Srednia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sek	Wysłano KB/sek
get_view	1500	5	5	9	10	13	2	30	0.00%	367.6/sec	18809.96	44.88
RAZEM	1500	5	5	9	10	13	2	30	0.00%	367.6/sec	18809.96	44.88

Rysunek 28: Zagregowane dane dla end-point'u /game/view



Rysunek 29: Wykres dla end-point'u /game/view

2.1.6.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

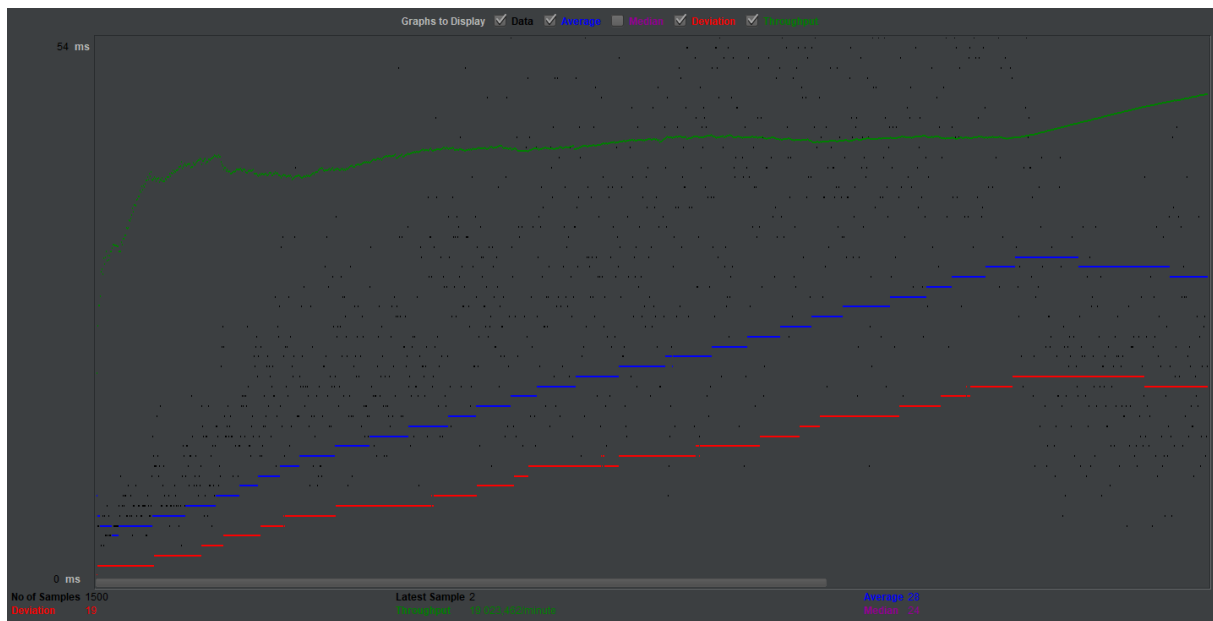
Wyświetlaj tylko: ☐ Błędy ☒ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Srednia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sek	Wysłano KB/sek
post_move	1500	28	24	54	66	93	2	151	0.00%	317.1/sec	129.73	146.57
RAZEM	1500	28	24	54	66	93	2	151	0.00%	317.1/sec	129.73	146.57

Rysunek 30: Zagregowane dane dla end-point'u /game/move





Rysunek 31: Wykres dla end-point'u /game/move

### 2.1.7 Dodatkowy test

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
100	2	50

Dane zagregowane													
Nazwa: Dane zagregowane													
Uwagi:													
Zapisuj/Czytaj wyniki z pliku													
Nazwa pliku	Przełączaj... Zapisuj do loga/Wyświetlaj tylko: <input type="checkbox"/> Błędy <input type="checkbox"/> Sukcesy <input type="button" value="Konfiguruj"/>												
Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec	
get_view	5000	238	172	216	711	2001	2	2011	1,88%	324,9/sec	16326,72	38,92	
RAZEM	5000	238	172	216	711	2001	2	2011	1,88%	324,9/sec	16326,72	38,92	

Rysunek 32: Wykres dla end-point'u /game/view

Dane zagregowane													
Nazwa: Dane zagregowane													
Uwagi:													
Zapisuj/Czytaj wyniki z pliku													
Nazwa pliku	Przełączaj... Zapisuj do loga/Wyświetlaj tylko: <input type="checkbox"/> Błędy <input type="checkbox"/> Sukcesy <input type="button" value="Konfiguruj"/>												
Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec	
post_move	5000	301	236	444	776	2001	4	2012	1,70%	293,0/sec	131,58	133,03	
RAZEM	5000	301	236	444	776	2001	4	2012	1,70%	293,0/sec	131,58	133,03	

Rysunek 33: Wykres dla end-point'u /game/move

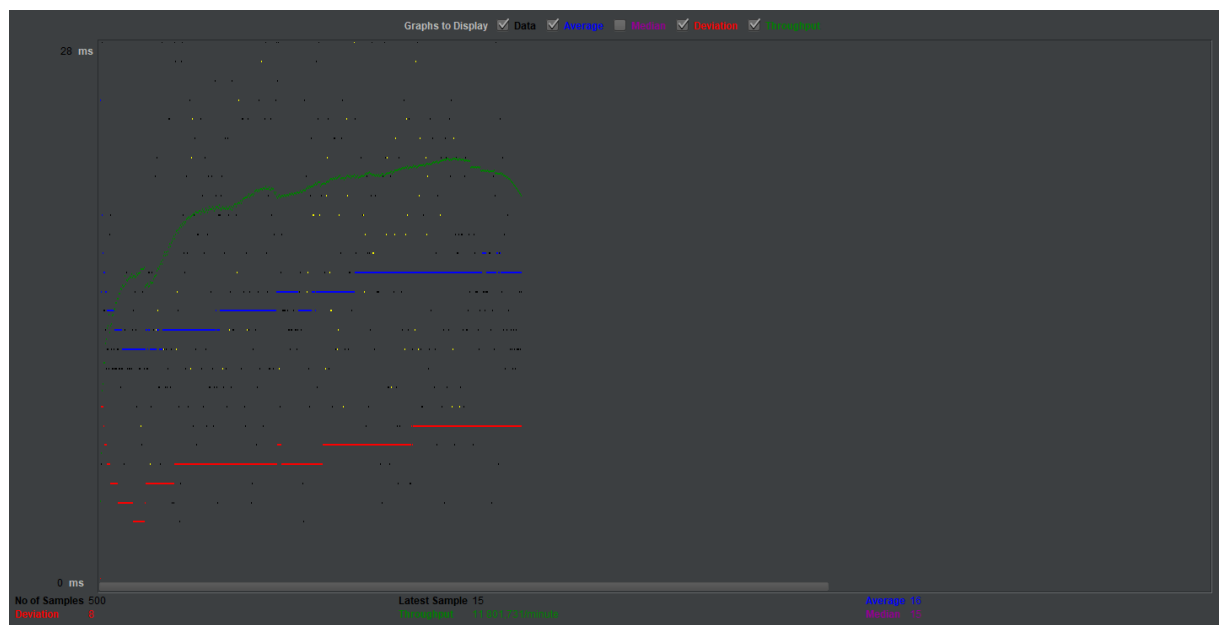
## 2.2 Stan gry w pliku tekstowym

Aplikacja została trochę zmodyfikowana, określony został inny sposób przechowywania oraz odczytu stanu gry z pliku. Kod znajduje się tutaj:

*rlis\_jmeter\scrabble\game\app\_file.csv*

### 2.2.1 Test 1

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
5	2	50

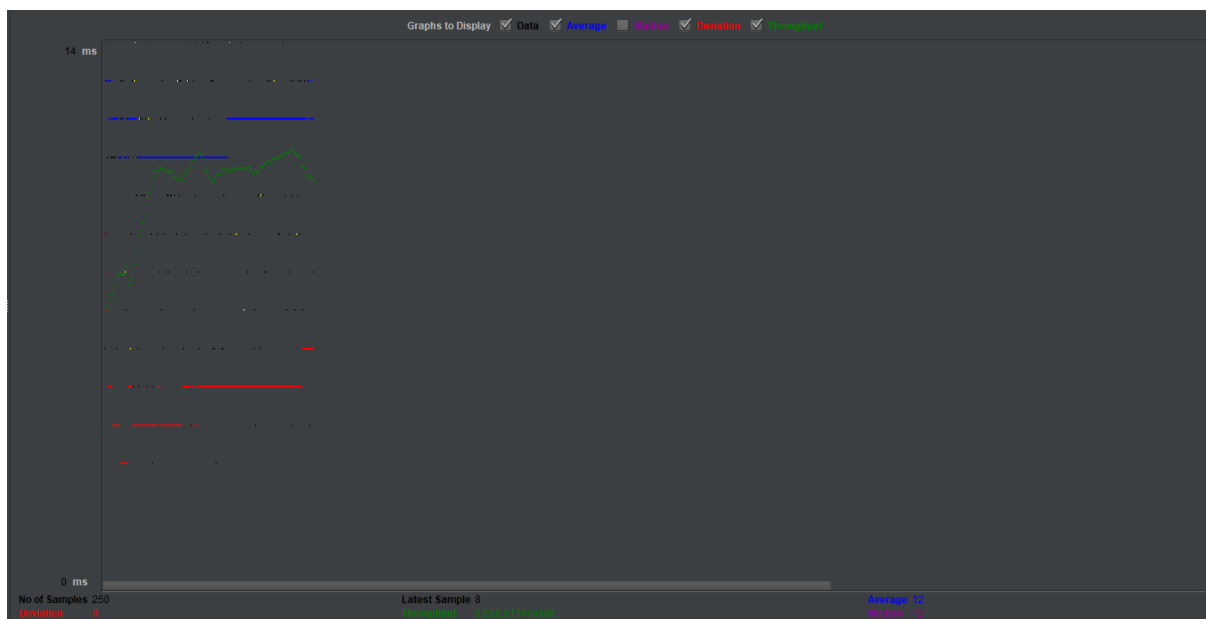


Rysunek 34: Wspólny wykres dla obu end-point'ów

#### 2.2.1.1 GET - /game/view

Dane zagregowane											
Nazwa: Dane zagregowane											
Uwagi:											
Zapisuj/Czytaj wyniki z pliku											
Nazwa pliku								Przełączaj	Zapisuj do loga/Wyświetlaj tylko:	Błędy	Sukcesy
Wykres	# Liczba próbek	Srednia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec
get_View	250	11	11	18	21	30	3	32	15.20%	115.2/sec	5034.02
RAZEM	250	11	11	18	21	30	3	32	15.20%	115.2/sec	5034.02

Rysunek 35: Zagregowane dane dla end-point'u /game/view



Rysunek 36: Wykres dla end-point'u */game/view*

### 2.2.1.2 POST - */game/move*

**Dane zagregowane**

Nazwa: Dane zagregowane

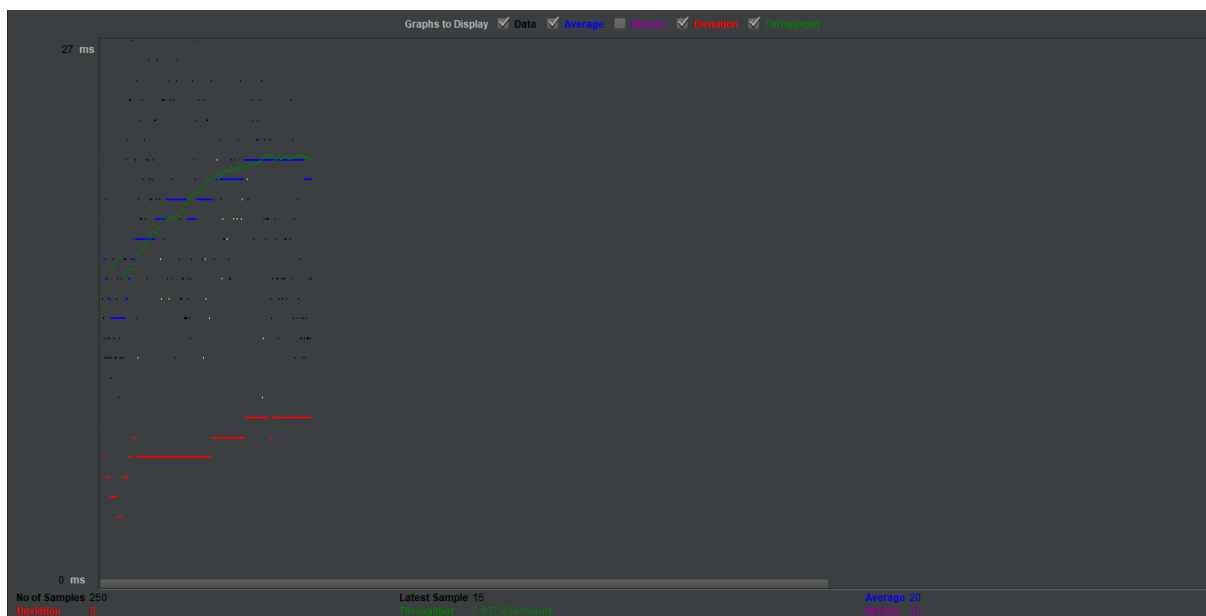
Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku Przeglądaj... Zapisuj do loga/Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	250	20	19	33	36	43	9	54	9.60%	98.5/sec	40.70	45.50
RAZEM	250	20	19	33	36	43	9	54	9.60%	98.5/sec	40.70	45.50

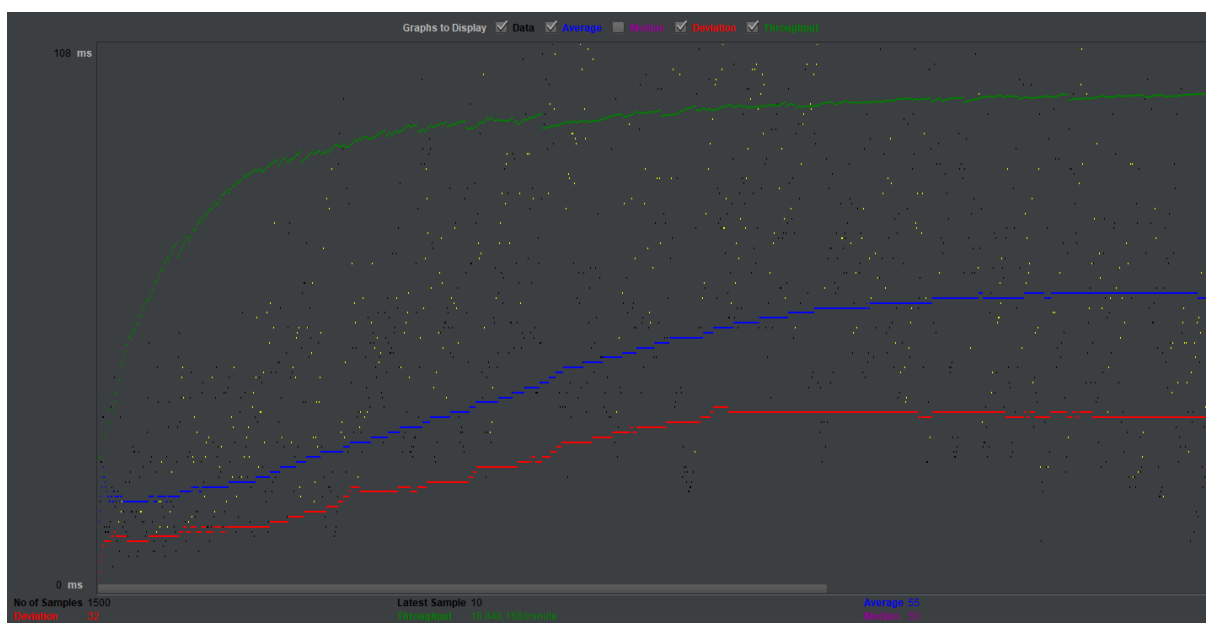
Rysunek 37: Zagregowane dane dla end-point'u */game/move*



Rysunek 38: Wykres dla end-point'u */game/move*

## 2.2.2 Test 2

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
15	2	50



Rysunek 39: Wspólny wykres dla obu end-point'ów

### 2.2.2.1 GET - /game/view

Dane zagregowane

Nazwa:

Dane zagregowane

Uwagi:

ZapisujCzytaj wyniki z pliku

Nazwa pliku

Przełączaj...

Zapisuj do logaWyświetlaj tylko: 

☐ Błędy

☐ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	750	44	41	78	95	122	3	133	31.73%	178.1/sec	6280.77	21.74
RAZEM	750	44	41	78	95	122	3	133	31.73%	178.1/sec	6280.77	21.74

Rysunek 40: Zagregowane dane dla end-point'u /game/view



Rysunek 41: Wykres dla end-point'u /game/view

### 2.2.2.2 POST - /game/move

Dane zagregowane

Nazwa:

Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj...

Zapisuj do logaWyświetlaj tylko: 

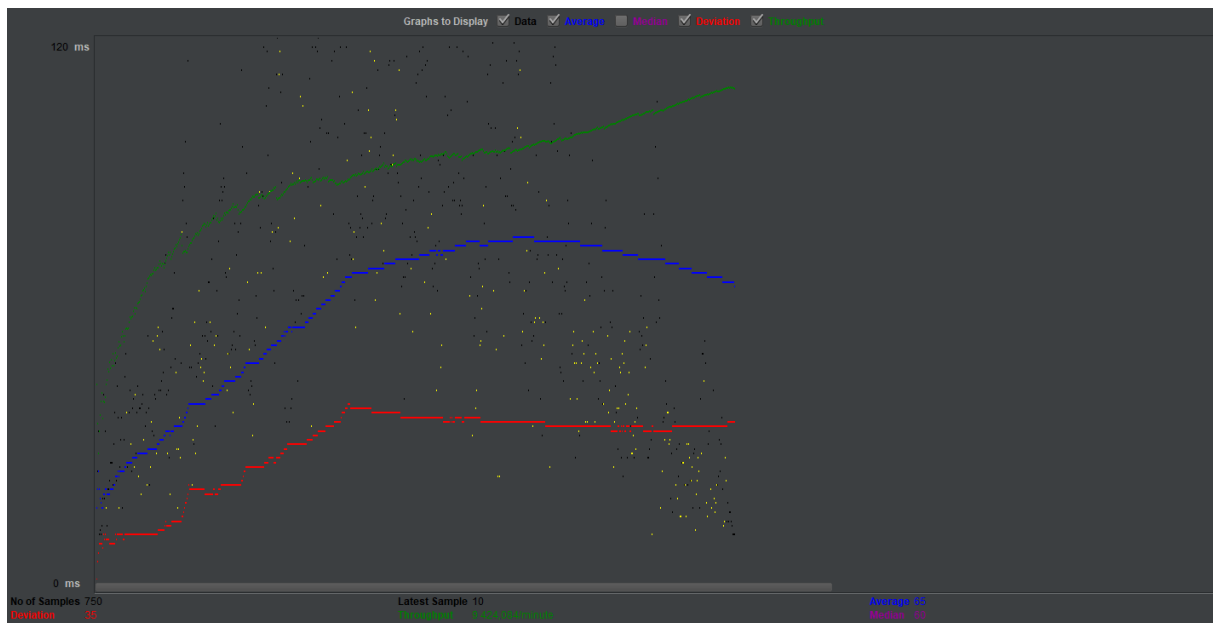
☐ Błędy

☐ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	750	65	60	117	127	156	10	224	30.13%	157.1/sec	66.30	72.48
RAZEM	750	65	60	117	127	156	10	224	30.13%	157.1/sec	66.30	72.48

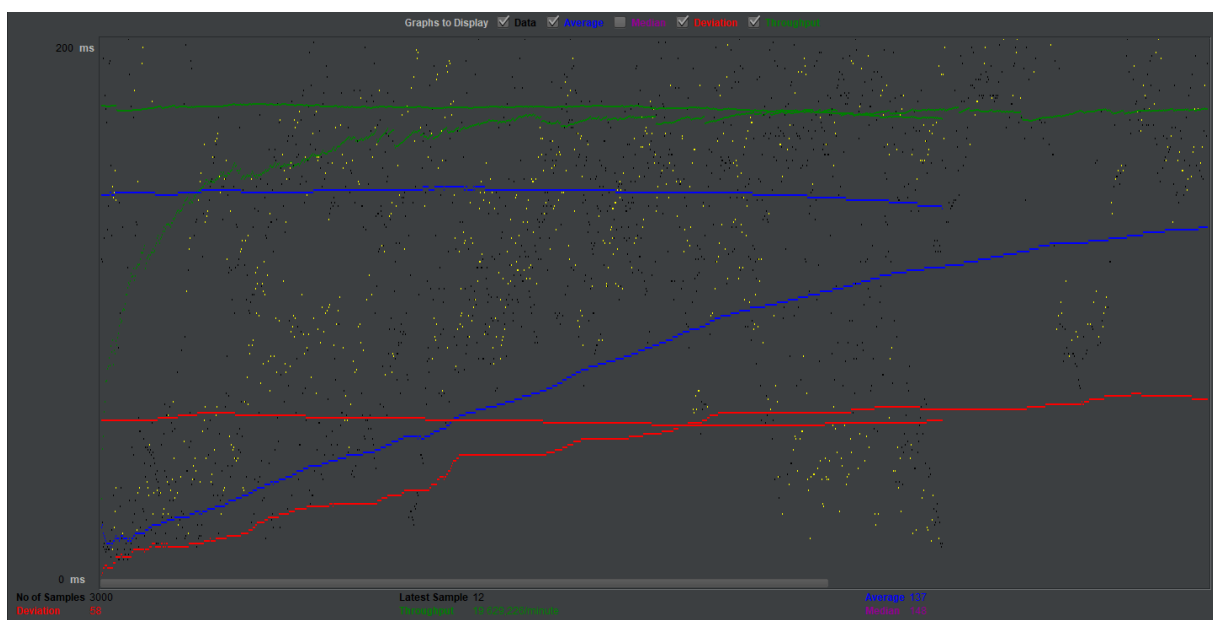
Rysunek 42: Zagregowane dane dla end-point'u /game/move



Rysunek 43: Wykres dla end-point'u */game/move*

### 2.2.3 Test 3

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
30	2	50



Rysunek 44: Wspólny wykres dla obu end-point'ów

2.2.3.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

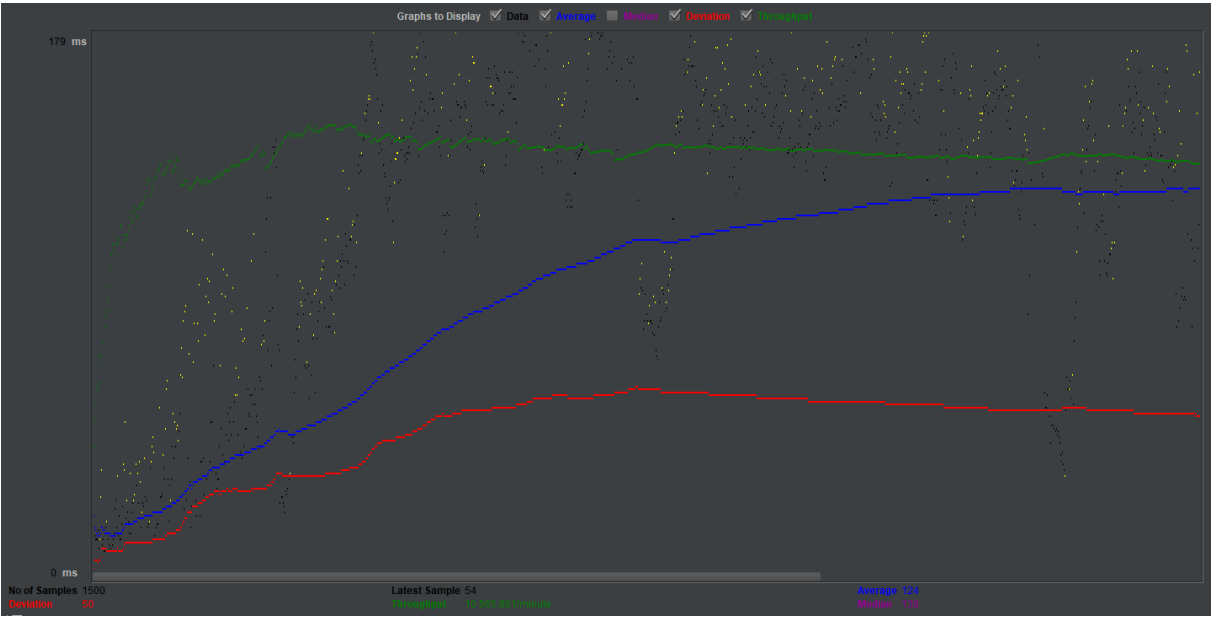
Przełączaj

Zapisuj do loga

Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy ☐ Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	1500	124	138	179	190	216	6	303	32.20%	176.1/sec	6168.07	21.50
RAZEM	1500	124	138	179	190	216	6	303	32.20%	176.1/sec	6168.07	21.50

Rysunek 45: Zagregowane dane dla end-point'u /game/view



Rysunek 46: Wykres dla end-point'u /game/view

2.2.3.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

Zapisuj do loga

Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy ☐ Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	1500	151	162	213	236	353	9	445	31.53%	163.6/sec	69.16	75.55
RAZEM	1500	151	162	213	236	353	9	445	31.53%	163.6/sec	69.16	75.55

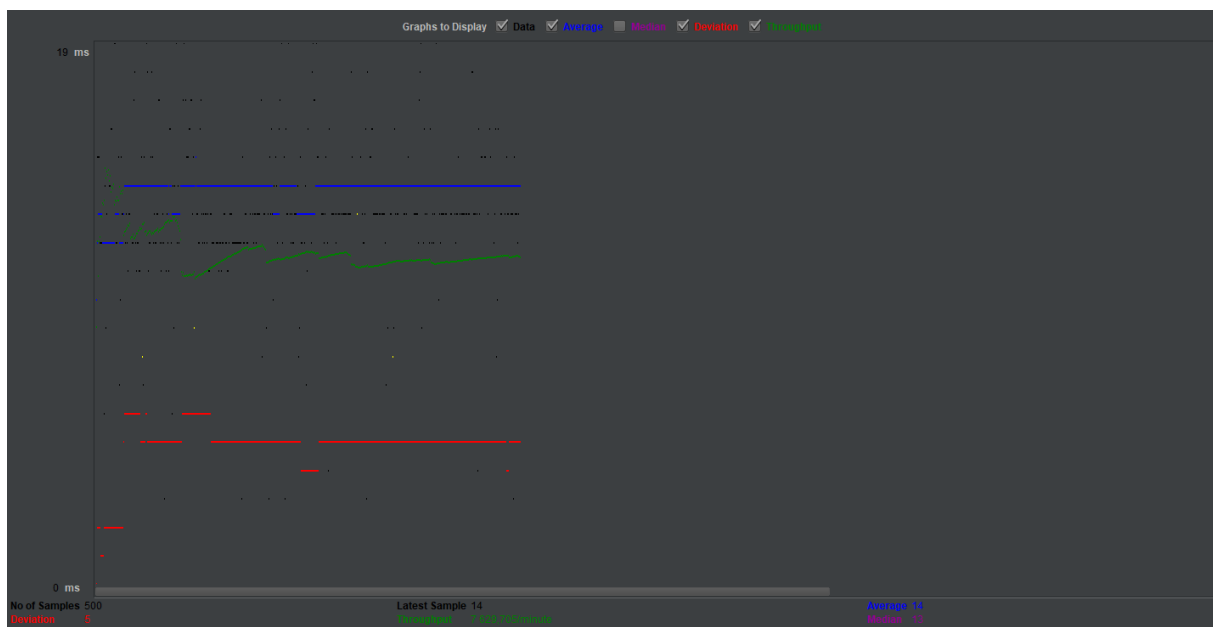
Rysunek 47: Zagregowane dane dla end-point'u /game/move



Rysunek 48: Wykres dla end-point'u /game/move

#### 2.2.4 Test 4

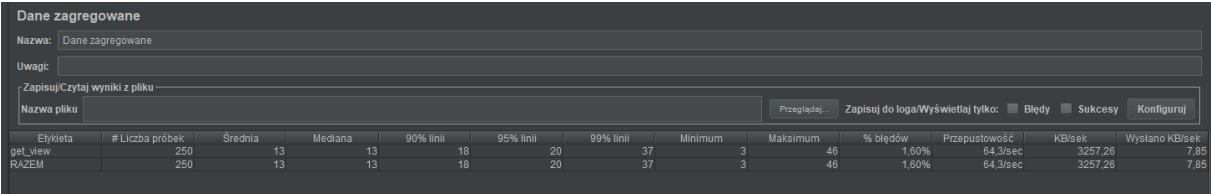
Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
5	4	50



Rysunek 49: Wspólny wykres dla obu end-point'ów



2.2.4.1 GET - /game/view

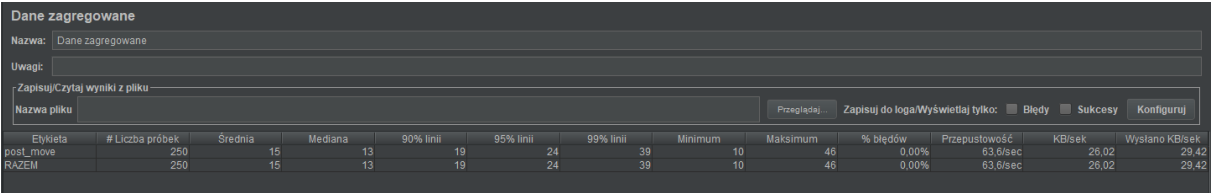


Rysunek 50: Zagregowane dane dla end-point'u /game/view

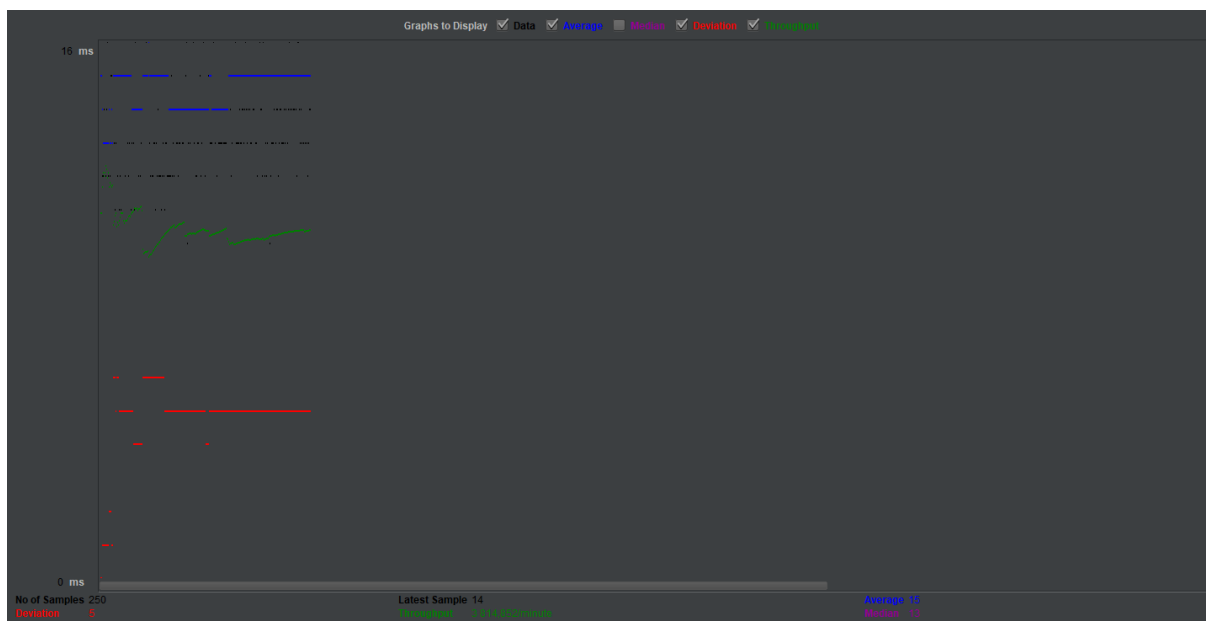


Rysunek 51: Wykres dla end-point'u /game/view

2.2.4.2 POST - /game/move



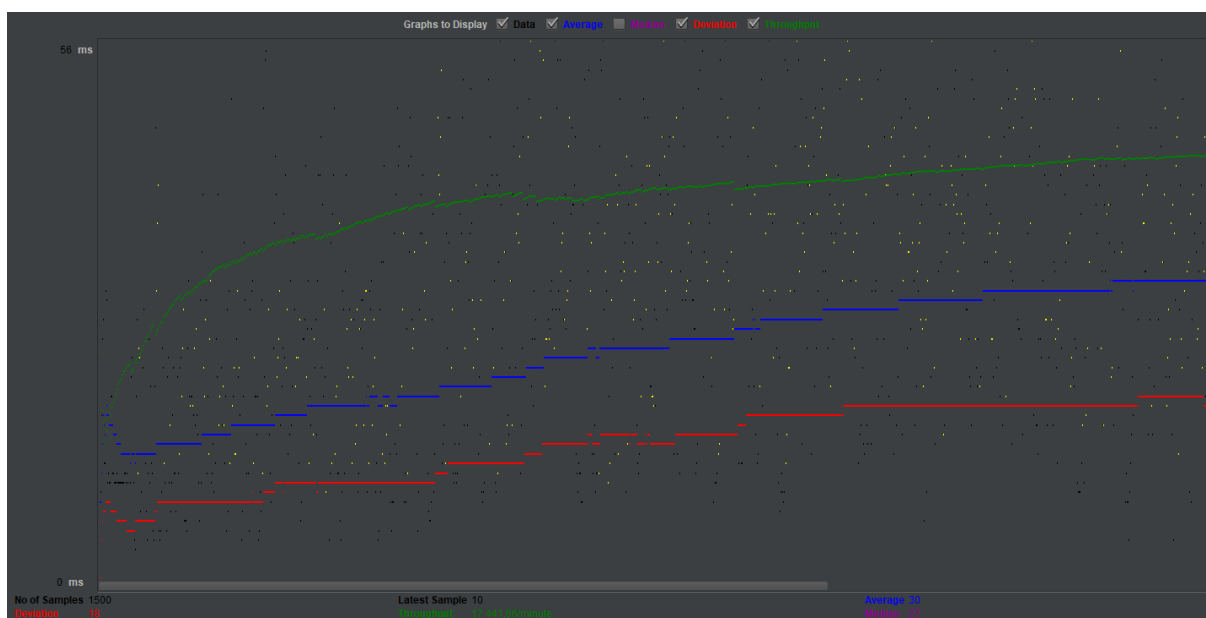
Rysunek 52: Zagregowane dane dla end-point'u /game/move



Rysunek 53: Wykres dla end-point'u /game/move

### 2.2.5 Test 5

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
15	4	50

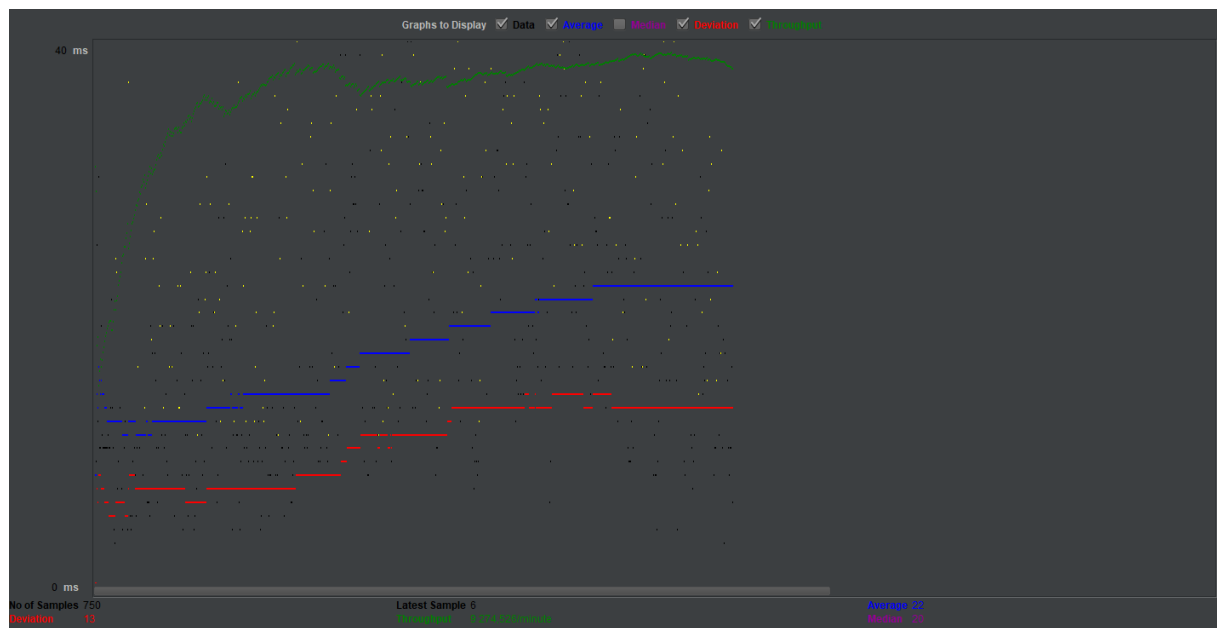


Rysunek 54: Wspólny wykres dla obu end-point'ów

2.2.5.1 GET - /game/view

Dane zagregowane													
Nazwa: Dane zagregowane													
Uwagi:													
Zapisuj/Czytaj wyniki z pliku													
Nazwa pliku										Przełączaj	Zapisuj do loga	Wyświetlaj tylko:	
												<input type="checkbox"/> Błędy	<input type="checkbox"/> Sukcesy
Konfiguruj													
Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec	
get_view	750	22	20	40	46	67	3	103	34.40%	154.6/sec	5240.85	18.87	
RAZEM	750	22	20	40	46	67	3	103	34.40%	154.6/sec	5240.85	18.87	

Rysunek 55: Zagregowane dane dla end-point'u /game/view

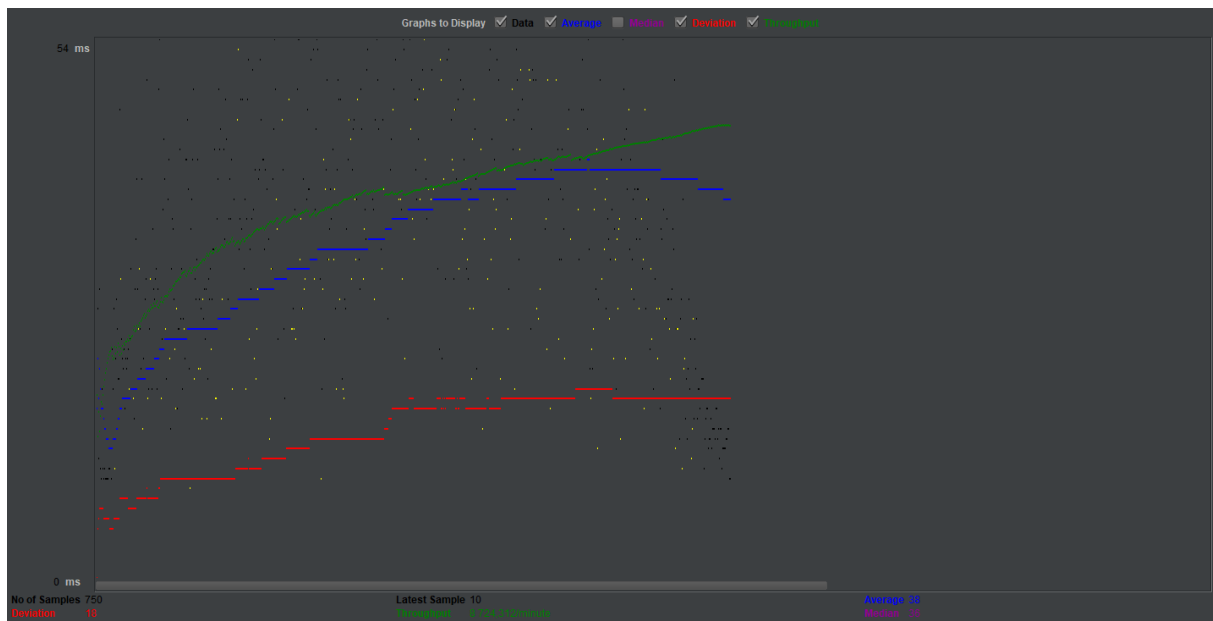


Rysunek 56: Wykres dla end-point'u /game/view

2.2.5.2 POST - /game/move

Dane zagregowane													
Nazwa: Dane zagregowane													
Uwagi:													
Zapisuj/Czytaj wyniki z pliku													
Nazwa pliku										Przełączaj	Zapisuj do loga	Wyświetlaj tylko:	
												<input type="checkbox"/> Błędy	<input type="checkbox"/> Sukcesy
Konfiguruj													
Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec	
post_move	750	38	36	65	75	97	9	111	29.73%	145.4/sec	61.35	67.22	
RAZEM	750	38	36	65	75	97	9	111	29.73%	145.4/sec	61.35	67.22	

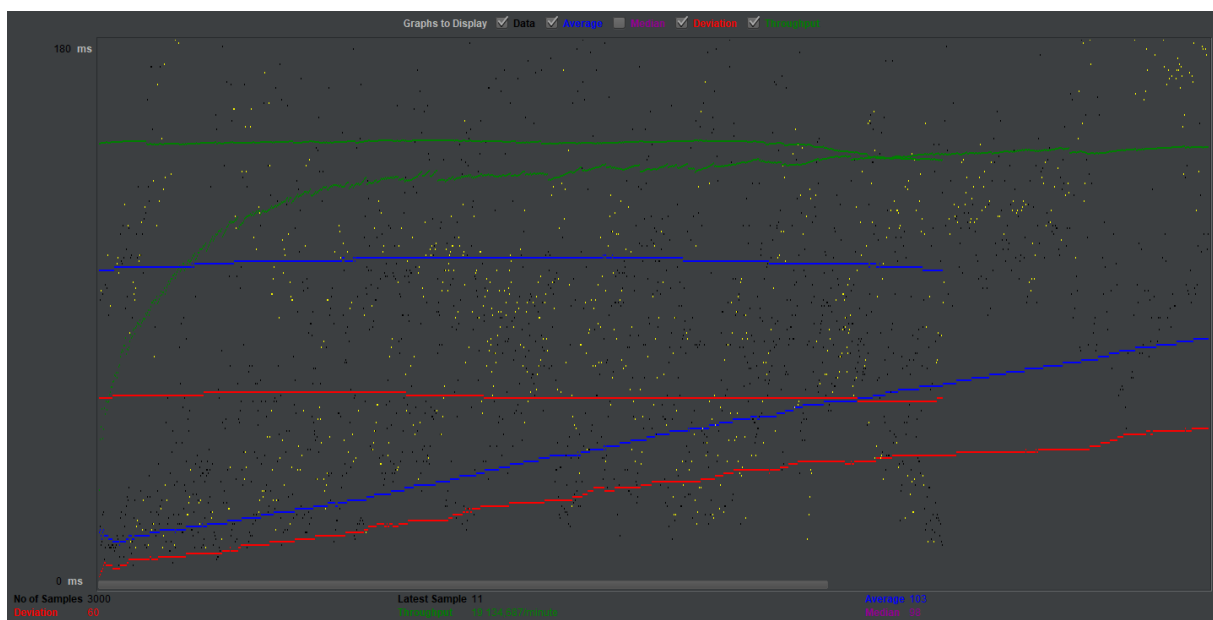
Rysunek 57: Zagregowane dane dla end-point'u /game/move



Rysunek 58: Wykres dla end-point'u */game/move*

## 2.2.6 Test 6

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
30	4	50



Rysunek 59: Wspólny wykres dla obu end-point'ów

2.2.6.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

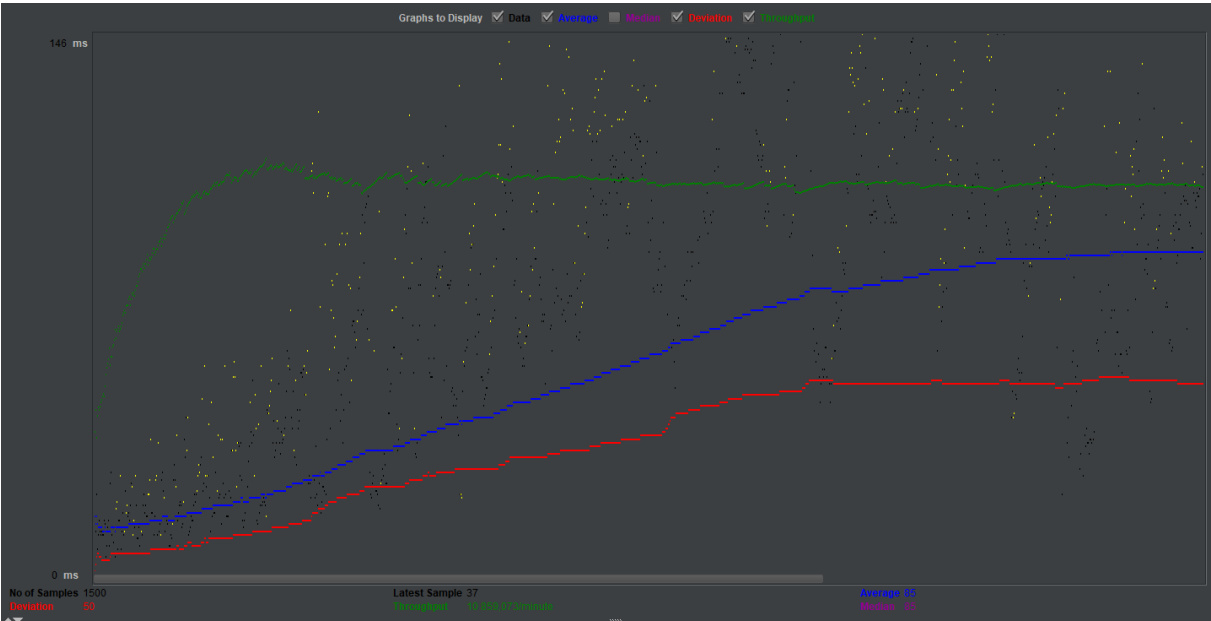
Przełączaj...

Zapisuj do loga/Wyświetlaj tylko: 

Błędy  Sukcesy  Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sek	Wysłano KB/sek
get_view	1500	85	85	146	172	241	4	320	33,40%	181,0/sec	6228,49	22,09
RAZEM	1500	85	85	146	172	241	4	320	33,40%	181,0/sec	6228,49	22,09

Rysunek 60: Zagregowane dane dla end-point'u /game/view



Rysunek 61: Wykres dla end-point'u /game/view

2.2.6.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

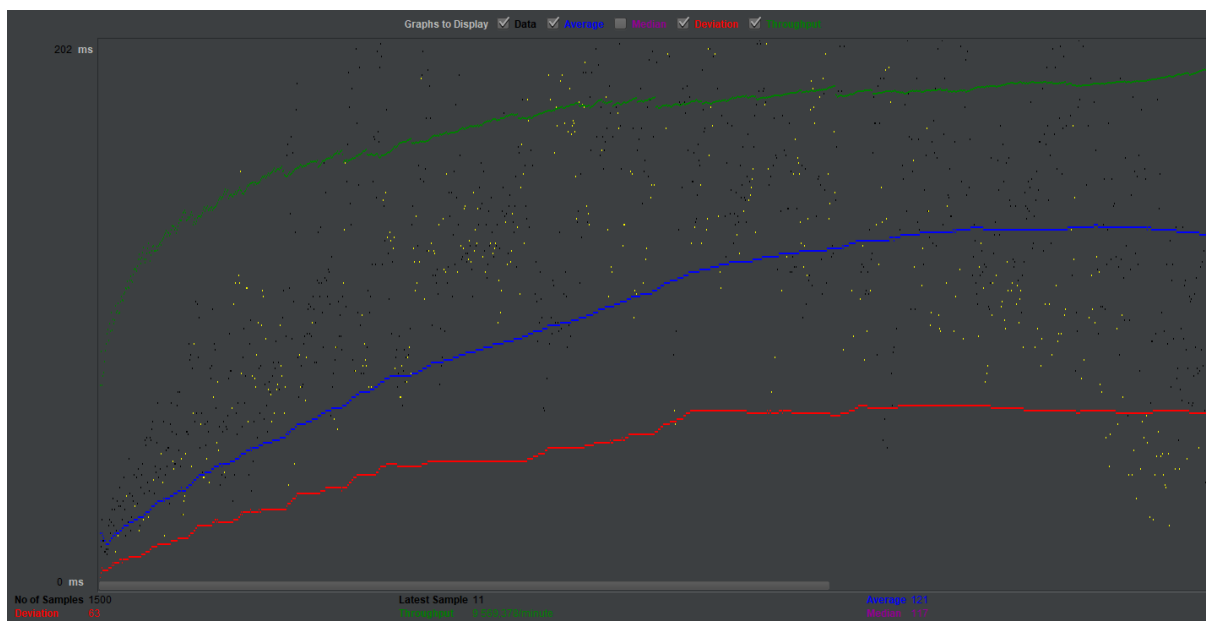
Przełączaj...

Zapisuj do loga/Wyświetlaj tylko: 

Błędy  Sukcesy  Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sek	Wysłano KB/sek
post_move	1500	121	117	202	237	317	9	386	29,60%	159,5/sec	67,29	73,64
RAZEM	1500	121	117	202	237	317	9	386	29,60%	159,5/sec	67,29	73,64

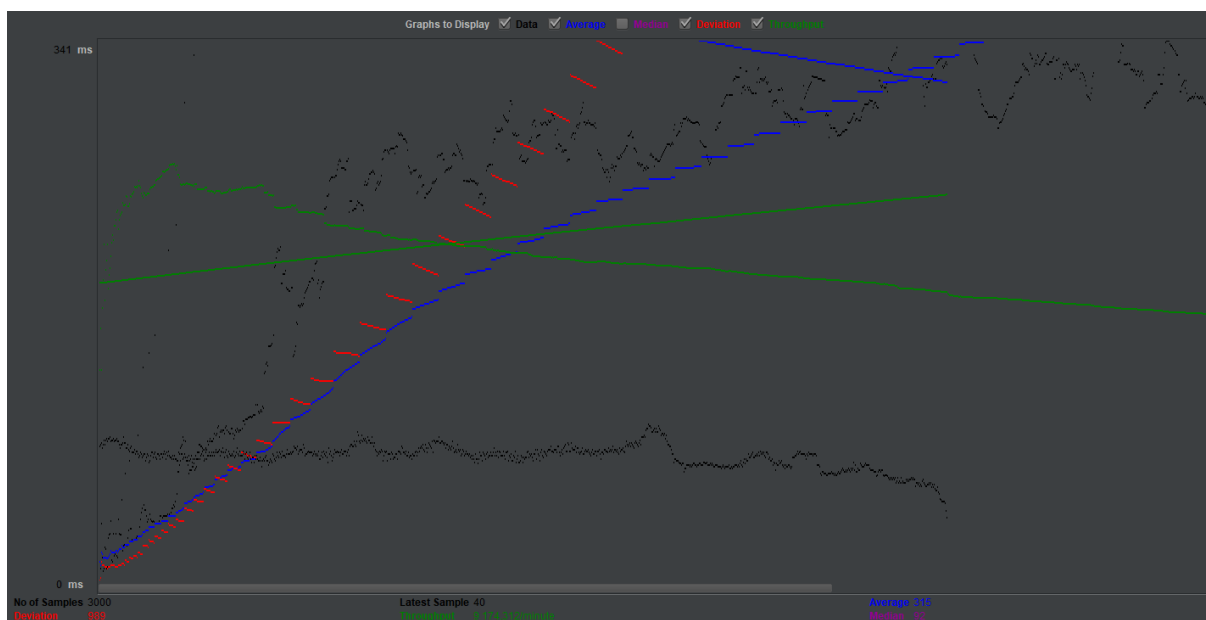
Rysunek 62: Zagregowane dane dla end-point'u /game/move



Rysunek 63: Wykres dla end-point'u /game/move

## 2.2.7 Dodatkowy test

Liczba użytkowników	Uruchom w ciągu (sekund)	Liczba powtórzeń
30	2	50



Rysunek 64: Wspólny wykres dla obu end-point'ów

2.2.7.1 GET - /game/view

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

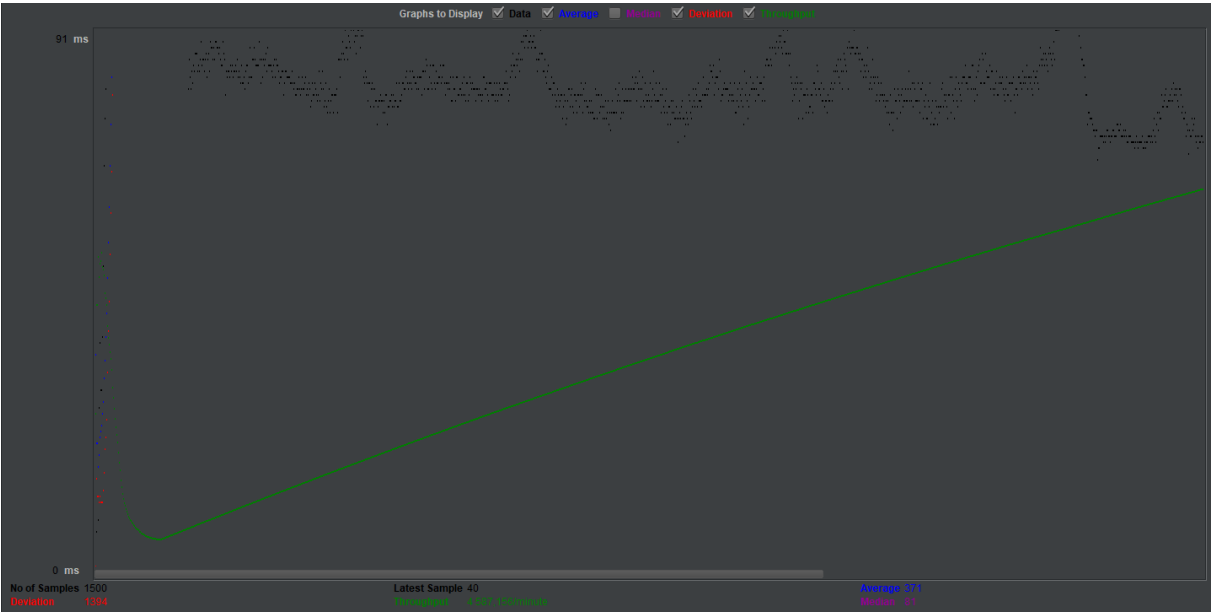
Zapisuj do loga

Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
get_view	1500	371	81	90	967	8975	6	9304	0.00%	76,5/sec	3932,63	9,33
RAZEM	1500	371	81	90	967	8975	6	9304	0.00%	76,5/sec	3932,63	9,33

Rysunek 65: Zagregowane dane dla end-point'u /game/view



Rysunek 66: Wykres dla end-point'u /game/view

2.2.7.2 POST - /game/move

Dane zagregowane

Nazwa: Dane zagregowane

Uwagi:

Zapisuj/Czytaj wyniki z pliku

Nazwa pliku

Przełączaj

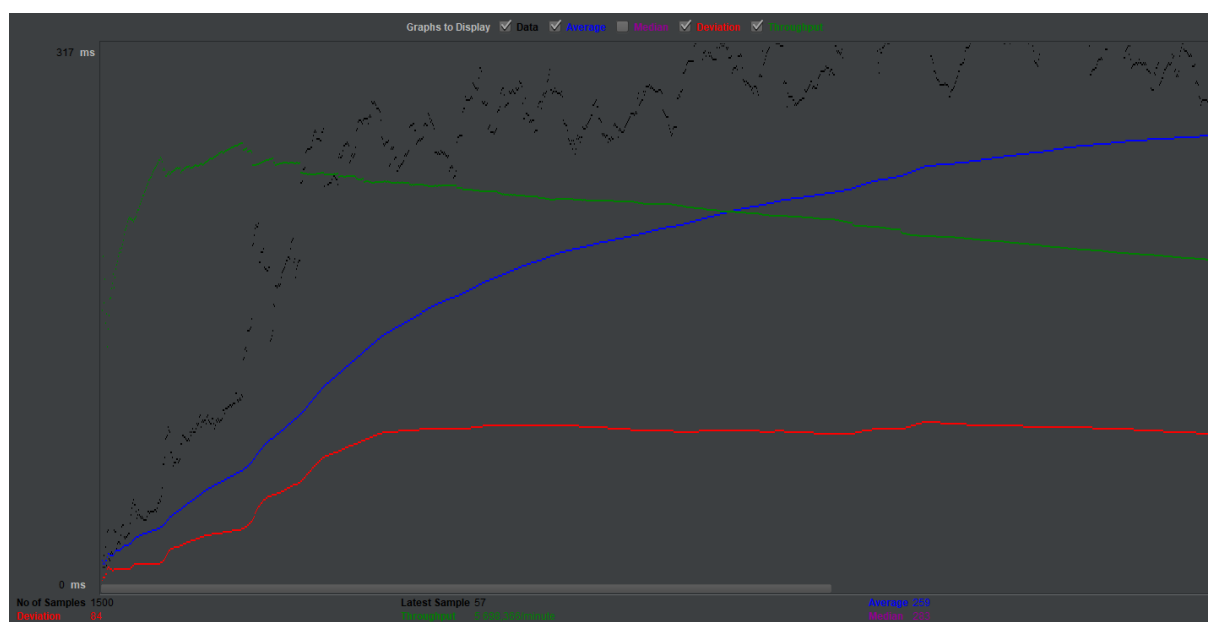
Zapisuj do loga

Wyświetlaj tylko: ☐ Błędy ☐ Sukcesy

Konfiguruj

Etykieta	# Liczba próbek	Średnia	Mediana	90% linii	95% linii	99% linii	Minimum	Maksimum	% błędów	Przepustowość	KB/sec	Wysłano KB/sec
post_move	1500	259	283	327	350	434	8	444	0.00%	95,0/sec	38,86	43,88
RAZEM	1500	259	283	327	350	434	8	444	0.00%	95,0/sec	38,86	43,88

Rysunek 67: Zagregowane dane dla end-point'u /game/move



Rysunek 68: Wykres dla end-point'u */game/move*



### 3 Analiza wyników

Tablica S1: Zagregowane wyniki dla end-point'u /game/move

	Test	Wątki	Czas [s]	Mediana [ms]	Minimum [ms]	Maksimum [ms]	% błędów	Przepustowość
cache	Test 1	5	2	4	2	50	0.00	129.9/sek
	Test 2	15	2	14	2	55	0.00	314.7/sek
	Test 3	30	2	61	3	190	0.00	317.1/sek
	Test 4	5	4	4	2	27	0.00	73.6/sek
	Test 5	15	4	4	1	29	0.00	191.0/sek
	Test 6	30	4	24	2	151	0.00	317.1/sek
	Dodatkowy	100	2	236	4	2012	1.70	293.0/sek
plik	Test 1	5	2	19	9	54	9.60	98.5/sek
	Test 2	15	2	60	10	224	30.13	157.1/sek
	Test 3	30	2	162	9	445	31.53	163.6/sek
	Test 4	5	4	13	10	46	0.00	63.6/sek
	Test 5	15	4	36	9	111	29.73	145.4/sek
	Test 6	30	4	121	9	386	29.60	159.5/sek
	Dodatkowy	30	2	283	8	444	0.00	95.0/sek

Tablica S2: Zagregowane wyniki dla end-point'u /game/view

	Test	Wątki	Czas [s]	Mediana [ms]	Minimum [ms]	Maksimum [ms]	% błędów	Przepustowość
cache	Test 1	5	2	4	1	20	0.00	134.8/sek
	Test 2	15	2	5	1	13	0.00	357.3/sek
	Test 3	30	2	44	2	78	0.00	371.8/sek
	Test 4	5	4	3	1	13	0.00	74.6/sek
	Test 5	15	4	3	1	16	0.00	193.6/sek
	Test 6	30	4	5	2	30	0.00	367.6/sek
	Dodatkowy	100	2	172	2	2011	1.88	324.9/sek
plik	Test 1	5	2	11	3	32	15.20	115.2/sek
	Test 2	15	2	41	3	133	31.73	178.1/sek
	Test 3	30	2	138	6	303	32.20	176.1/sek
	Test 4	5	4	13	3	46	1.60	64.3/sek
	Test 5	15	4	20	3	103	34.40	154.6/sek
	Test 6	30	4	85	4	320	33.40	181.0/sek
	Dodatkowy	30	2	81	6	9304	0.00	76.5/sek

Sample #	Start Time	Thread Name	Etykieta	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:25:58.624	post_threads 2-1	post_move	91	✓	419	461	90	66
2	19:25:58.655	post_threads 2-2	post_move	61	✓	419	485	61	35
3	19:25:58.715	post_threads 2-1	post_move	18	✓	419	485	18	1
4	19:25:58.717	post_threads 2-2	post_move	20	✓	419	488	20	1
5	19:25:58.737	post_threads 2-2	post_move	14	✓	419	488	14	1
6	19:25:58.734	post_threads 2-1	post_move	20	✓	419	458	20	1
7	19:25:58.751	post_threads 2-2	post_move	19	✓	419	482	19	1
8	19:25:58.770	post_threads 2-2	post_move	12	✓	419	467	12	1
9	19:25:58.754	post_threads 2-1	post_move	30	✓	419	461	30	1
10	19:25:58.783	post_threads 2-2	post_move	29	✓	419	467	29	0
11	19:25:58.784	post_threads 2-1	post_move	31	✓	419	476	31	1
12	19:25:58.786	post_threads 2-3	post_move	30	✓	419	488	29	0
13	19:25:58.813	post_threads 2-2	post_move	19	✓	419	485	19	0
14	19:25:58.815	post_threads 2-1	post_move	22	✓	419	479	22	2
15	19:25:58.816	post_threads 2-3	post_move	30	✓	419	470	30	1
16	19:25:58.832	post_threads 2-2	post_move	22	✗	463	488	21	1
17	19:25:58.838	post_threads 2-1	post_move	19	✓	419	458	19	0

Rysunek 69: Przykładowe linijki danych

Sample #	Start Time	Thread Name	Etykieta	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	19:25:58.624	get_threads 1-1	get_view	78	✓	49843	125	77	66
2	19:25:58.648	get_threads 1-2	get_view	62	✓	50111	125	62	42
3	19:25:58.710	get_threads 1-2	get_view	13	✓	50111	125	13	1
4	19:25:58.724	get_threads 1-2	get_view	7	✓	50111	125	7	0
5	19:25:58.732	get_threads 1-2	get_view	13	✓	50380	125	13	0
6	19:25:58.745	get_threads 1-2	get_view	22	✓	50851	125	22	1
7	19:25:58.703	get_threads 1-1	get_view	66	✗	463	125	66	1
8	19:25:58.770	get_threads 1-1	get_view	8	✓	50918	125	8	0
9	19:25:58.767	get_threads 1-2	get_view	11	✓	50918	125	11	1
10	19:25:58.780	get_threads 1-3	get_view	24	✓	51184	125	24	1
11	19:25:58.778	get_threads 1-1	get_view	26	✓	51184	125	26	1
12	19:25:58.778	get_threads 1-2	get_view	27	✓	51184	125	27	1
13	19:25:58.805	get_threads 1-3	get_view	3	✓	51184	125	3	0
14	19:25:58.805	get_threads 1-1	get_view	5	✓	51453	125	5	0
15	19:25:58.805	get_threads 1-2	get_view	15	✓	51448	125	15	0
16	19:25:58.810	get_threads 1-1	get_view	18	✓	51448	125	18	1
17	19:25:58.808	get_threads 1-3	get_view	21	✓	51448	125	21	1

Rysunek 70: Przykładowe linijki danych

### 3.1 Wnioski

Po przeprowadzeniu każdego z zaplanowanych testów zostały wykonane jeszcze dwa dodatkowe - po jednym z osobna dla różnych sposobów przechowywania danych. Jako, że żaden z testów 1-6 (bez dodatkowego) dla cache'u nie dał jakichkolwiek błędów to zostały znalezione parametry dla których ten % błędów jednak się pojawił - należało podnieść liczbę wątków (użytkowników) do 100 i wtedy odsetek błędów zbliżył się do wartości 2%. Dodatkowy test dla aplikacji z danymi przechowywanymi w pliku polegał na dodaniu ochrony sekcji krytycznej (różne wątki próbują korzystać z tego samego pliku) i tym samym odsetek błędów wyniósł 0%, lecz czasy zdecydowanie wzrosły. W tabeli S2 mimo niedużej mediany, widać olbrzymie maksimum - ponad 9000ms, a sama średnia - której nie ma w tabeli - wzrosła do prawie 300ms.

Na podstawie obu tabel S1 oraz S2 widać niemalże całkowitą niezawodność jaką dawało umieszczenie stanu gry w pamięci, gdyż dopiero uruchomienie 100-u wątków spowodowało pojawienie się jakiegokolwiek odsetku błędów. Analizując analogiczne testy 1-6 (bez dodatkowego) widać, że czas odpowiedzi (mediana) w przypadku trzymania danych w pliku wynosił kilka razy więcej (od 2 do 8) niż w przypadku trzymania danych w pamięci. Tyczy się to zarówno pierwszego end-point'u (GET), jak i drugiego (POST).

Dość nietypowo zachowały się testy 4-6 w stosunku do testów 1-3 (czyli dwukrotnie dłuższy czas testów), gdyż mimo, że zgodnie z przewidywaniami czas odpowiedzi był znacznie krótszy (niemalże dwa razy), to odsetek błędów nie zmalał, a nawet urosł w przypadku tabeli S2 (31.73 → 34.40 i 32.20 → 33.40)

Co do samego odsetku błędów to dla sposobu przechowywania danych w pliku i testów 1-6 (bez dodatkowego) widać - oprócz oczywistego powiązania z ilością wątków - powiązanie z przepustowością. Serwer ma pewną granicę, do której odsetek błędów ciągle wynosi 0% i są to okolice wartości 63.6/sek, co widać w tabeli S1 dla testu 4-go. Zaś jeśli spojrzeć na tabelę S2 to dla testu 4-go przepustowość wynosiła niewiele więcej i pojawił się nieznaczny odsetek błędów. Można się tu dopatrzeć pewnej zależności liniowej między przepustowością i odsetkiem błędów, o ile znajdzie się tę granicę, dla której pojawia się odsetek błędów. Zakładając, że to 63.6/sek jest tą granicą to należy teraz wziąć na przykład odsetek błędów wraz z przepustowością z testu z 1 z tabeli S1. Różnicą w przepustowości to 98.5-63.6, czyli 34.9/sek i dla tej różnicy odsetek błędów wzrósł o 9.60%, co powinno dawać wzrost odsetku błędów o około 2.75% na każde 10/sek przepustowości (lecz począwszy dopiero od 63.6/sek przepustowości). Patrząc teraz na pozostałe przepustowości dla każdego z testów można spróbować policzyć przewidywany

odsetek błędów, sprawdzić jak ma się to do rzeczywistych danych z tabeli i przekonać się czy zachodzi tutaj wcześniej wspomniana zależność:

Tabela [S1](#), test 2 -  $(157.1 - 63.6)/10 \cdot 2.75 = 25.71$  w porównaniu do rzeczywistego 30.13

Tabela [S1](#), test 3 -  $(163.6 - 63.6)/10 \cdot 2.75 = 27.50$  w porównaniu do rzeczywistego 31.53

Tabela [S1](#), test 5 -  $(145.4 - 63.6)/10 \cdot 2.75 = 22.50$  w porównaniu do rzeczywistego 29.73

Tabela [S1](#), test 6 -  $(159.5 - 63.6)/10 \cdot 2.75 = 26.37$  w porównaniu do rzeczywistego 29.60

Tabela [S2](#), test 1 -  $(115.2 - 63.6)/10 \cdot 2.75 = 14.19$  w porównaniu do rzeczywistego 15.20

Jak widać, wartości te nie różnią się znacznie i można wysnuć tezę o liniowym powiązaniu między odsetkiem błędów i przepustowością.

Analizując screenshot'y [70](#) oraz [69](#) widać, że metoda POST wysyła znacznie więcej danych na serwer, a metoda GET pobiera znacznie więcej (o wiele więcej).