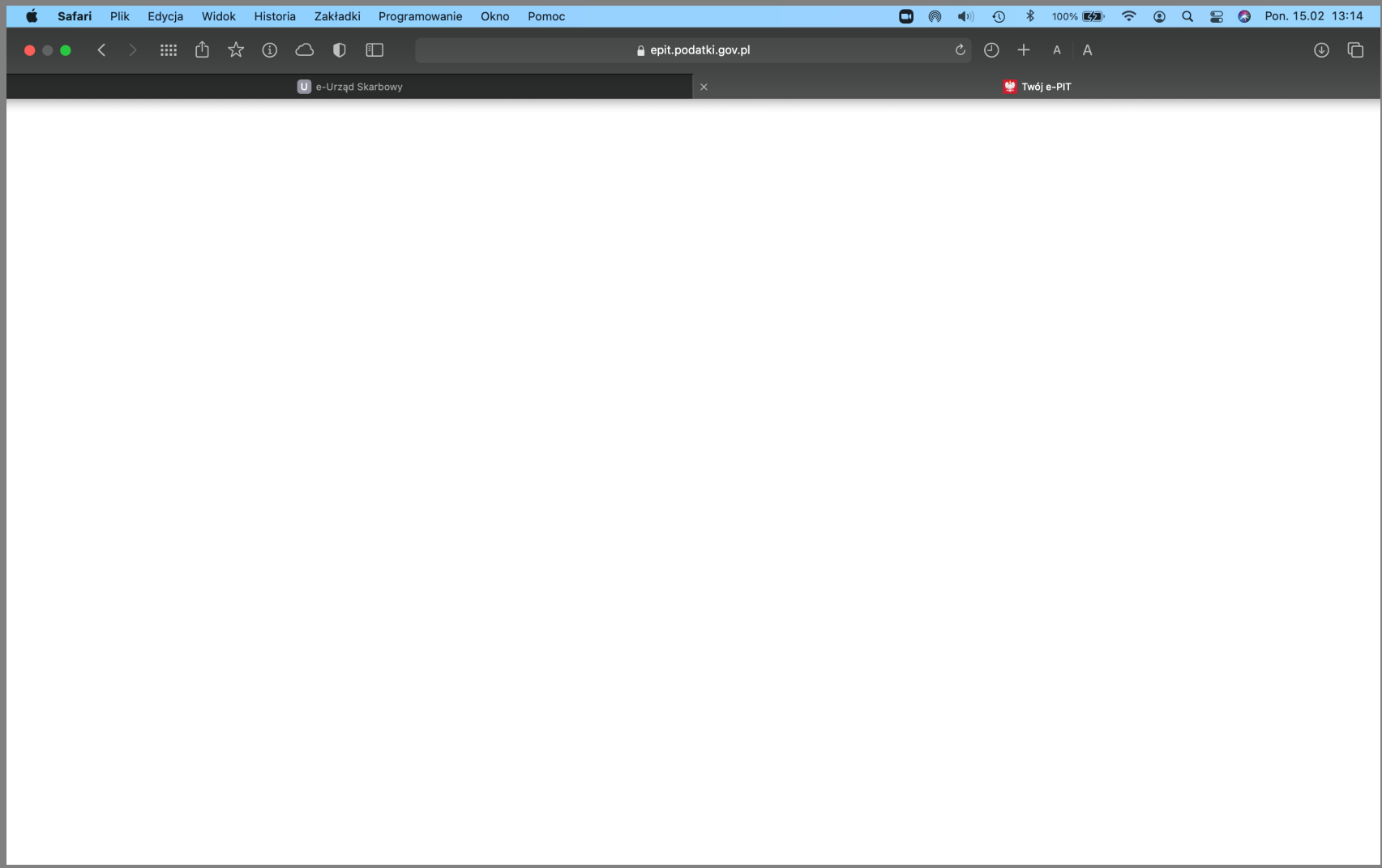

Safari Plik Edycja Widok Historia Zakładki Programowanie Okno Pomoc
100%
Pon. 15.02 13:14
epit.podatki.gov.pl
e-Urząd Skarbowy
Twój e-PIT

Ocena wydajności przykładowego systemu webowego

Tomasz RAK

Department of Computer and Control Engineering
Rzeszow University of Technology

RZESZOW UNIVERSITY OF TECHNOLOGY

Work-in-Progress

Table of Contents

Table of Contents

Publications

QPN models[a]

[a] Rak T.: **Modeling Web Client and System Behavior**, (**2020**) doi:10.3390/info11060337
Rak T.: **Cluster-Based Web System Models for Different Classes of Clients in QPN**, (**2019**) doi:10.1007/978-3-030-21952-9_26
Rak T.: **Performance Modeling Using Queueing Petri Nets** (**2017**) doi:10.1007/978-3-319-59767-6_26
Rak T.: **Response Time Analysis of Distributed Web Systems Using QPNs**, (**2015**) doi:10.1155/2015/490835
Rak T.: **Performance Analysis of Distributed Internet System Models Using QPN Simulation**, (**2014**) doi:10.15439/2014F366
Rak T.: **Performance Analysis of Cluster-Based Web System Using the QPN Models**, (**2014**) doi:10.1007/978-3-319-09465-6_25

Parametry, od których zależy czas odpowiedzi

- *Service/Resource Demand, Residence Time*
- *Workload Intensivity*

Czas odpowiedzi (response time) jest równy sumie czasów obsługi w poszczególnych zasobach (residence time), gdzie: i - liczba miejsc:

$$R = \sum_{i}^{k=1} R_k^{'} \tag{1}$$

Czas obsługi w zasobie (residence time) jest sumą czasu spędzonego w kolejce (queueing time) i średniego czasu obsługi dla zasobu (resource demand):

$$R_k^{'} = Q_k + D_k \tag{2}$$

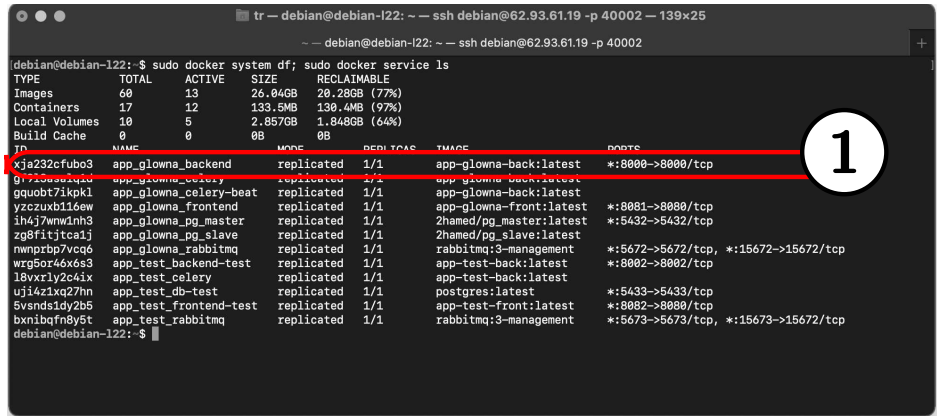
gdzie, czas spędzony w kolejce (czaso oczekiwania) na zasób to $Q_k = \sum_i^{k=1} q_k$ i średni czas obsługi w określonym zasobie to $D_k = \sum_i^{k=1} d_k$.

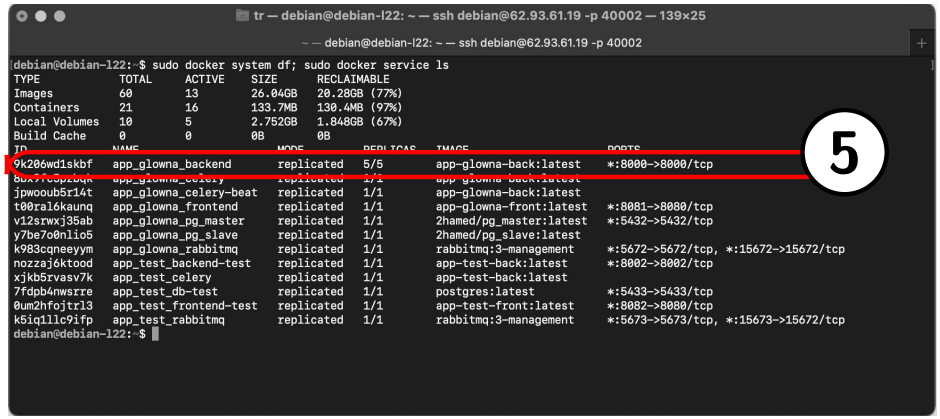
Średni czas obsługi w określonym zasobie, z wyłączeniem czasu oczekiwania na zasób.

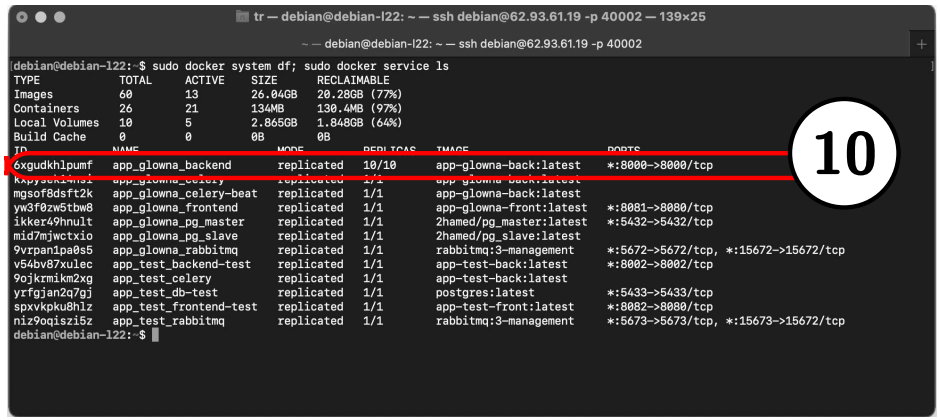
[https://research.spec.org/tools/overview/librede.html]

Table of Contents

Docker Engine in Swarm Mode







Debian 10 (2020.05) GL
CPU: 4 RAM: 10GB HDD: 100GB

Debian 10 (2020.05) GL
CPU: 8 RAM: 20GB HDD: 100GB

Debian 10 (2020.05) GL
CPU: 12 RAM: 30GB HDD: 100GB

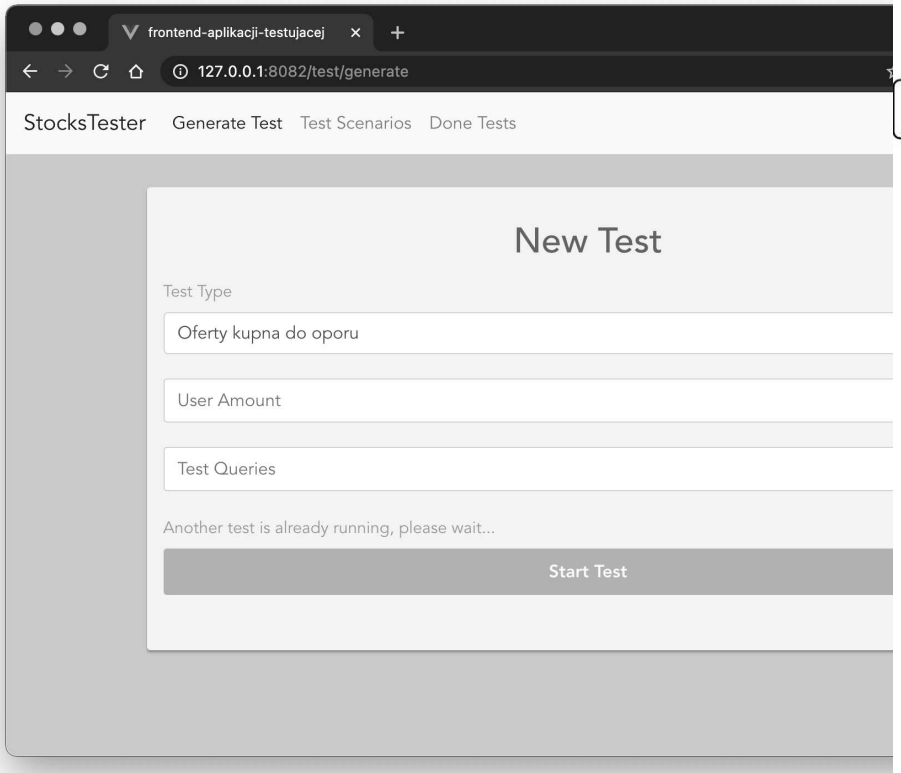
Transactions and API Endpoints

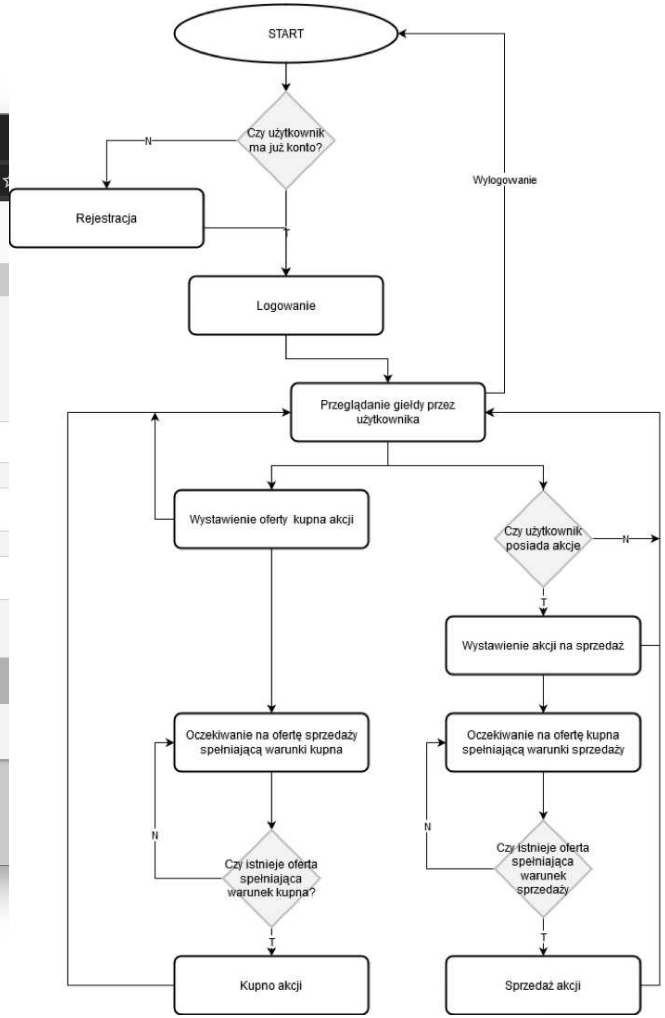
- User registration
- User login
- Buying stock
- Selling stock
- Display the list of companies
- Display details of a given company
- Display the user's profile
- Display offers
- Create an offer

- (1) User registration
- (2) User login; (3) User logout
- (4) Create a new purchase offer; (5) Delete the purchase offer
- (6) Create a new sale offer; (7) Delete the sale offer
- (8) Return the list of companies
- (9) Return the list of all companies; (10) Return details about the company
- (11) Details of the current user, adding users, editing the user
- (12) Return the current user's wallet status
- (13) Return the list of resources owned by the user
- (14) Return the list of active sell/buy offers for a given user
- (15) Return the list of completed transactions for a given user
- (16) Return the list of all available actions
- (17) Allow you to buy stocks at the current price
- (18) Allow you to sell stocks at the current price
- (19) Return the list of all buy and sell orders: active and closed
- (20) Return the list of all buy and sell orders for a given action: active and closed

Table of Contents

Benchmark


frontend-aplikacji-testujacej
127.0.0.1:8082/test/generate
StocksTester
Generate Test
Test Scenarios
Done Tests
New Test
Test Type
Oferty kupna do oporu
User Amount
Test Queries
Another test is already running, please wait...
Start Test


START
Czy użytkownik ma już konto?
N
T
Rejestracja
Logowanie
Wylogowanie
Przeglądanie giełdy przez użytkownika
Wystawienie oferty kupna akcji
Czy użytkownik posiada akcję
N
T
Wystawienie akcji na sprzedaż
Oczekiwanie na ofertę sprzedaży spełniającą warunki kupna
Oczekiwanie na ofertę kupna spełniającą warunki sprzedaży
N
Czy istnieje oferta spełniająca warunek kupna?
N
Czy istnieje oferta spełniająca warunek sprzedaży
T
T
Kupno akcji
Sprzedaż akcji

Tests Scenarios

S1 - kupuj do oporu i wystawiaj oferty sprzedaży (kdoiwos)

- Rejestracja użytkownika
- Logowanie
- Lista wszystkich dostępnych zasobów
- Stan portfela obecnego użytkownika
- Kupno pojedynczego zasobu
- Lista zasobów posiadanych przez użytkownika
- Oferty sprzedaży
- Lista obecnych ofert sprzedaży/kupna danego użytkownika
- Lista zrealizowanych transakcji danego użytkownika

S2 - kupuj i sprzedawaj (kis)

- Rejestracja użytkownika
- Logowanie
- Lista wszystkich dostępnych zasobów
- Stan portfela obecnego użytkownika
- Kupno pojedynczego zasobu
- Lista zasobów posiadanych przez użytkownika
- Oferty sprzedaży
- Sprzedaż pojedynczego zasobu
- Lista obecnych ofert sprzedaży/kupna danego użytkownika
- Oferty sprzedaży

S3 - kupuj kolejne dopóki są fundusze (kkdsf)

- Rejestracja użytkownika
- Logowanie
- Lista wszystkich dostępnych zasobów
- Stan portfela obecnego użytkownika
- Kupno pojedynczego zasobu

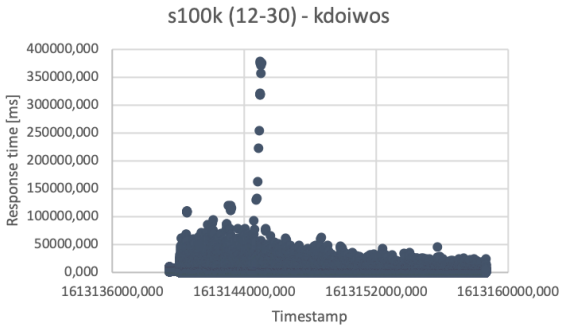
Multi-Container Laboratory Environment

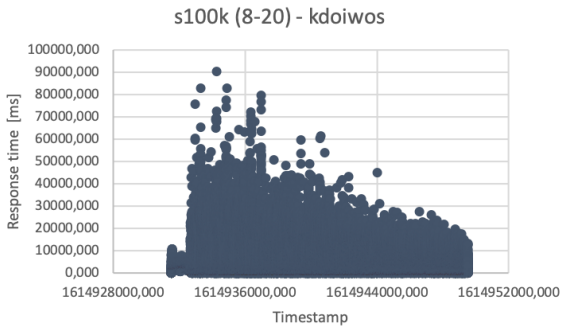
Scenario	S1	S2	S3

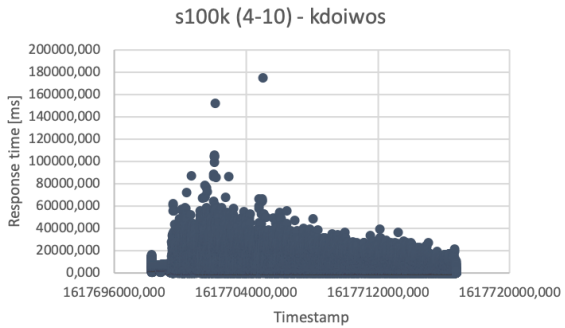
Parameter	Exp. 1			Exp. 2			Exp. 3		
Processors	12			8			4		
RAM [GB]	30			20			10		
Container [a]	1	5	10	1	5	10	1	5	10
1/*Think_time* - [req/s] [b]	6,242	4,849	3,826	6,732	5,009	2,975	6,366	3,152	1,644
Think_time - [s]	0,166	0,207	0,261	0,155	0,200	0,337	0,164	0,317	0,610

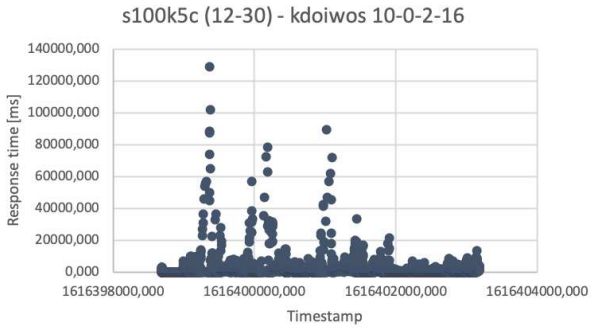
[a] Number of database connections in all cases is equal 90 per container.
[b] Number of clients (workload) in all cases is equal 90.

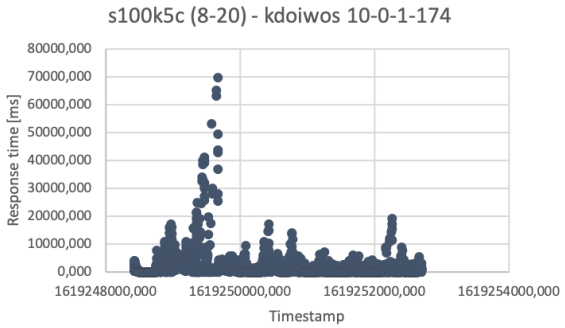
Response Time - 100000 [req]

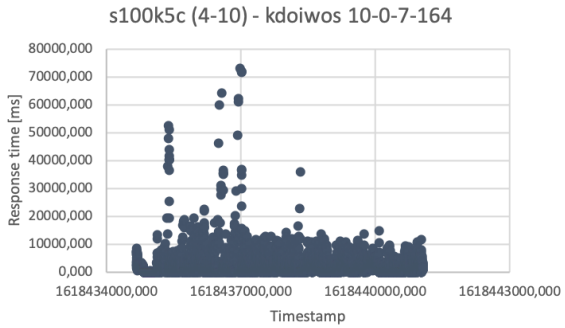


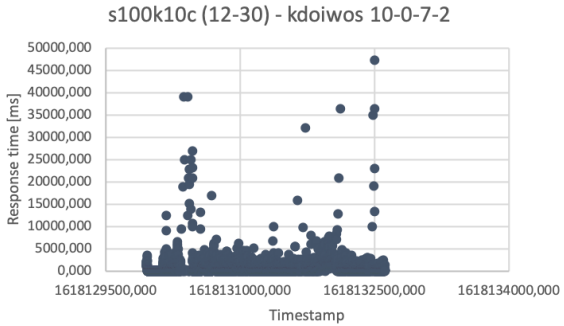


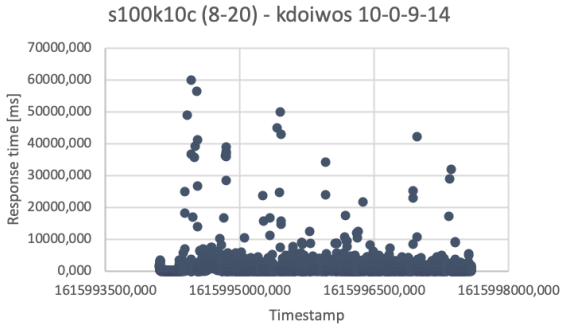


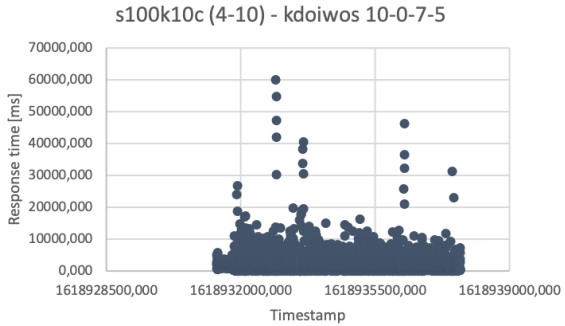




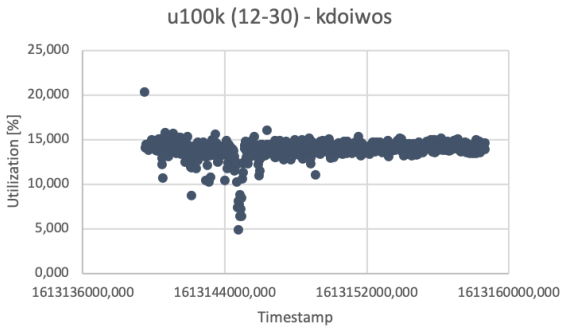


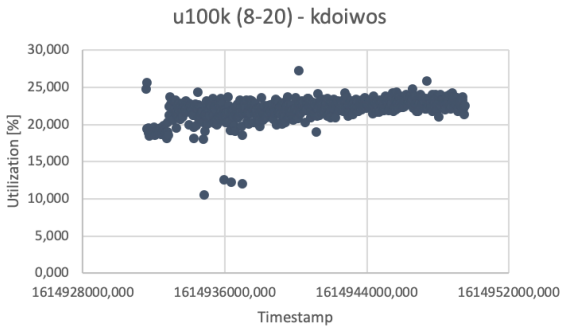


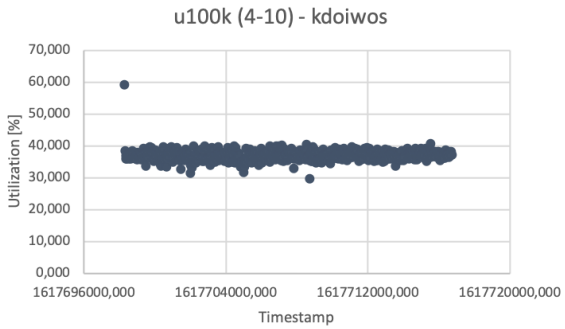


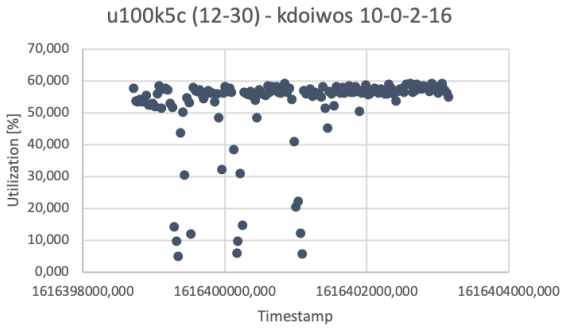


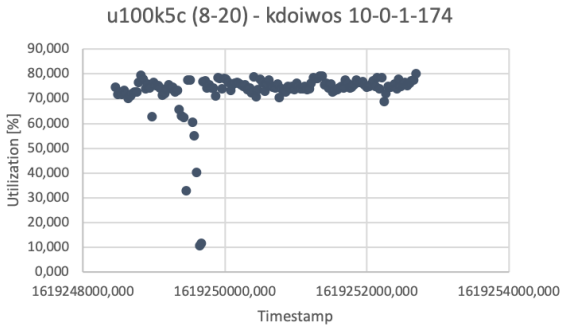
Utilization - 100000 [req]

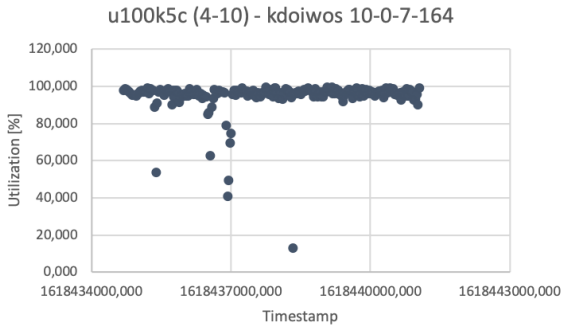


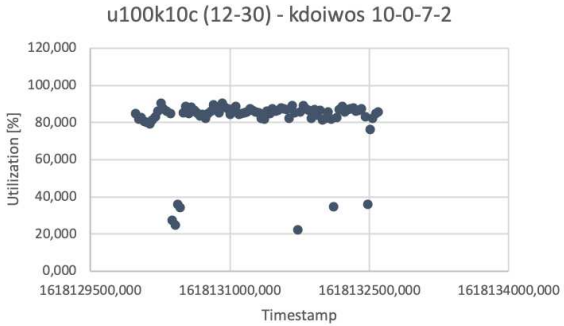


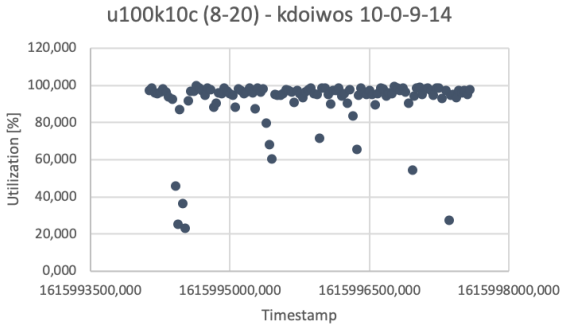


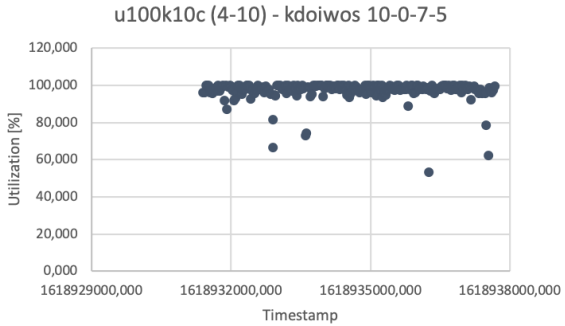




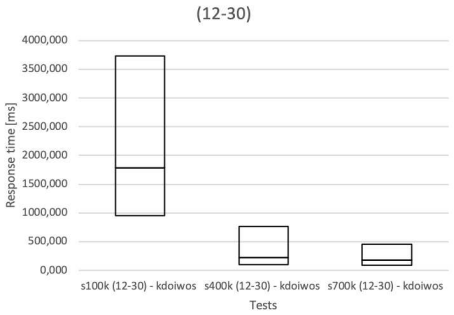




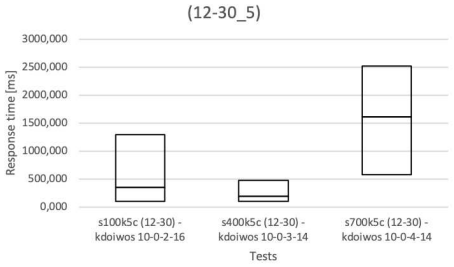




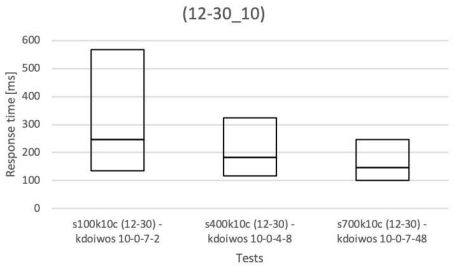
Response Time - 100000, 400000, 700000 [req]



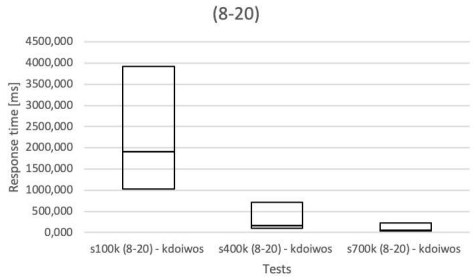
0,904[s]



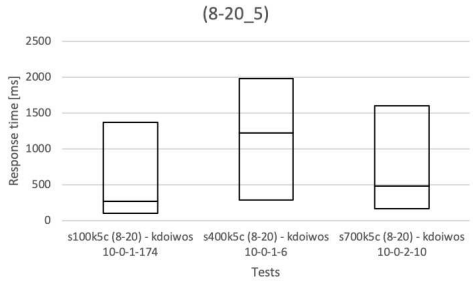
1,014[s]



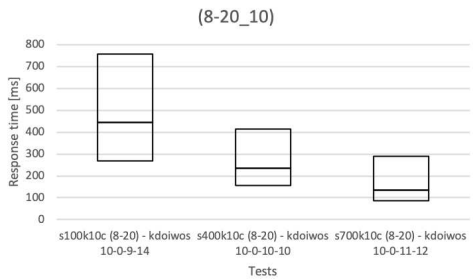
0,599[s]



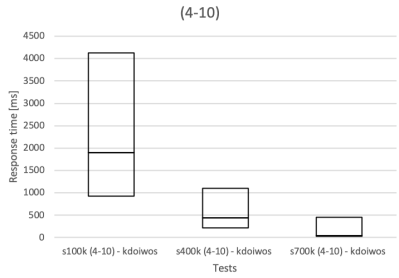
1,128[s]



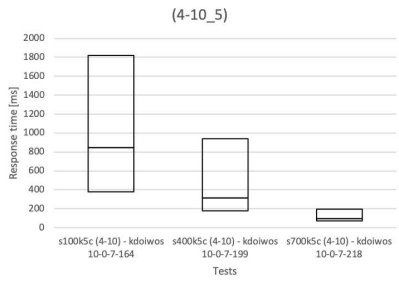
0,928[s]



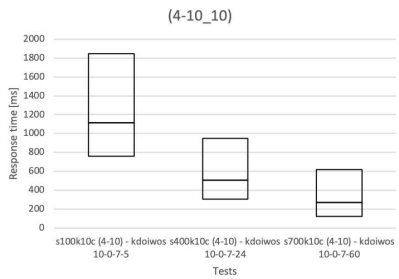
0,871[s]



0,804[s]

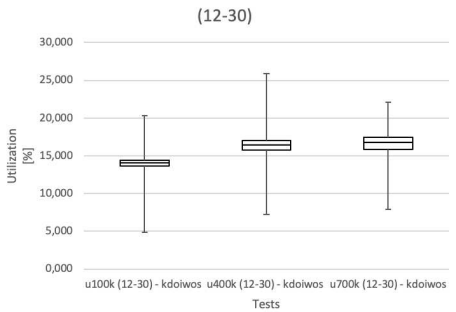


1,455[s]

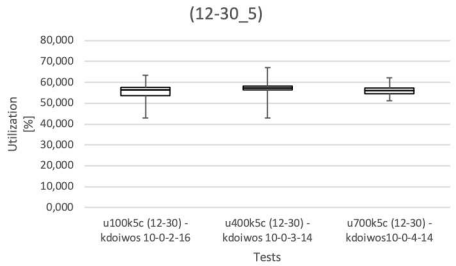


0,846[s]

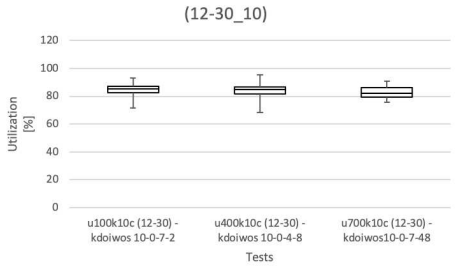
Utilization - 100000, 400000, 700000 [req]



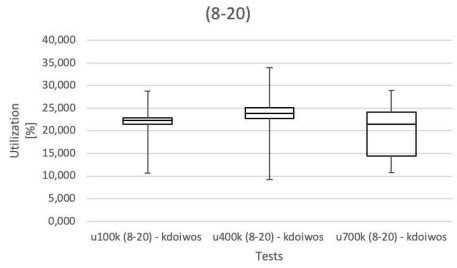
15[%]



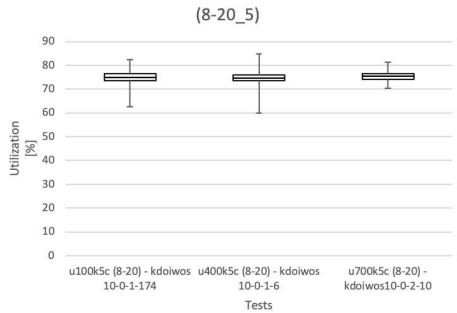
54[%]



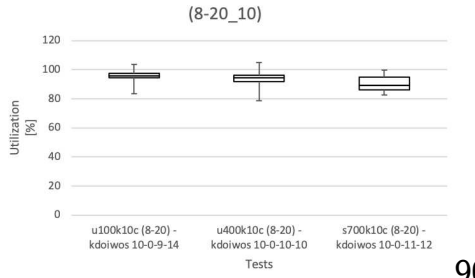
81[%]



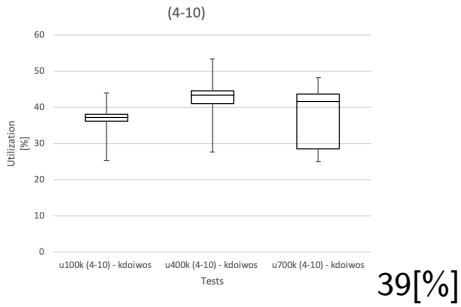
22[%]



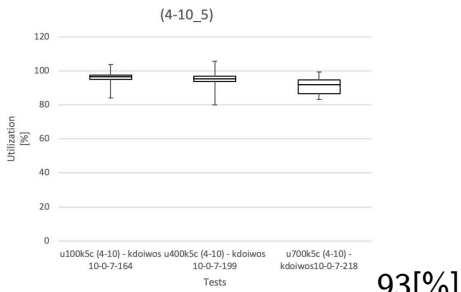
72[%]



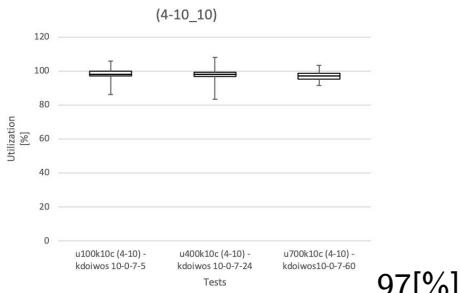
90[%]



39[%]



93[%]



97[%]

CSV Files

service10-0-2-39kdoiwos700k.csv

Add Column | Delete Column | Add Row | Delete Row

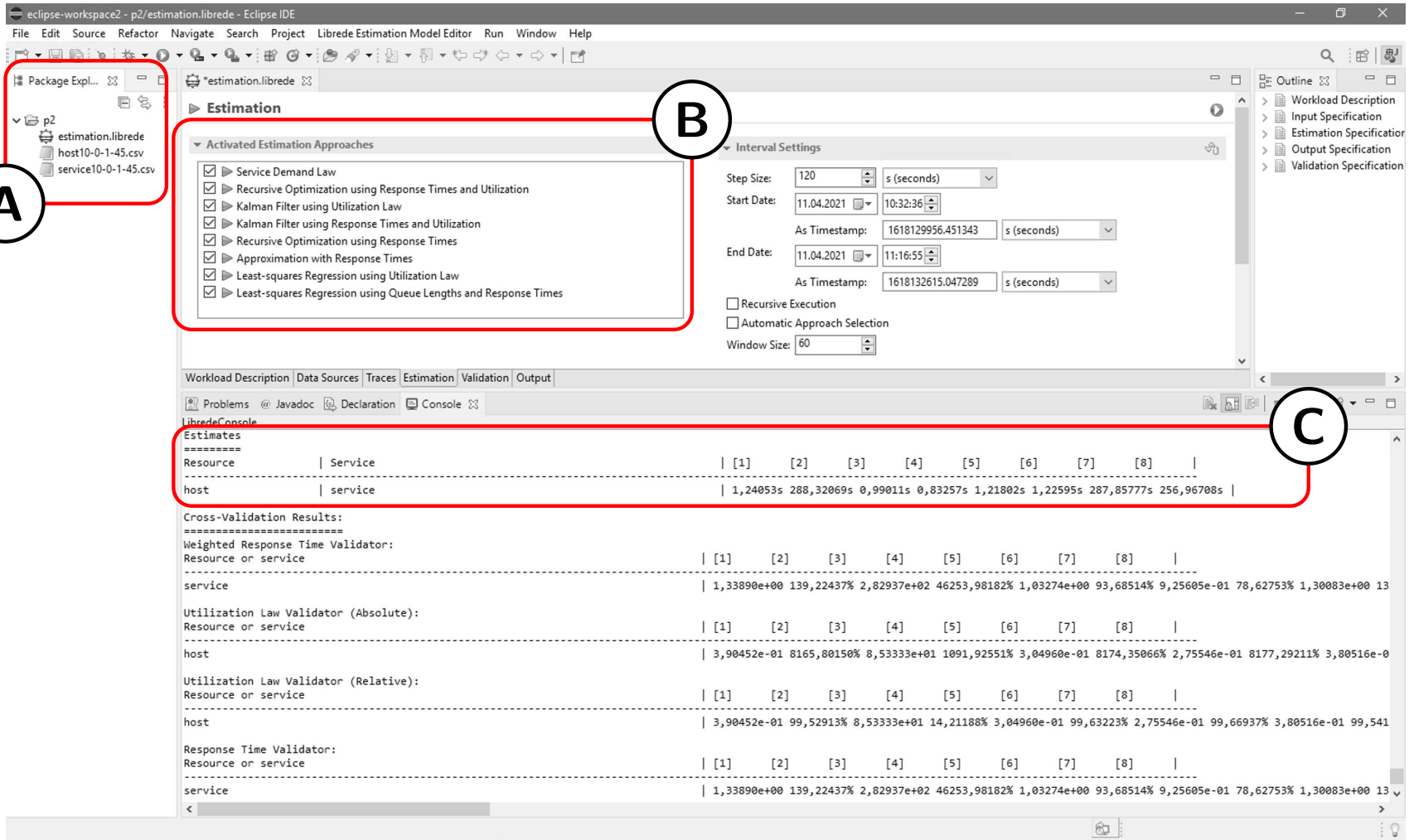
A	B
1613373696.191592	1753.28
1613373696.477225	1678.788
1613373696.610037	1797.967
1613373696.71201	1825.4660000000001
1613373696.899689	1944.281
1613373696.908943	2205.3889999999997
1613373696.909508	2130.187
1613373697.007817	1554.035
1613373697.219769	1987.484
1613373697.339968	2473.657
1613373697.38726	1815.132
1613373697.40506	2685.619
1613373697.433444	2025.426
1613373697.607027	3051.282
1613373697.61033	2449.798
1613373697.649196	2908.2309999999998
1613373697.64965	1915.103
1613373697.669059	2292.038
1613373697.690058	2793.94
1613373697.695824	2214.272
1613373697.78446	2908.925
1613373697.816416	2522.0099999999998
1613373697.889472	1511.943
1613373698.347836	2646.2870000000003
1613373698.58034	2376.822
1613373698.772177	2561.2180000000003
1613373698.980211	2559.198
1613373698.981579	2792.2349999999997

host10-0-2-39kdoiwos700k.csv

Add Column | Delete Column | Add Row | Delete Row

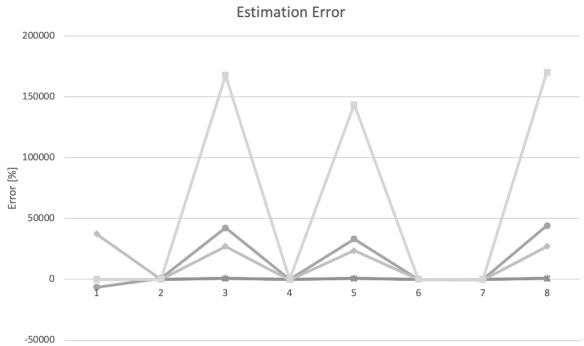
A	B
1613373696	16.773873873873868
1613373726	22.089999999999996
1613373756	15.398872180451129
1613373786	16.288888888888888
1613373816	16.486290322580643
1613373846	16.274297188755025
1613373876	15.531600000000003
1613373906	17.109090909090902
1613373936	16.948387096774198
1613373966	17.23089430894309
1613373996	17.32521008403361
1613374026	17.339999999999996
1613374056	17.661410788381737
1613374086	17.773333333333333
1613374116	17.5984
1613374146	18.56639004149378
1613374176	18.3688524590164
1613374206	17.9771186440678
1613374236	17.695238095238103
1613374266	18.063274336283182
1613374296	16.811740890688245
1613374326	17.689719626168227
1613374356	17.55248868778281
1613374386	17.81792452830188
1613374416	17.740888888888904
1613374446	17.58333333333333
1613374476	18.61674208144796
1613374506	18.604977375565614
1613374536	17.484507042253522
1613374566	18.026146788990836

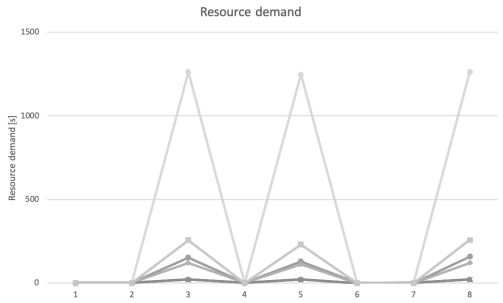
Estimation Approaches



Methods

(1)Approximation with Response Times;(2)Kalman Filter using Response Times and Utilization;(3)Kalman Filter using Utilization Law;(4)Least-squares Regression using Queue Lengths and Response Times;(5)Least-squares Regression using Utilization Law;(6)Recursive Optimization using Response Times;(7)Recursive Optimization using Response Times and Utilization;(8)Service Demand Law





Estimation Error

Error [%]

100
0
-100
-200
-300
-400

	Recursive Optimization using Response Times	Recursive Optimization using Response Times and Utilization
12-30s12	0,646818098	-0,897905511
12-30_5s12	79,38060097	0,430180122
12-30_10s12	0,351120179	0,230979692
8-20s8	0,771952155	1,04651969
8-20_5s8	3,154876279	-322,1905826
8-20_10s8	0,394169807	0,355737927

A

B

Resource demand

Resource demand [s]

1,6
1,2
0,8
0,4
0

	Recursive Optimization using Response Times	Recursive Optimization using Response Times and Utilization
12-30s12	0,576156667	1,41902
12-30_5s12	1,049172	0,405357333
12-30_10s12	0,411842667	0,318710333
8-20s8	0,434016667	1,377866667
8-20_5s8	0,678253333	0,385805333
8-20_10s8	0,503761667	0,411330333

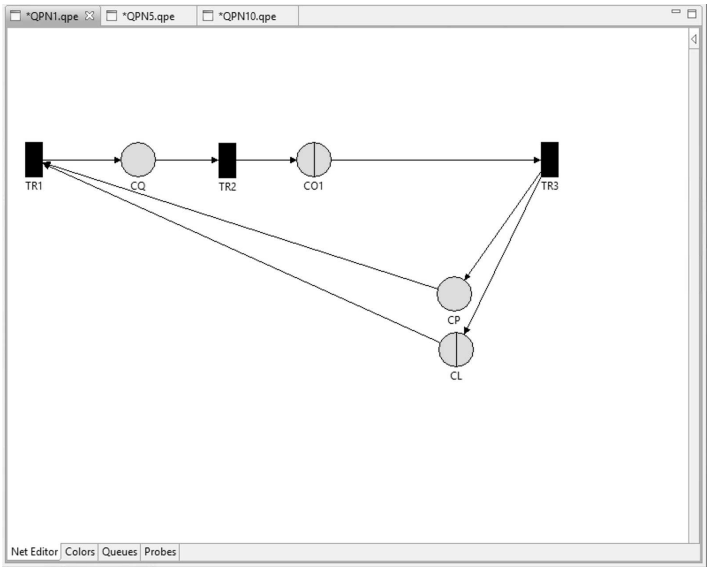
Average Resource Demand

	Resource_demand [s]	1/*Resource_demand* [req/s]
12-30 $^{(a)}$	0,997588333	1,002417497
12-30_5 $^{(b)}$	0,727264667	1,37501524
12-30_10 $^{(b)}$	0,3652765	2,737652162
8-20 $^{(a)}$	0,905941667	1,10382383
8-20_5 $^{(b)}$	0,532029333	1,879595611
8-20_10 $^{(b)}$	0,457546	2,185572598
4-10 $^{(a)}$	0,849838333	1,176694391
4-10_5 $^{(b)}$	~~0,397753~~	~~2,514123086~~
4-10_10 $^{(b)}$	~~0,736487259~~	~~1,35779674~~

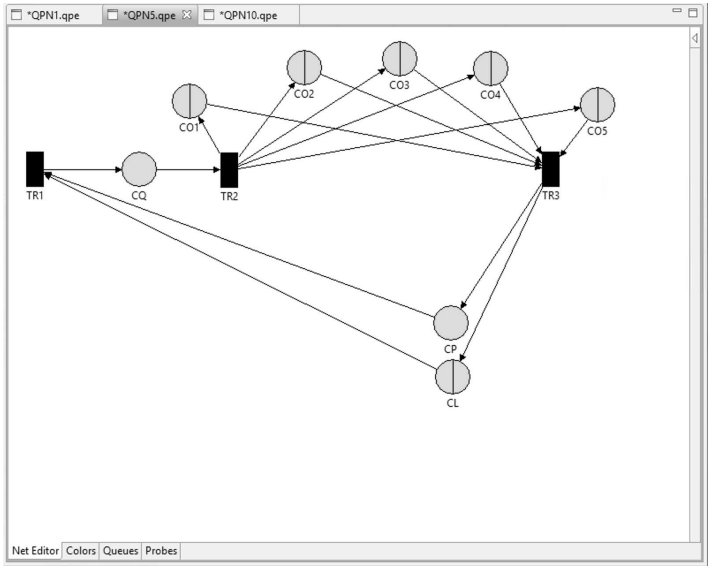
$^{(a)}$ One container.
$^{(a)}$ The single container.

Table of Contents

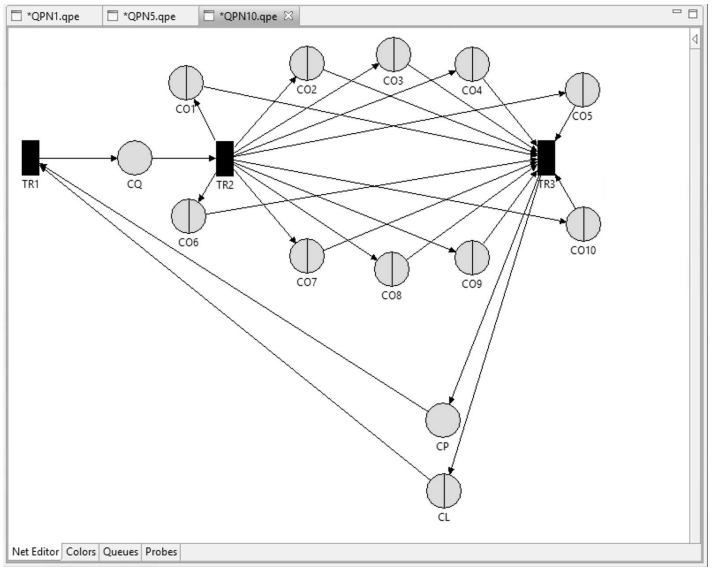
QP Net



QPN1



QPN5



QPN10

Input Parameters of Simulations (Client and System)

Scenario	S1	S2	S3

Parameter	Sim. 1			Sim. 2			Sim. 3		
Number of servers [a]	12			8			4		
Model	QPN1	QPN5	QPN10	QPN1	QPN5	QPN10	QPN1	QPN5	QPN10
CL queueing place	90			90			90		
X_{CL} - [req/s]	6,242	4,849	3,826	6,732	5,064	2,975	6,366	3,152	1,644
CP place [b]	90			90			90		
X_{CO_i} - [req/s] [c]	1,002	1,375	2,737	1,103	1,879	2,185	0,164	0,317	0,610

[a] FCFS scheduling strategy.
[b] Connections for containers – Initial marking.
[c] i - number of containers (1, 5, 10).

Response Time

	12-30 $^{(a)}$	12-30_5$^{(b)}$	12-30_10$^{(b)}$
Simulation [s]	1,003483	1,222707	0,528284
Measured [s]	0,904	1,014	0,599
Error [%]	-11,00475664	-20,58254438	11,80567613

	8-20 $^{(a)}$	8-20_5$^{(b)}$	8-20_10$^{(b)}$
Simulation [s]	1,075831	0,832224	0,711255
Measured [s]	1,128	0,928	0,871
Error [%]	4,624911348	10,32068966	18,34041332

	4-10 $^{(a)}$	4-10_5$^{(b)}$	4-10_10$^{(b)}$
Simulation [s]	0,851	1,914936	2,031159
Measured [s]	0,804	1,455	0,846
Error [%]	-5,845771144	-31,61072165	-140,0897163

$^{(a)}$ One container.
$^{(b)}$ The single container.

Conclusions

The convergence of simulation results with the real system results confirms model correctness.

- We can use the proposed analysis to apply the modification of the container-based system without interfering into the system construction or software (main achievement).
- It is possible to analyze the influence of the containers number for the system response time (practical value).
- The modeling approach presented in this presentation differs from my previous works where it was based on resource demand measurement of native system.

Daniel A. Menascé

"Verify and validate the models (...) a certain acceptable mergin of error (...) resource utilizations within 10%, system throughput within 10%, and response time within 20% are consideren acceptable."

Ocena wydajności przykładowego systemu webowego

Thank you for your attention!

Related Works (2021):

Czachórski, T.: Time-Dependent Performance of a Multi-Hop Software Defined Network
Herbst, J., et al.: SuanMing: Explainable Prediction of Performance Degradations in Microservice Applications
Iosup, A., et al.: An Analysis of Distributed Systems Syllabi With a Focus on Performance-Related Topics
Kounev, S.: A New Course on Systems Benchmarking - For Scientists and Engineers
Nguyen, V.Q., et al.: Efficiently Estimating Joining Cost of Subqueries in Regular Path Queries
Rygielski, P.: A Simulation-Based Optimization Framework for Online Adaptation of Networks
Zatwarnicki, K.: Acquisition and Modeling of Website Parameters

估計並不總是關於計算。
"Estimation is not always about doing calculations."