



Modelowanie QPME i system webowy

Tomasz RAK

Katedra Informatyki i Automatyki
Politechnika Rzeszowska

2 kwietnia 2014

Modelowanie i nie tylko...



Agenda

- Modelowanie
- System rzeczywisty
- Testy systemu webowego
- Symulacje z użyciem QPME

Motywacja

Nie można ciągle dodawać nowych urządzeń, w celu zwiększenia wydajności.^a

^aSamolej, S., Szmuc, T.: HTCPNs-based modelling and evaluation of dynamic computer cluster reconfiguration. 7054 LNCS, 97–108 (2012) doi:10.1007/978-3-642-28038-2_8

Zatwarnicki, K., Zatwarnicka, A.: Modeling Operation of Web Service. CCIS 291, 384–393. (2013) doi:10.1007/978-3-642-38865-1_39

Nie każdy system można/wolno zwirtualizować lub umieścić w chmurze (IaaS, PaaS, SaaS).

System komputerowy a model wydajnościowy



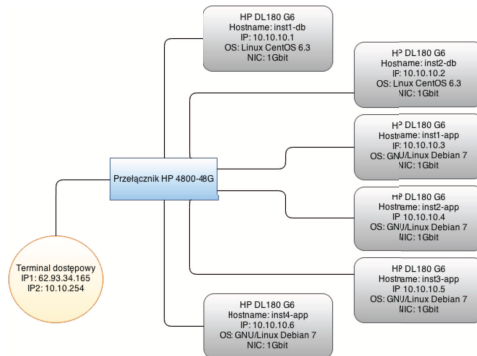
Agenda

- Modelowanie - teoria
- System rzeczywisty \leftrightarrow 6
- Testy systemu webowego
- Symulacje z użyciem QPME

Agenda

- Modelowanie
- System rzeczywisty - DayTrader
- Testy systemu webowego ↔ 13
- Symulacje QPME

Schemat budowy środowiska testowego (HP ProLiant DL180 G6)



```

[root@inst1-db ~]# cat /proc/cpuinfo | grep 'model name' | head -n 1
model name      : Intel(R) Xeon(TM) CPU           E5520  @ 2.27GHz
[root@inst1-db ~]# cat /proc/cpuinfo | grep 'cpu cores' | head -n 1
cpu cores       : 4
[root@inst1-db ~]# cat /proc/meminfo | grep 'MemTotal'
MemTotal:       8182276 kB
[root@inst1-db ~]#
  
```


Serwery

Nazwa hosta	Adres IP	System operacyjny	Zadania
client	10.10.10.1	CentOS6.3 x64	Generator ruchu
inst-db	10.10.10.2	CentOS6.3 x64	Baza danych
inst1-app	10.10.10.3	Debian7 x64	Węzeł i Load balancer Węzeł Węzeł Węzeł Terminal
inst2-app	10.10.10.4	Debian7 x64	
inst3-app	10.10.10.5	Debian7 x64	
inst4-app	10.10.10.6	Debian7 x64	
	10.10.10.254	CentOS 6.3 x64	

- Pula wątków serwera aplikacji
- Pula połączeń do bazy danych

Etapy przygotowania

Instalacja i konfiguracja:

- bazy danych Oracle 11g oraz utworzenie instancji bazy
- serwera GlassFish 3.1 Open Source Edition
- load balancera (Apache Tomcat Connector)

Aplikacje:

- DayTrader
- „Klient”

Trade Performance Benchmark Sample

- Część projektu Apache Geronimo od 2012r.
- Bazuje na komponentach technologii Java EE 6
- DayTrader pozwala na: założenie konta w systemie, logowanie, zarządzanie kontem, przeglądania posiadanych akcji, przeglądanie notowań akcji, kupowanie oraz sprzedawanie
- Może zostać uruchomiony na każdym serwerze aplikacyjnym (?)

The screenshot displays the 'Trade Home' page for a user named 'Welcome sld-0'. The page includes a navigation menu on the left with links for Home, Account, Portfolio, Quotes/Trade, Logout, and Trade. The main content area is divided into several sections:

- User Statistics:**
 - account ID: 2000
 - account created: 2009-12-10 16:31:28.446
 - total login: 1742
 - session created: Wed Apr 14 09:35:09 CEST 2010
- Account Summary:**
 - cash balance: \$ 871989.50
 - number of holdings: 39
 - total of holdings: \$ 127761.00
 - net of: \$ 999750.50
 - cash/holding: \$ 1000000.00
 - opening balance:
 - current gain(loss): \$ -249.204 (+0.00%)
- Market Summary:**
 - Market Summary: 2010-04-14
 - Trade Stock Index (7333): 90.69 (+0.06%)
 - Index Values: 24556.0
- Top Gains:**

Symbol	Gain	Change
ALCO	1896.0	▲
ALCO	200.0	▲
ALCO	200.0	▲
ALCO	200.0	▲
ALCO	100.0	▲
- Top Losses:**

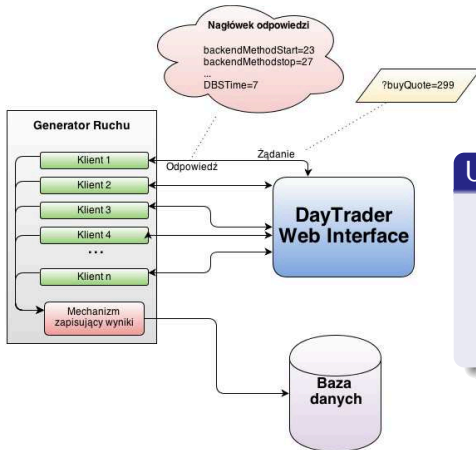
Symbol	Loss	Change
ALCO	100.0	▼
ALCO	100.0	▼
ALCO	100.0	▼
ALCO	100.0	▼
ALCO	100.0	▼

Poprawki i modyfikacje

- Poprawiono współpracę z serwerem aplikacji
- Poprawiono błędy w zarządzaniu zasobami bazy danych
- Poprawiono problemy w generowaniu unikalnych identyfikatorów
- Zmiany dotyczące zbierania informacji na temat wykonywanego żądania¹

¹Internetowe środowisko produkcyjne na bazie klastra serwerów - Artur Guzik >

Parametry zapytania



Udział procentowy

<i>Login / Logout</i>	→	4 [%]
<i>GetHome</i>	→	20 [%]
<i>GetPortfolio</i>	→	12 [%]
<i>ShowQuote</i>	→	40 [%]
<i>BuyQuote</i>	→	5 [%]
<i>SellQuote</i>	→	5 [%]
<i>ShowAccount</i>	→	10 [%]
<i>UpdateProfile</i>	→	4 [%]

Agenda

- Modelowanie
- System rzeczywisty
- Testy systemu webowego - działanie
- Symulacje z użyciem QPME ↔ 31

Rodzaje testów

Wydajność systemu w zależności od:

- obciążenia
- rozmiaru puli wątków serwera
- zmiennej liczby elementów klastra

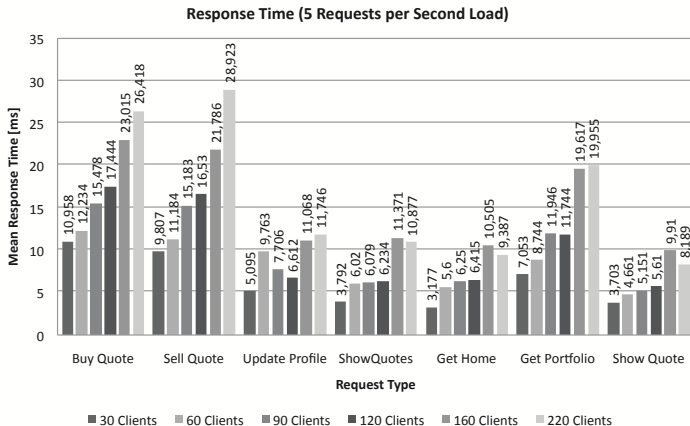
Cel

- Wyznaczenie czasu obsługi (service demand)
- Późniejsza weryfikacja wyników symulacji

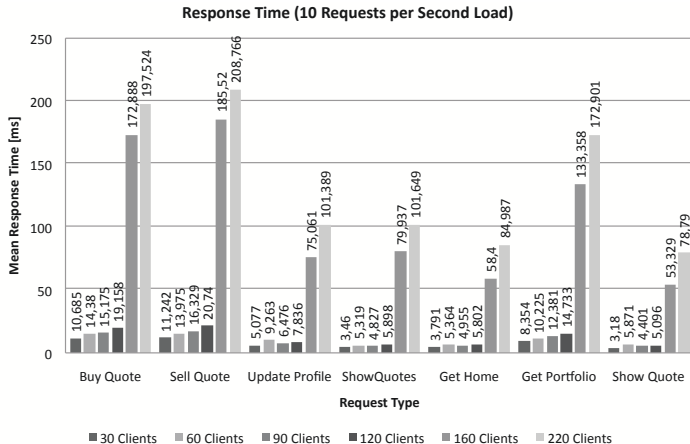
Parametry wejściowe

Serwer/Parametry	Test 1	Test 2	Test 3	Test 4
Sprzęt				
Generator ruchu	10.10.10.1	10.10.10.1	10.10.10.1	10.10.10.1
Klaster	Nie	Nie	Nie	Nie
Load balancer	Brak	Brak	Brak	Brak
GlassFish	10.10.10.3	10.10.10.3	10.10.10.3	10.10.10.3
Baza danych	10.10.10.2	10.10.10.2	10.10.10.2	10.10.10.2
Systemowe				
Pula aktywnych wątków	120	120	120	120
Pula aktywnych połączeń z bazą	120	120	120	120
Obciążenie				
Liczba zapytań na sekundę [zapytań/s]	5	10	15	20
Generator ruchu				
Liczba aktywnych klientów	30 60 90 120 160 220	30 60 90 120 160 220	30 60 90 120 160 220	30 60 90 120 160 220
Liczba grup	1	1	1	1
Czas pomiędzy grupami [s]	300	300	300	300

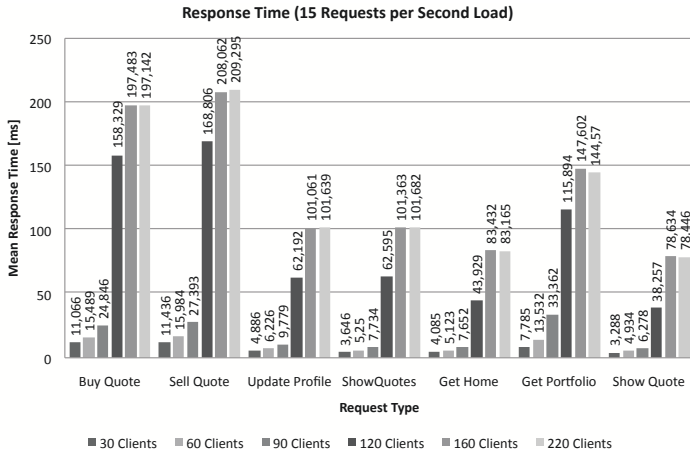
Test 1 – Obciążenie 5 [zapytań/s] : Response Time



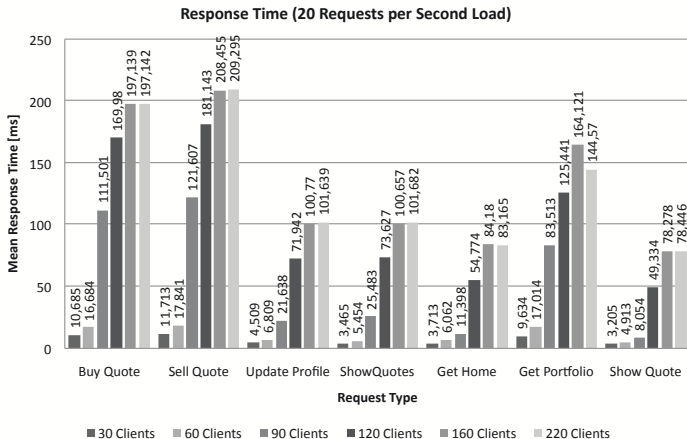
Test 2 – Obciążenie 10 [zapytań/s] : Response Time



Test 3 – Obciążenie 15 [zapytań/s] : Response Time



Test 4 – Obciążenie 20 [zapytań/s] : Response Time

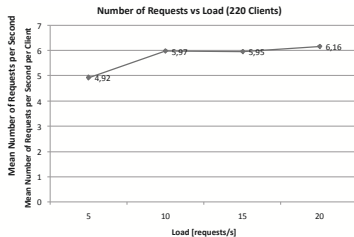
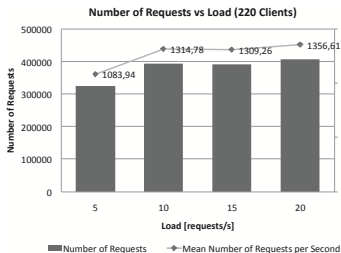


Średni czas wykonania zapytania dla 20 [zapytań/s]

	Buy Quote	Sell Quote	Update Profile	Show Quotes	Get Home	Get Portfolio	Show Profile
30 klientów							
System [ms]	13,120	14,058	6,940	5,903	6,115	12,090	5,680
FE+BE [ms]	10,685	11,713	4,509	3,465	3,713	9,634	3,205
BE [ms]	10,202	11,224	4,229	3,083	3,056	8,733	2,959
220 klientów							
System [ms]	248,222	260,469	153,363	152,973	134,657	195,766	129,568
FE+BE [ms]	197,142	209,295	101,639	101,682	83,165	144,570	78,446
BE [ms]	196,398	208,682	101,386	101,212	79,836	141,070	78,248

Czas spędzony przez zapytania w kolejce!

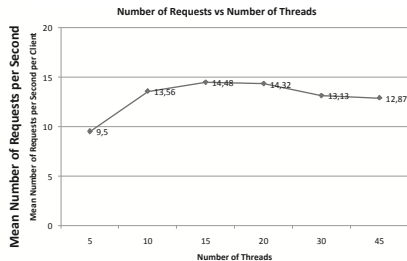
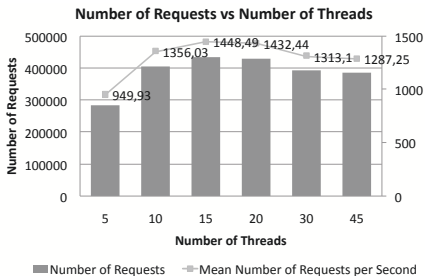
Test 1, 2, 3 i 4 – Liczba zrealizowanych zapytań w stosunku do obciążenia : Number of Requests



Parametry wyjściowe

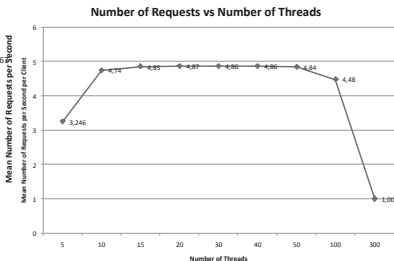
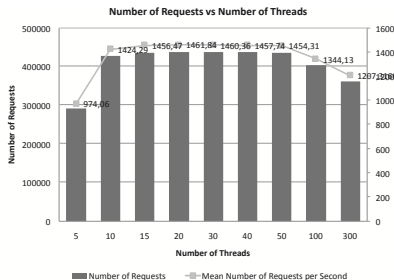
Serwer/Parametry	Test 1	Test 2
Sprzęt		
Generator ruchu	10.10.10.1	10.10.10.1
Klaster	Nie	Nie
Load Balancer	Brak	Brak
GlassFish	10.10.10.3	10.10.10.3
Baza danych	10.10.10.2	10.10.10.2
Systemowe		
Pula aktywnych wątków	5, 10, 15, 20, 30, 45	5, 10, 20, 30, 40, 50, 100, 300
Pula aktywnych połączeń z bazą	20	300
Obciążenie		
Liczba zapytań na sekundę	Nieograniczona	Nieograniczona
Generator ruchu		
Liczba aktywnych klientów	100	300
Liczba grup	10	1
Czas pomiędzy grupami [s]	30	300

Test1 - Zależności między pulą wątków serwera a pulą połączeń do bazy danych : Number of Requests



Średnio w teście (300 [s]) wykonanych zostało ok. 1400 [zapytań/s]!

Test2 - Wpływ liczby wątków na ogólną wydajność systemu : Number of Requests

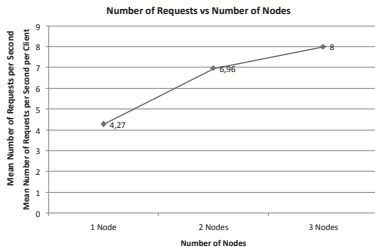
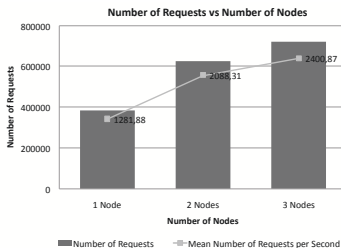


Średnio w teście (300 [s]) wykonanych zostało ok. 1400 [zapytań/s]!

Parametry wejściowe

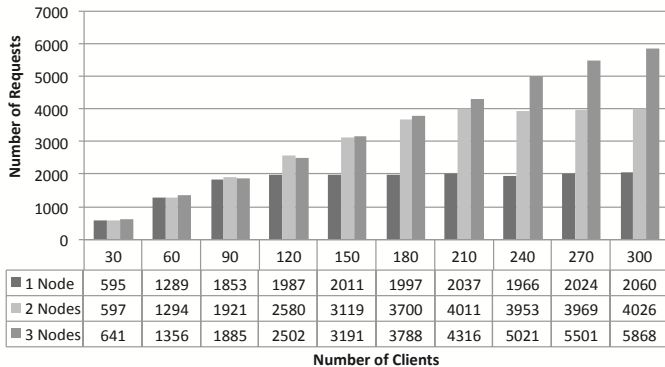
Serwer/Parametry	Test 1 (Buy Quote - Group), 4 (Buy Quote), 7 (All Quotes)	Test 2 (Buy Quote - Group), 5 (Buy Quote), 8 (All Quotes)	Test 3 (Buy Quote - Group), 6 (Buy Quote), 9 (All Quotes)
Sprzętowe			
Generator ruchu	10.10.10.1	10.10.10.1	10.10.10.1
Load Balancer	Brak	10.10.10.3	10.10.10.3
GlassFish	10.10.10.3	DAS: 10.10.10.3, Instancje klastra: 10.10.10.4, 10.10.10.5	DAS: 10.10.10.3, Instancje klastra: 10.10.10.4, 10.10.10.5, 10.10.10.6
Baza danych	10.10.10.2	10.10.10.2	10.10.10.2
Systemowe			
Pula wątków	30	2 x 30	3 x 30
Pula połączeń z bazą	40	2 x 40	3 x 40
Obciążenie (generator ruchu)			
Liczba zapytań [zapytań/s]	15	15	15
Liczba aktywnych klientów	30; 60; 90; 120; 150; 180; 210; 240; 270; 300, 30; 120; 210; 300, 30; 120; 210; 300	30; 60; 90; 120; 150; 180; 210; 240; 270; 300, 30; 120; 210; 300, 30; 120; 210; 300	30; 60; 90; 120; 150; 180; 210; 240; 270; 300, 30; 120; 210; 300, 30; 120; 210; 300
Liczba grup	10, 1, 1	10, 1, 1	10, 1, 1
Czas pomiędzy grupami [s]	30, 300, 300	30, 300, 300	30, 300, 300

Test 1, 2, 3 – Liczba zrealizowanych zapytań w stosunku do obciążenia : Number of Requests

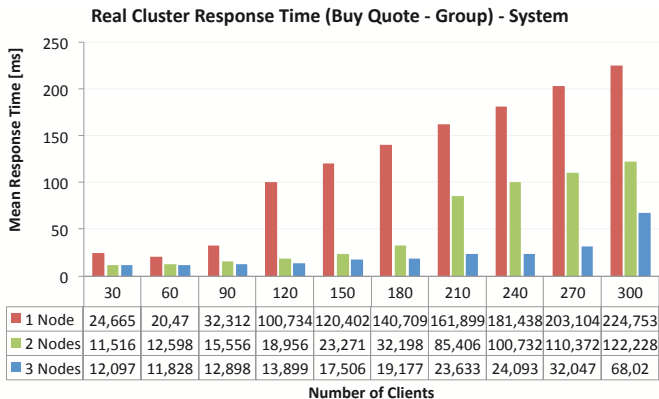


Test 1, 2, 3 – Liczba zrealizowanych zapytań (Buy Quote) w stosunku do obciążenia : Number of Requests

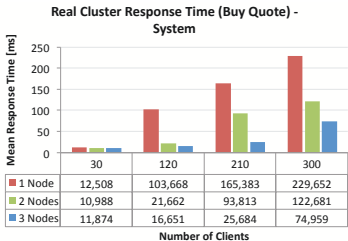
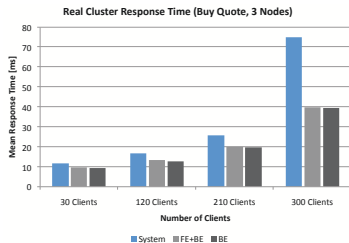
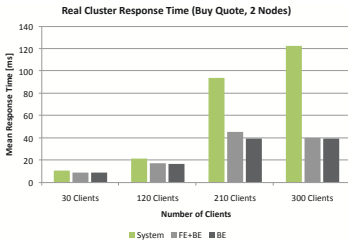
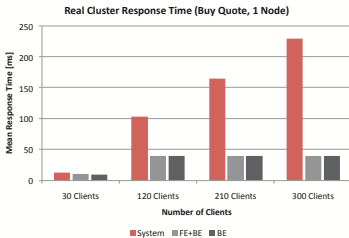
Number of Requests vs Number of Nodes vs Number of Clients
(Buy Quote)



Wydajność a rozmiar klastra (Test 1, 2, 3) : Response Time

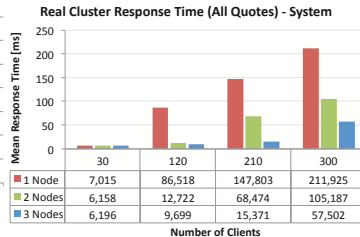
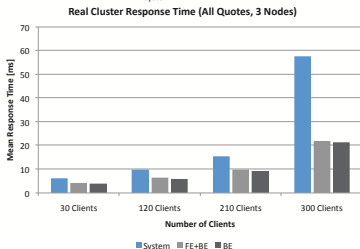
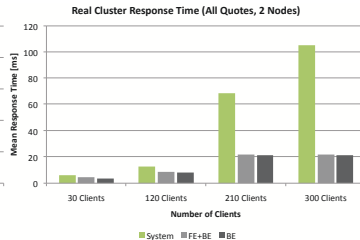
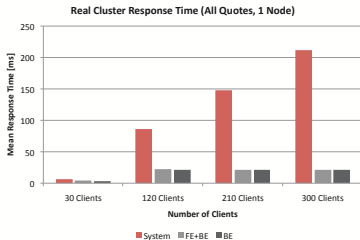


Wydajność a rozmiar klastra (Test 4, 5, 6) : Response Time



	30	120	210	300
1 Node	12,508	103,668	165,383	229,652
2 Nodes	10,988	21,662	93,813	122,681
3 Nodes	11,874	16,651	25,684	74,959

Wydajność a rozmiar klastra (Test 7, 8, 9) : Response Time



Agenda

- Modelowanie
- System rzeczywisty
- Testy systemu webowego
- Symulacje z użyciem QPME - analizy

Parametry, od których zależy czas odpowiedzi

- *Service Demand, Residence Time*
- *Workload Intensity*

Response Time:

$$R = \sum_n^{i=1} R'_i \quad (1)$$

Residence Time:

$$R'_i = Q_i + D_i \quad (2)$$

Queuing Time:

$$Q_i = \sum_n^{i=1} W_i \quad (3)$$

Service Demand:

$$D_i = \sum_n^{i=1} S_i \quad (4)$$

Średni czas obsługi w określonym zasobie, z wyłączeniem czasu oczekiwania na zasób. Nie zależy od obciążenia!

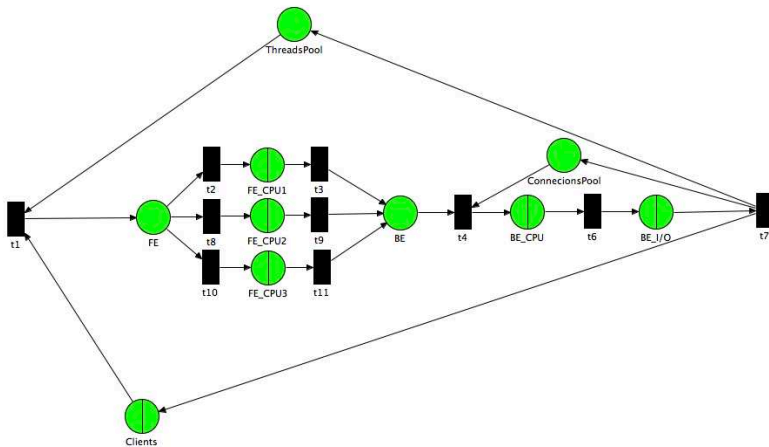
Całkowity czas odpowiedzi

Całkowity czas odpowiedzi:

$$\begin{aligned}
 R_{TOTAL} = & \\
 & R_{CLIENT_{(DEPOSITORY)}} + R_{FE_PLACE_{(QUEUE)}} + \\
 & \sum_{i=1}^n R_{FE_CPUi_{(QUEUE)}} + \sum_{i=1}^n R_{FE_CPUi_{(DEPOSITORY)}} + \\
 & R_{BE_PLACE_{(QUEUE)}} + R_{BE_CPU_{(QUEUE)}} + R_{BE_IO_{(QUEUE)}}
 \end{aligned} \tag{5}$$

- $R_{FE_PLACE_{(QUEUE)}}$ i $R_{BE_PLACE_{(QUEUE)}}$ - są to miejsca użyte do zatrzymania przychodzących zapytań, gdy oczekują one na wątki (serwer aplikacji) i procesy (baza danych)

Model QPN (8c)



Parametry wejściowe

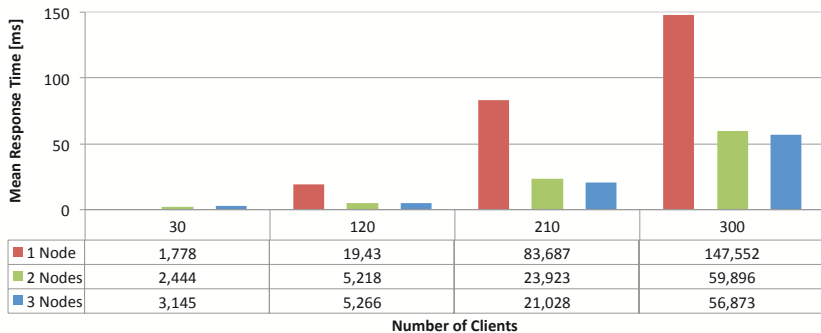
Model/Parametry	Symulacja 7 (One Type/All Quotes)	Symulacja 8 (One Type/All Quotes)	Symulacja 9 (One Type/All Quotes)
QPME			
Load Balancer Kolejki front-end	Nie <i>FE_CPU1</i>	Tak Instancje klastra: <i>FE_CPU1, FE_CPU2</i>	Tak Instancje klastra: <i>FE_CPU1, FE_CPU2, FE_CPU3</i>
Kolejki back-end	<i>BE_CPU, BE_IO</i>	<i>BE_CPU, BE_IO</i>	<i>BE_CPU, BE_IO</i>
Systemowe			
Pula wątków	30	60	90
Pula połączeń z bazą	40	80	120
Obciążenie (klient)			
Liczba zapytań [zapytań/s]	15	15	15
Liczba aktywnych klientów	30; 120; 210; 300	30; 120; 210; 300	30; 120; 210; 300
Liczba grup	1	1	1

Czas obsługi

$D_{FE_CPUi} = 0, 71$ [ms]; $D_{BE_CPU} = 0, 4$ [ms]; $D_{BE_I/O} = 0, 25$ [ms]; $D_{Client} = 66, 67$ [ms]

Wyniki testów (Symulacja 7, 8, 9 - One Type/All Quotes) : Response Time

Cluster Simulation Response Time (One Type/All Quotes) - System



Wnioski z przeprowadzonych prac

- Przygotowano system rzeczywisty i modele symulacyjne
- Modele odwzorowują rzeczywistość

Przyszłe prace

Charakterystyka obciążenia (model klienta):

- Źródło zapytań
- Klasy zapytań
- Użycie elementów sprzętowo-programowych przez poszczególne klasy

Różne typy/klasy klientów to różne zachowania

Różne sposoby przybywania zapytań, popytu na usługi czy trasowania sieciowego.

Performance Analysis of Cluster-based Web System using the QPN Models

(29th International Symposium on Computer and Information Sciences)

Dziękuję za uwagę!

Modelowanie 5

System rzeczywisty 6

Testy systemu webowego 13

Symulacje z użyciem QPME 31

Simulations can be used as experiments.