

Testy wydajnościowe projektu

Mateusz Król

June 18, 2018

1 Wprowadzenie

Przy budowie każdej aplikacji przychodzi moment na jej testy. Na zajęciach napisano taki właśnie projekt, który ma pomagać w Narodowym Forum Muzyki we Wrocławiu przy rejestracji pracowników na wydarzenia

Aplikację poddawano testom lokalnie na serwerze stawianym przez symfony i bazie danych MySQL, do komunikacji z którą używano Doctrine. Testy skupiały się na kilku aspektach:

1. Obciążanie serwera transakcjami generowanymi dla jednego adresu:

a) bez zapytań SQL

b) z zapytaniami typu Select, zawierającymi JOIN

2. Obciążanie serwera transakcjami generowanymi przez wielu użytkowników dla kilku podstron

a) bez zapytań SQL

b) z zapytaniami SQL typu Select, zawierającymi JOIN

Następnie interpretowano dane i wnioskowano na ich podstawie, czy aplikacja będzie w stanie zapewnić płynność wymiany danych nawet przy wysokim obciążeniu.

3. Obciążanie serwera transakcjami generowanymi przez wielu użytkowników dla jednej/kilku podstron, zawierającymi zapytania SQL po próbach ich uproszczenia.

Następnie podjęto testy wydajnościowe na rzeczywistym obiekcie, tj. aplikacji webowej NFM, której bliźniaczą kopię postawiono na innym koncie hostingowym.

2 Opis narzędzia oraz zakresu

2.1 Narzędzie

Jako narzędzie symulujący ruch na serwerze użyto Siege w wersji 3.0.8

2.2 Metodologia testów

1) Stress test - będziemy bombardować naszą aplikację transakcjami, aż do momentu załamania się

2) regression tests - sprawdzimy wpływ optymalizacji zapytań SQL na wydajność pracy aplikacji

2.3 Informacje dodatkowe

1. Testy przeprowadzano w maksymalnie izolowanym środowisku. Pozamykano wszystkie procesy poboczne.

2. Jako transakcję rozumiemy otwarcie socket-u dla klienta, realizację zapytania a następnie transfer danych i zamknięcie socket-u po zakończeniu przesyłu. Dopiero po spełnieniu wszystkich tych kroków mówimy, że transakcja jest kompletna.

3 Testy

3.1 brak zapytań SQL oraz jeden URL

W pierwszej części testów postanowiono poddać próbie serwer witryny, jednocześnie nie zadając mu żadnych poleceń SQL. Wyniki zebrano w tabeli oraz przedstawiono na wykresach:

Objaśnienia symboli zawartych w tabelach:

Availability - procent socketów przetworzonych przez serwer podzielone przez wszystkie próby połączenia z serwerem

Elapsed Time - Czas trwania całego testu

Data transfered - łączna pojemność przesłanych informacji (zawiera nagłówki oraz zawartość strony)

Response Time - średni czas jaki każdy z klientów musiał czekać na obsługę swojej transakcji

Transaction rate - średnia ilość transakcji jakie serwer wykonywał w przeciągu sekundy

Throughput - średni transfer wysyłowy z serwera do klientów

Concurrency - średnia liczba jednoczesnych połączeń. Liczba wzrasta wraz ze spadkiem wydajności serwera

Successful transactions - liczba pozytywnie zakończonych transakcji

Failed transactions - liczba nieukończonych, zakończonych błędem transakcji (błędy - przekroczenie czasu 30s oczekiwania na odpowiedź, reset połączenia)

Longest transaction - Czas jaki musiał upłynąć aby najdłuższa transakcja zakończyła się sukcesem.

Opis							
Klientów	50	100	200	300	400	500	600
Availability	100	100	100	100	100	100	100
Elapsed time	10.09	10.13	10.34	10.36	11.43	13.62	16.23
Data transferred	2.07	4.14	8.27	12.32	16.55	20.69	24.82
Response time	0.08	0.14	0.25	0.35	0.79	3.00	2.79
Transaction rate	4.96	9.88	19.35	28.96	35.29	32.13	37.83
Throughput	0.21	0.41	0.80	1.11	1.28	1.33	1.41
Concurrency	0.45	4.35	4.85	10.23	27.15	87.05	101.76
Successful transactions	50	100	200	300	400	500	600
Failed Transaction	0	0	0	0	0	0	0
Longest Transaction	0.19	0.34	0.62	0.89	2.11	6.58	8.77

Opis							
Klientów	700	800	850	900	1000	1100	1200
Availability	100	100	99.41	98.85	97.93	96.39	95.28
Elapsed time	18	20.31	22.43	23.25	28.43	39.05	41.72
Data transferred	28.96	33.10	29.30	36.87	40.14	43.87	47.30
Response time	5.58	4.97	4.95	6.62	7.94	10.63	12.87
Transaction rate	39.58	39.70	37.78	38.39	34.68	27.17	27.56
Throughput	1.45	1.46	1.59	1.64	1.43	1.13	0.91
Concurrency	187.82	195.24	186.94	254.39	274.39	288.60	355.87
Successful transactions	700	800	847	891.33	970.33	1060.33	1143.33
Failed Transaction	0	0	3	8.67	29.67	39.67	56.67
Longest Transaction	11.06	13.10	16.37	18.33	24.86	32.32	35.03

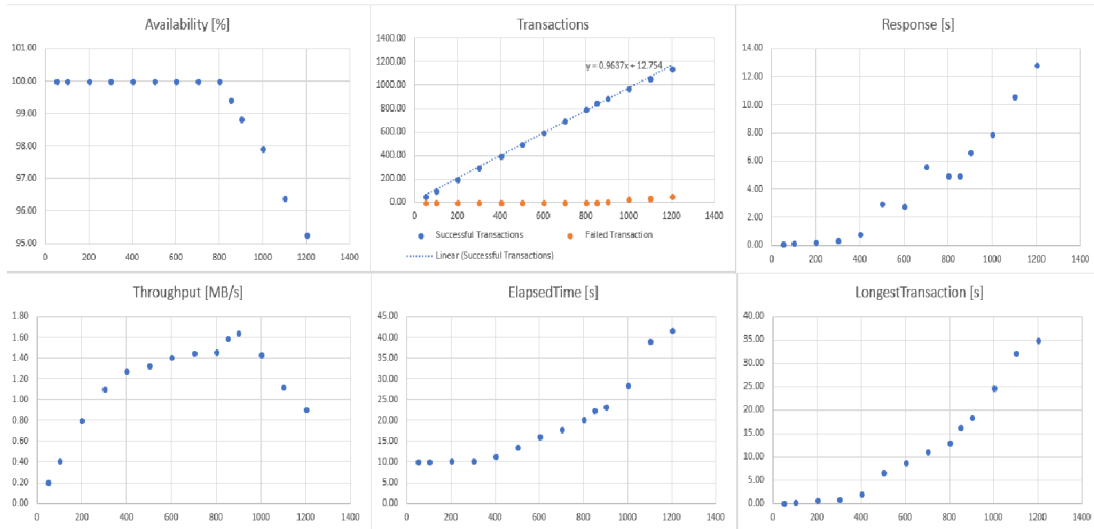


Figure 1: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

Wnioski i spostrzeżenia:

- Wydajność serwera jest bardzo dobra, ponieważ pierwsze nieudane transakcje pojawiają się dopiero przy 900 użytkownikach a do 1200 użytkowników wydajność serwera nie spadła poniżej 95
- wszystkie zależności czasowe wzrastają zależnością zbliżoną do kwadratu swojego argumentu
- bardzo ciekawa zależność uwidoczniła się na wykresie mówiącym o przepustowości serwera [Mb/s] widzimy, że przy 900 userach serwer pracuje z największą wydajnością, dalej przepustowość serwera spada. Jednocześnie po przekroczeniu 900 klientów pojawiły się pierwsze nieudane transakcje, co wskazuje na silną korelację między liczbą nieudanych transakcji a przepustowością serwera.
- Nie udało się przeprowadzić testów dla większej ilości Klientów niż 1200, ponieważ zasoby komputera nie pozwalały na otwarcie wystarczającej ilości plików do symulowania więcej niż 1200 klientów.

3.2 Zapytania SQL (z JOIN) oraz jeden URL

W poprzedniej części badania wykazały, że nasz serwer jest w stanie poradzić sobie z dość wzmorzonym ruchem, jednak nie wymagaliśmy od niego zbyt wiele pracy, ponieważ jedynie ładowaliśmy stronę powitalną, która nie wymagała połączenia z bazą danych.

W tej części spróbujemy załadować podstronę, która zawiera najbardziej złożone zapytanie SQL występujące w serwisie tj. Select z łączaniem 3 tabel wraz z sortowaniem i liczeniem (liczono wydarzenia w danym miesiącu). Wyniki zebrano w tabeli oraz zaprezentowano na wykresach:

Opis	10	50	100	150	200	250	300	350	400
Klientów	10	50	100	150	200	250	300	350	400
Availability	100	100	100	100	100	100	100	100	98.17
Elapsed time	8.87	10.14	12.93	18.37	27.57	30.55	36.78	42.95	46.38
Data transferred	9.41	47.04	94.08	141.15	188.16	235.19	282.25	329.27	369.48
Response time	0.18	0.43	1.58	4.66	7.05	10.50	13.45	16.67	18.47
Transaction rate	1.13	4.93	7.74	8.17	8.25	8.18	8.15	8.16	8.37
Throughput	1.06	4.63	7.28	7.68	7.77	7.67	7.67	7.68	7.88
Concurrency	0.21	2.10	12.23	38.03	58.17	89.57	109.72	135.86	154.39
Successful transactions	10	50	100	150	200	250	300	350	359.33
Failed Transaction	0	0	0	0	0	0	0	0	7.33
Longest Transaction	0.35	1.13	3.26	8.44	14.22	24.21	29.40	35.95	39.05

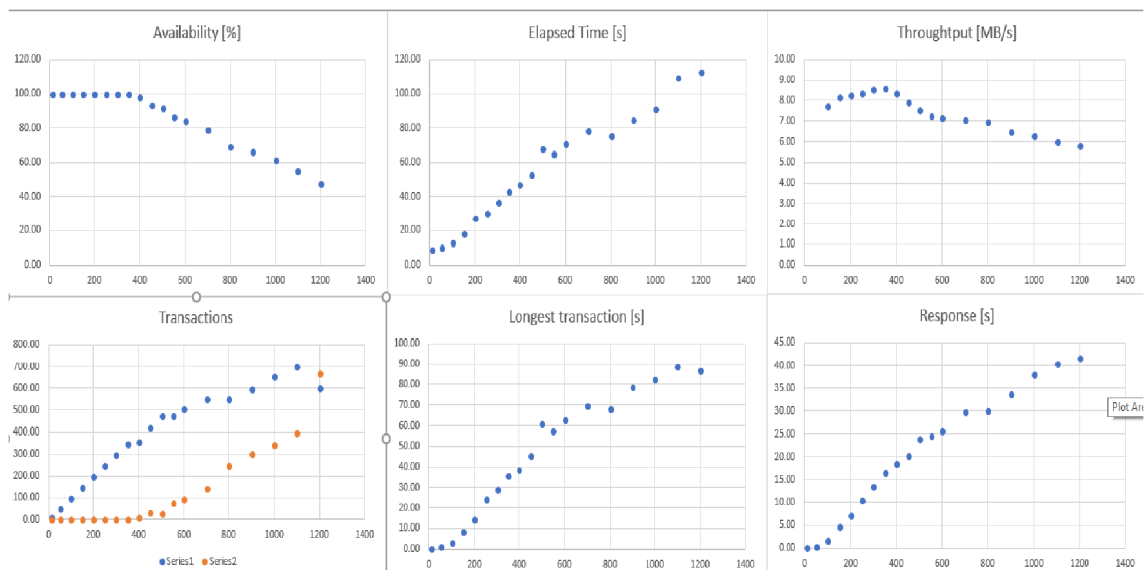


Figure 2: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

Opis	450	500	550	600	700	800	900	1000	1100	1200
Klientów	450	500	550	600	700	800	900	1000	1100	1200
Availability	93.27	91.54	86.33	84.17	79	69.17	66.45	61.54	54.98	47.59
Elapsed time	52.91	67.93	95.02	70.6	78.40	75.47	84.76	90.88	109.33	112.98
Data transferred	394.81	444.36	446.46	475.07	514.25	520.56	561.92	616.89	660.09	563.84
Response time	20.38	23.91	24.71	25.74	29.85	29.85	29.95	33.79	38.01	40.45
Transaction rate	7.94	7.54	7.25	7.16	7.06	6.97	6.50	6.28	6.01	5.82
Throughput	7.47	6.62	6.98	6.73	6.64	6.92	6.56	6.85	6.19	6.07
Concurrency	161.87	168.14	182.24	191.17	211.21	220.20	235.76	318.24	272.63	841.16
Successful transactions	419	472	474	505	553	553	598	655	701	599
Failed Transaction	30	27	75	95	147	246	302	344	398	667
Longest Transaction	45.31	60.98	57.63	62.99	96.73	67.98	78.67	82.86	88.90	87.08

Spostrzeżenia i wnioski:

- Widać zauważalny wzrost nieudanych tranzakcji w porównaniu do poprzedniego zestawienia, stąd wniosek, że zapytania SQL mają duży wpływ na powodzenia transakcji.
- Pozostałe zależności zachowują się bardzo podobnie do poprzednich zestawień
- Również i tym razem jesteśmy w stanie zauważyć, w którym momencie serwer przestaje zwiększać swoją wydajność przepustową. Znowu powodowane jest to wzrostem transakcji zakończonych niepowodzeniami.
- Na wykresie przedstawiającym transakcję udane i nieudane widzimy punkt załamania równy około 1200 klientów. Przy takim obciążeniu serwera mamy więcej nieudanych transakcji niż udanych.

3.3 Dodatkowe badanie

W poprzednim zestawieniu udało nam się znaleźć punkt załamania się serwisu oraz punkt, w którym liczba nieudanych transakcji przewyższyła te zakończone powodzeniem. W tym teście badano jak zachowa się krzywa nieudanych transakcji po przewyższeniu krzywej przedstawiającej udane transakcje w dalszych etapach. Bazując na wcześniejszych obserwacjach zwiększymy komplikacje zapytania SQL, tak aby zwracało całą bazę przez łączenia tabel, powinno to spowodować wzrost nieudanych transakcji. Wyniki przedstawia wykres:

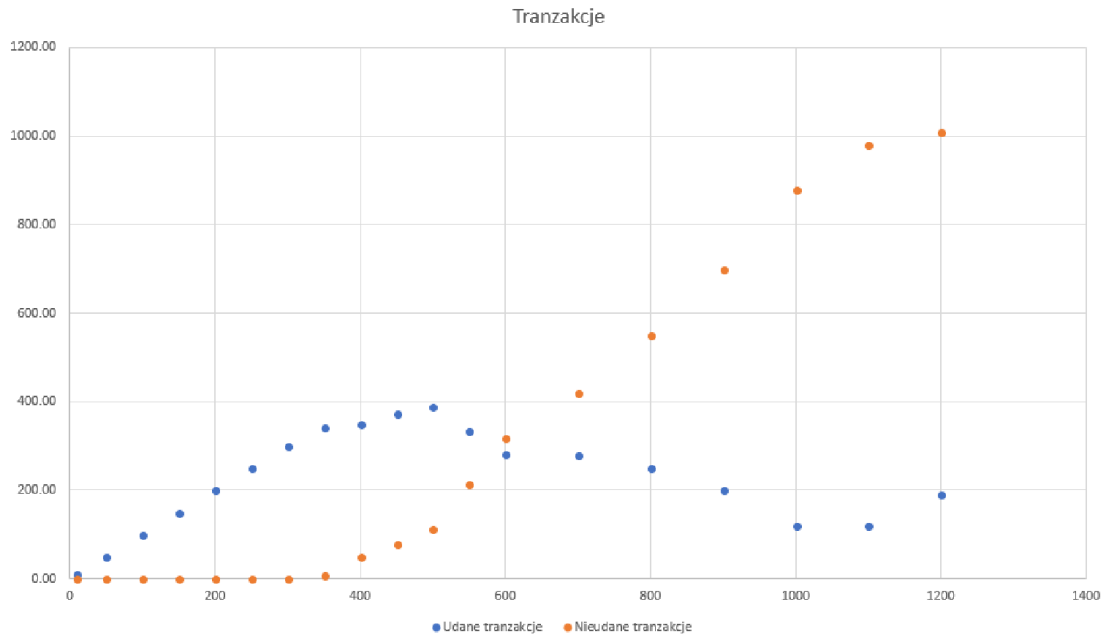


Figure 3: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

Wnioski i spostrzeżenia:

- Widzimy, że wykres kształtem może przypominać parabolę, stąd widzimy, że wzrost klientów będzie niósł za sobą duże utraty powodzeń w transakcjach
- po tym badaniu widzimy, jak duży wpływ może mieć nieprzemyślane korzystanie z zasobów przechowywanych w bazie danych.

3.4 Analiza wymagań projektu

Musimy na chwilę zatrzymać się w testach wydajnościowych, aby zastanowić się nad kolejnym aspektem.

W poprzedniej części bombardowaliśmy jeden URL wieloma użytkownikami, jednak w rzeczywistości rzadko spotykamy się z sytuacją, kiedy wszyscy użytkownicy przeglądają jedną podstronę naszego serwisu. Dlatego w tej części skupimy się na podziale obciążeń na poszczególne podstrony. Poniżej zestawiono prognozę obciążenia głównych podstron serwisu:

- Zapisy 0.2
- Dodawanie wydarzeń 0.1
- Przegląd wydarzeń 0.5
- Logowanie 0.1
- Sprawdzanie obecności 0.1

Obciążenia rozłożono według schematu każdy użytkownik na 10 połączeń 5x przegląda wydarzenia, 2 razy odwiedza zapisy itd.

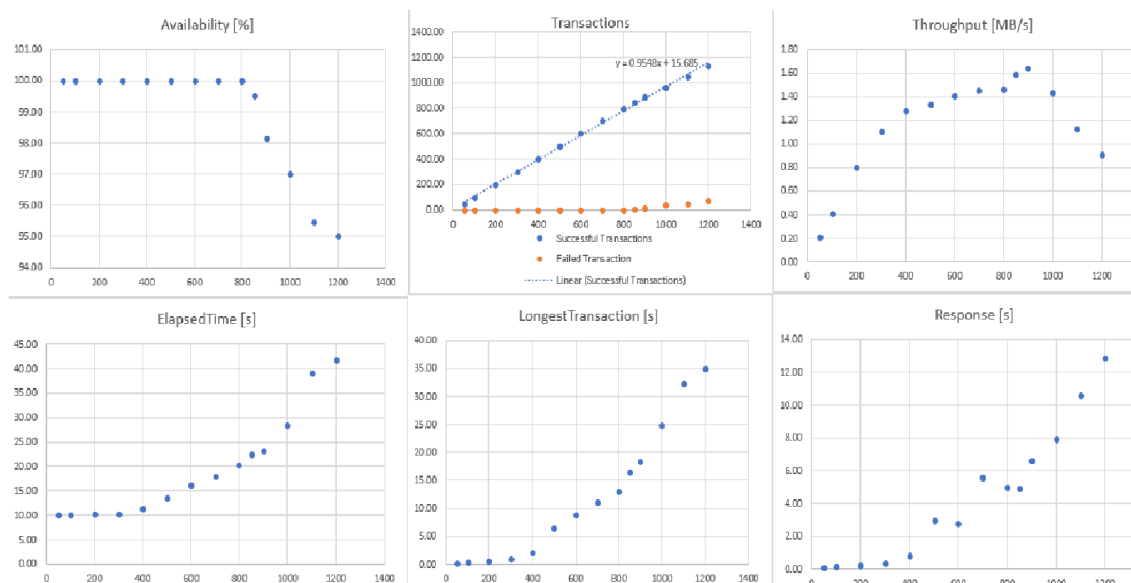


Figure 4: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

3.5 Brak zapytań SQL oraz wiele URL

Opis	50	100	200	300	400	500	600
Klientów	50	100	200	300	400	500	600
Availability	100	100	100	100	100	100	100
Elapsed time	10.24	10.33	10.54	10.66	11.23	13.82	16.83
Data transferred	2.26	4.85	8.47	11.52	16.84	21.19	25.12
Response time	0.08	0.14	0.25	0.35	0.79	3.00	2.79
Transaction rate	4.96	9.88	19.35	28.96	35.29	32.13	37.83
Throughput	0.21	0.41	0.80	1.11	1.28	1.33	1.41
Concurrency	0.45	4.35	4.85	10.23	27.15	87.05	101.76
Successful transactions	50	100	200	300	400	500	600
Failed Transaction	0	0	0	0	0	0	0
Longest Transaction	0.19	0.34	0.62	0.89	2.11	6.58	8.77

Opis	700	800	850	900	1000	1100	1200
Klientów	700	800	850	900	1000	1100	1200
Availability	100	100	99.41	98.85	97.93	96.39	95.28
Elapsed time	18	20.31	22.43	23.25	28.43	39.05	41.72
Data transferred	28.96	33.10	29.30	36.87	40.14	43.87	47.30
Response time	5.58	4.97	4.95	6.62	7.94	10.63	12.87
Transaction rate	39.58	39.70	37.78	38.39	34.68	27.17	27.56
Throughput	1.45	1.46	1.59	1.64	1.43	1.13	0.91
Concurrency	187.82	195.24	186.94	254.39	274.39	288.60	355.87
Successful transactions	700	800	847	891.33	970.33	1060.33	1143.33
Failed Transaction	0	0	3	8.67	29.67	39.67	56.67
Longest Transaction	11.06	13.10	16.37	18.33	24.86	32.32	35.03

Wniski i spostrzeżenia:

- Obsługa wielu adresów nie miała dużego wpływu na obniżenie wydajności serwisu. Zanotowano jedynie niewielki spadek, może być to spowodowane ładowaniem obrazków na stronie z zapisami, czy dużej ilości div-ów w wydarzeniach

3.6 Wydajność dla zapytań SQL (JOIN) dla wielu URL

W tej części badano zachowanie serwisu w przypadku dużego obciążenia na różnych podstronach. W przypadku strony do logowania nie było powiązań z bazą danych. Na stronie ładującej wydarzenie zostawiono zagnieżdżone JOIN-y. Na pozostałych podstronach zachowano rzeczywisty układ serwisu.

Opis	10	50	100	150	200	250	300	350	400
Klientów	100	100	100	100	100	100	99,58	86,25	80
Availability	8.75	10.24	12.65	18.65	27.65	30.56	36.45	42.92	47.01
Elapsed time	9,31	48,65	94,89	142,16	189,56	235,95	284,96	329,65	369,56
Data transferred	0,18	0,56	1,89	4,98	7,56	10,65	13,87	16,98	18,95
Response time	1.06	4,63	6,56	6,89	7,10	7,40	7,85	8,61	8,37
Transaction rate	1.06	4.63	7.28	7.68	7.77	7.70	7.67	7.67	7.88
Throughput	0.21	2.15	12,65	40,58	59,65	89,98	109,65	135,96	156,90
Concurrency	10	50	100	150	200	250	300	350	345
Successful transactions	0	0	0	0	0	0	0	0	55
Failed Transaction	0.45	0.87	1.35	1.65	2.56	2.89	3.2	3.15	4.65
Longest Transaction									

Opis	450	500	550	600	700	800	900	1000	1100	1200
Klientów	80	76	70.91	68.33	60	56.25	52.78	49	46.55	44.17
Availability	52.34	68.65	65.23	70.26	79.50	79.15	84.52	90.32	1009.63	125.65
Elapsed time	394.32	444.36	446.85	478.65	514.32	520.68	520.66	651.84	561.33	616.75
Data transferred	20.46	24.12	24.62	25.45	30.21	33.41	38.24	40.45	41.22	42.66
Response time	7.94	6.62	6.49	6.31	6.42	6.23	6.12	6.01	5.98	5.81
Transaction rate	7.47	6.62	6.98	6.73	6.64	6.92	6.56	6.85	6.19	6.07
Throughput	162.56	169.32	181.32	191.35	214.26	224.55	238.46	318.65	575.23	530.12
Concurrency	360	380	690	410	420	451	778	496	515	532
Successful transactions	90	120	160	184	279	349	425	510	685	858
Failed Transaction	4.89	5.34	5.26	6.14	7.59	8.64	9.12	10.35	11.21	12.56
Longest Transaction										

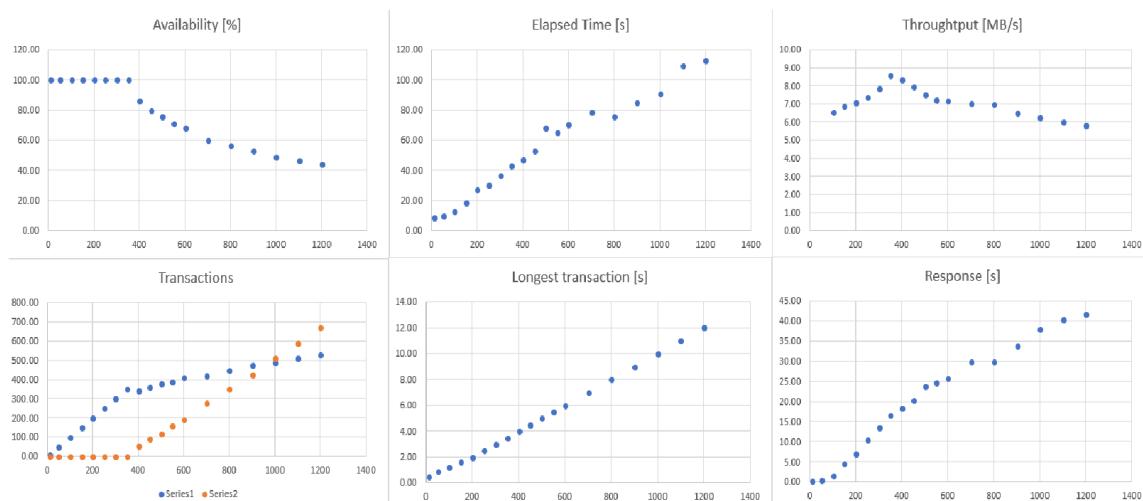


Figure 5: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

Wnioski i spostrzeżenia:

- Zapytania do wielu podstron okazały się bardziej uciążliwe dla serwera niż bombardowanie jednej podstrony
- Połowa zapytań i tak trafiała do strony, która posiadała bardzo zagnieżdżone zapytanie SQL

3.7 Optymalizacja zapytania SQL

Zgodnie z poprzednimi spostrzeżeniami składnia zapytania SQL zdaje się mieć duży wpływ na wymianę danych pomiędzy serwerem a klientem. W tym momencie spróbujemy zoptymalizować zapytanie SQL, tak, aby nie zawierało JOIN-ów a zastosowania podzapytania, nie będziemy wybierać wszystkich informacji, a jedynie te potrzebne do wyświetlenia, kolejnym rozsądnym pomysłem będzie ograniczenie wyświetlanych wydarzeń, ponieważ i tak dodawane są chronologicznie. W dalszej części badań skupimy się jedynie na udanych/nieudanych transakcjach, ponieważ to właśnie ten parametr zachowywał się najmniej przewidywalnie w poprzednich zestawieniach.

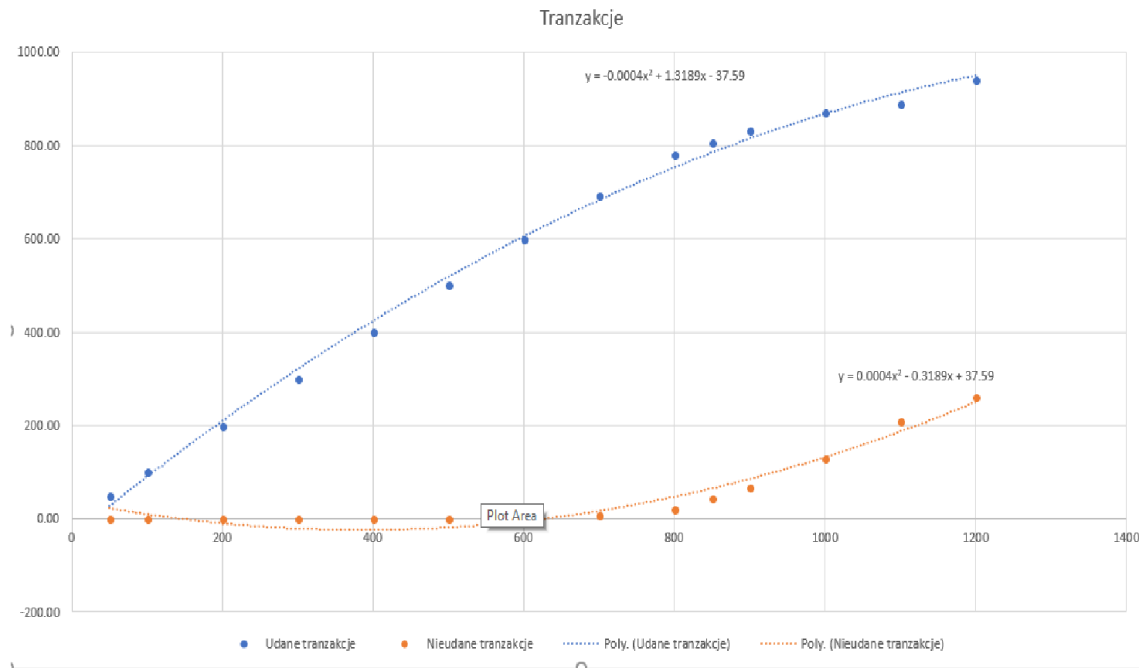


Figure 6: Prezentacja podstawowych danych po obciążeniu lokalnego serwera

Wnioski i Spostrzeżenia:

- Optymalizacje zapytania SQL wykazuje ogromną zmianę w obciążeniu serwera. Udało się wyeliminować punkt przecięcia się krzywych reprezentujących transakcje pomyślne z niepomyślnymi..
- Aproksymacja wykresu jednak pokazuje, że problem i tak narasta wielomianowo i wykresy przecięłyby się w okolicy 2000 klientów. Jednak jak na jednowątkową obsługę zapytań to wyniki są bardzo powytywne

3.8 Testy na rzeczywistym serwerze

Jak mogliśmy zauważyć testy aplikacji lokalnie wypadły całkiem dobrze. Następnym etapem będą testy aplikacji, która jest postawiona na serwerze 000webhost.com. Aplikacja dla NFM-u również działa na tym hostingu.

Testy muszą niestety mieć dużo prostrzą formę ponieważ hosting jest darmowy, co niesie za sobą spore ograniczenia w ruchu.

Możemy wysłać jedynie 500 zapytań na serwer w ciągu minuty.

Ograniczona jest również liczba zapytań do baz danych w ciągu doby, dlatego testy przybiorą trochę inną formę. Postaramy się na maxa wykorzystać przydzielone nam zasoby i zobaczyć czy serwer jest w pełni responsywny.

Test 1:

Symulujemy 1200 użytkowników próbujących połączyć się z podstroną zawierającą zapytanie SQL do wyświetlenia wydarzeń. Wyniki:

Availability: 93.67

Elapsed Time: 38.36s

Data transmitted: 20.98Mb

Response Time: 14.46s

Transaction Rate: 29.30 trans/sec

Throughput: 0.5Mb/s Concurrency: 258.36

Successful Transactions: 1124

Failed Transaction: 76

Longest Transaction: 24.96

Shortest Transaction: 0.37

Wnioski: Serwer wyrzucił kilka błędów 502 oraz kilka razy połączenie przedawniło się, ale zdecydowanie zdał testy wydajnościowe. Aplikacja nie była pisana pod wielki ruch na serwerze. Ma ona jedynie służyć małej grupie ludzi (około 100)

Test 2:

Obciążymy teraz serwer zgodnie z wcześniejszą analizą rozłożenia tuchu na poszczególnych podstronach. Wyniki:

Availability: 97.37

Elapsed Time: 36.48s

Data transmitted: 15.58Mb

Response Time: 16.16s

Transaction Rate: 31.30 trans/sec

Throughput: 0.4Mb/s Concurrency: 198.36

Successful Transactions: 1168

Failed Transaction: 32

Longest Transaction: 21.21

Shortest Transaction: 0.17

Wnioski: Serwer w rzeczywistym obciążeniu spisuje się nawet lepiej, wynika to prawdopodobnie, z tego, że nie na wszystkich stronach wymaga się komunikacji z bazą danych.

4 References

<https://dev.mysql.com/doc/> - oficjalna dokumentacja MySQL

<http://www.mif.pg.gda.pl/homepages/mate/bazydanych/transakcje.html> - opis transakcji w DB

<http://roman.ptak.staff.iiar.pwr.wroc.pl/BDwykladnr3ver5.pdf> - wykład dotyczący MySQL Dr. Ptaka

<https://web.stanford.edu/class/cs245/homeworks/b+tree/MySQLInnoDB.B+Tree.pdf> - opis struktur danych stosowanych w bazach danych