Performance Modeling Using Queueing Petri Nets

Tomasz RAK

Department of Computer and Control Engineering
Rzeszow University of Technology

We need high performance to process more RPS.

Table of contents

Table of contents

Related works (Chen X., Kounev S., Koziolek H., Meier P., Rathfelder C., Spinner S., Zatwarnicki K.)

We can not always add more and more new devices to improve performance, because the initial and maintenance cost will become too high. Power consumption depends on the load and on the number of running nodes in the cluster-based distributed Web system.

The question:

What is the performance of the system?

The main aim of the work was to develop models of cluster-based distributed Web system. The related works can be divided into publications based on analysis of QN and PN models.

My approach joins LT and PE

- Educated Guess
- Load Testing (LT)
- Performance Engineering (PE) models (provide some recommendations to realize the required performance level):
  - Performance model (used to predict performance of the system under study)
  - Availability model
  - Reliability model
  - Cost model

Daniel A. Menascé

"PE analyzes the expected performance characteristics of a system during the different phases of its life cycle."

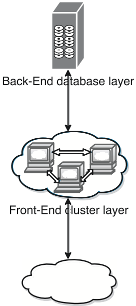
Computer system (experiments) and performance model (simulations) – response time parameter

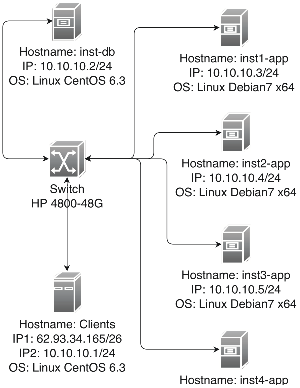
Workload ⇓	Configuration parameters ⇓	Workload model ⇓	Configuration parameters ⇓
Computer system		Performance model	
⇓		⇓	
Measured performance parameters		Calculated performance parameters	

Performance metrics e.g.: throughput, response time.

Table of contents

E-commerce system deployment (HP ProLiant DL180 G6)





The FE layer for presenting and processing requests. The BE layer stores the systems data.

Laboratory Cluster Environment

Server/Parameters	Experiment no. 2 (All Quotes)
"Clients" GlassFish AS nodes Oracle DBS node	10.10.10.1 10.10.10.4-5 10.10.10.2
AS threads pool DBS connections pool	2 x 30 2 x 40
Number of RPS	15
Number of clients	30; 120; 210; 300
Experiment time [s]	300

Installation and configuration of services:

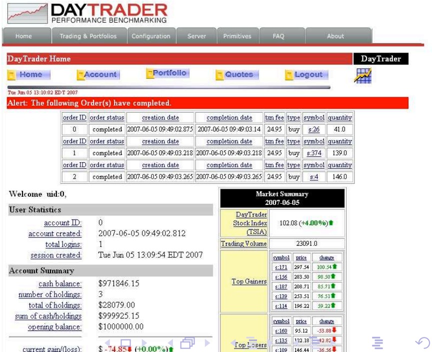
- Apache Tomcat Connector load balancer
- GlassFish 3.1 Open Source Edition application server
- Oracle 11g database

Applications:

- "Clients"
- DayTrader

Sample of trade performance benchmark – Internet Stock Exchange (one class of Internet systems)

- Apache Geronimo part (IBM product)
- Java EE 6 technology
- It allows users to: login, view their portfolio, lookup stock quotes, and buy or sell stock shares
- Benchmark with dynamically changing offer
- The state of the system is constantly changing

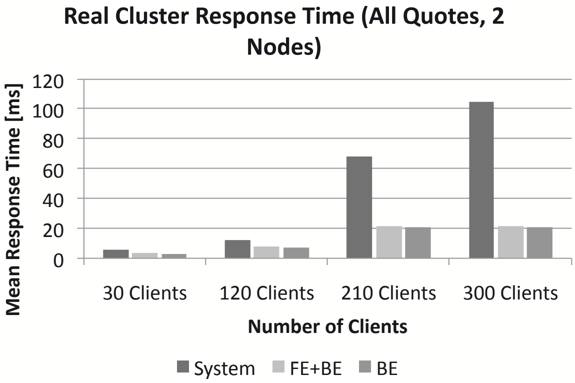


Transaction emulates a specific classes of client session

Query	Transaction	Parameters	Description
buy	Buy Quote	*symbol* – stocks symbols; *quantity* – number	Buy and return the number of specified stocks
sell	Sell Quote	*holdingId* – stocks ID, which will be sold	Sell indicated stocks
update_profile	Update Profile	*password* and *cpassword* – new password; *fullname* – name and surname; *address* – address; *creditcard* – credit card number; *email* – email address	Update the logged-in user profile
quotes	Show Quotes	*symbols* – comma-separated stocks to display	Display information about the required stocks
home	Get Home	–	Generates a logged-in user's homepage
portfolio	Get Portfolio	–	Display a list of stocks held by the user
account	Show Account	–	Display the logged-in user profile
login	–	*uuid* – user ID; *password* – user password	Log the user in the system (session is created on the server side and its identifier returned in cookie)
logout	–	–	Close the user session

Behavior of sell and buy requests concerns all parts of the system resources. My previous works were based on one class of requests, and now in this investigation I used all classes of requests (types).

Mean response time for all requests classes with different number of clients in clustered environments



- $RPS_{FE\_CPU} = 1400$
- $RPS_{BE\_I/O} = 7500$

Table of contents

Queueing Nets and Petri Nets (combination)

QNs – quantitative analysis

Queueing Nets have a queue, scheduling discipline and are suitable for modeling competition of equipment.

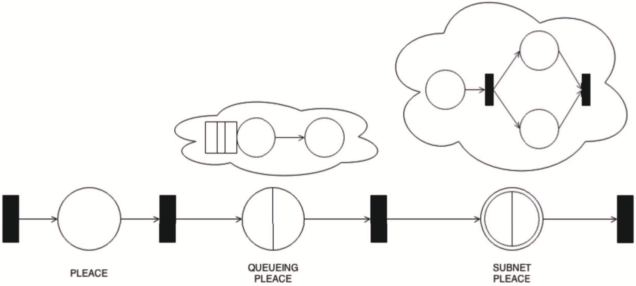
PNs – qualitative analysis

Petri Nets have tokens representing the tasks and are suitable for modeling software.

QPNs add queueing and time aspects to the net

Queueing Petri Nets have the advantages of Queueing Nets (e.g., evaluation of the system performance, the network efficiency) and Petri Nets (e.g., logical assessment of the system correctness). QPNs integrate hardware and software aspects of the system behaviour into the same model.

QP Net graphical notification (Queueing Petri net Modeling Environment)



- SimQPN (discrete event simulation engine)
- Queueing Petri Editor (Net Editor, Color Editor, Queues Editor)

Parameters that determine the response time

- Workload intensity, hardware and software parameters
- Queueing time, service demand was determined experimentally: $d_{FE\_CPU} = 1/RPS_{FE\_CPU} = 0,714$ [ms], $d_{BE\_I/O} = 1/RPS_{BE\_I/O} = 0,133$ [ms]

Response time:

$$R = \sum_{n}^{i=1} R_i^{'} \qquad (1)$$

Residence time:

$$R_i^{'} = Q_i + D_i \qquad (2)$$

Queueing time:

$$Q_i = \sum_{n}^{i=1} q_i \qquad (3)$$

Service demand:

$$D_i = \sum_{n}^{i=1} d_i \qquad (4)$$

Average service time in a particular resource, excluding the time waiting for the resource.

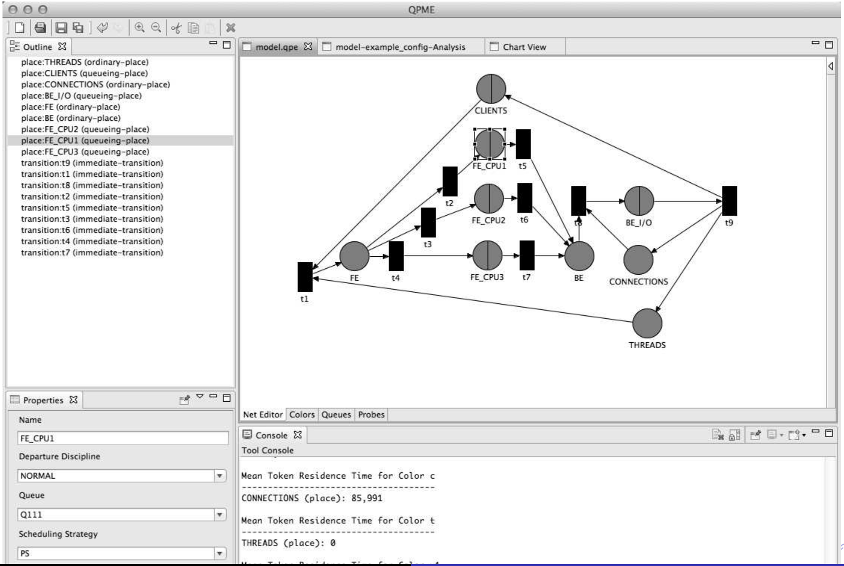
Table of contents

Description

- Servers of the FE layer are modeled using -/M/1/PS/∞ queueing systems (*FE_CPU* queueing places). The BE server is modeled by -/M/1/FIFO/∞ queueing system (*BE_I/O* queueing place).
- *FE* and *BE* places used to stop incoming requests when they await application server threads and database server connections respectively.
- Clients think time is modeled by IS scheduling strategy (*CLIENTS* queueing place).
- Application server threads pool and database server connections pool are modeled respectively by *THREADS* and *CONNECTIONS* places.

-/M/1/PS/∞ – randomly arriving requests, exponential service rate, one server, scheduling strategy, infinite queue.

QP Net – Model of the system with FE cluster (3 nodes)

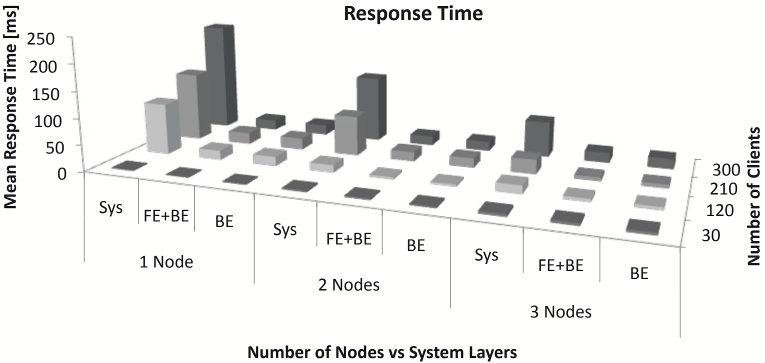


Input and workload parameters

Parameter	Value
d_{FE_CPUn} [a] [ms]	0.714 (1400 [RPS])
$d_{BE\_I/O}$ [ms]	0.133 (7500 [RPS])
THREADS place	30[b]
CONNECTIONS place	40[c]
CLIENTS queueing place	30; 120; 210; 300
$d_{CLIENTS}$ [ms]	66.67 (15 [RPS])
Simulation time [s]	300

[a] n – number of front-end nodes (1-3)
[b] 30, 60 and 90 threads for one, two and three FE nodes respectively – Initial marking per node
[c] 40, 80 and 120 connections for one, two and three FE nodes respectively – Initial marking per node

Mean response time
Response time error (300 clients, 15 RPS workload)



Number of nodes	Model [ms]	Measured [ms]	Error [%]
1	241,22	211,92	13,8
2	127,76	105,19	21,4
3	69,22	57,50	20,3

Conclusions - convergence of simulation results with the real system results confirms correctness

- We can use this analysis to apply the systems modification without interfering in the computer construction or in software. (main achievement)
- For the service response times, the relative prediction error was around 20%.
- The modeling approach presented in this paper differs from my previous works that they were based on QNs and Time Coloured PNs.
- The modeling approach presented in this paper differs from my last work based on QPNs because all types of tokens (requests classes) were not used earlier. The errors are smaller (about 2%).

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."

Performance Modeling Using Queueing Petri Nets (CN'17)

Thank you for your attention!

Answers for comments of reviewers:

- The QPN model was simulated using the method of non-overlapping batch means method to estimate steady state mean token residence times. The average predicted response times are within 95[%] confidence interval of the measured average response times. The data reported by SimQPN is very stable.
- The validation results show the main advantage of this model.
- We shall study the compromise between a perceived average response time and energy consumption (practical value). We could add new hardware elements, but this solution would mean greater costs.

> Suppose one of you wants to build a tower. Won't you first sit down and estimate the cost to see if you have enough money to complete it? – The Bible, Luke 14:28