

# Analysis and Neural Networks Modeling of Web Server Performances Using MySQL and PostgreSQL

Fontaine Rafamantanantsoa\*, Maherindefo Laha

University of Fianarantsoa, Fianarantsoa, Madagascar

Email: \*fontainerafamant@yahoo.fr

**How to cite this paper:** Rafamantanantsoa, F. and Laha, M. (2018) Analysis and Neural Networks Modeling of Web Server Performances Using MySQL and PostgreSQL. *Communications and Network*, 10, 142-151.

<https://doi.org/10.4236/cn.2018.104012>

**Received:** July 27, 2018

**Accepted:** October 20, 2018

**Published:** October 23, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

The purpose of this study is to analyze and then model, using neural network models, the performance of the Web server in order to improve them. In our experiments, the parameters taken into account are the number of instances of clients simultaneously requesting the same Web page that contains the same SQL queries, the number of tables queried by the SQL, the number of records to be displayed on the requested Web pages, and the type of used database server. This work demonstrates the influences of these parameters on the results of Web server performance analyzes. For the MySQL database server, it has been observed that the mean response time of the Web server tends to become increasingly slow as the number of client connection occurrences as well as the number of records to display increases. For the PostgreSQL database server, the mean response time of the Web server does not change much, although there is an increase in the number of clients and/or size of information to be displayed on Web pages. Although it has been observed that the mean response time of the Web server is generally a little faster for the MySQL database server, it has been noted that this mean response time of the Web server is more stable for PostgreSQL database server.

## Keywords

Web Server, Performances, Neural Network, Database Server, MySQL, PostgreSQL, Apache, Mean Response Time, Apache Benchmark, SNNS

## 1. Introduction

Different types of information are available on different websites around the world via the client-server system commonly known as the World Wide Web (www). Indeed, Web technology offers the ability to manipulate data stored on

database servers. Web server performance is important because users require the access speed to different information on the Internet. Alternatively, the number of clients, the type of database servers used, the quality of the network, the type and size of the information to be manipulated, affect the latency of the Web server.

On each server, the two characteristics that mark the clients are: the wait time closely related to the response time of the server, *i.e.* the time required by the browser to display the information, and the time taken by the server to satisfy customer requests. Previous studies [1] [2] [3], have already been done for the Apache server performance analysis used with the OS Free Berkeley Software Distribution (Free BSD). The performance assessment tools “Webstone” and “bsdstar” were used to analyze the effects of three optimization settings on the metrics of Web server performance on the one hand, and the various system resources, on the other hand. A simple queue model representing the behavior of a Web server saturated by using the mean value analysis algorithm has been also presented.

Other authors [4], have used a performance model based on the theory of queues (or LQN: Layered Queueing Network) to study the performance of an Apache-PHP Web application with PostgreSQL as the database server. The performance assessments were done by obtaining load test measures and resolving the LQN model. Model validation is realized by comparing the model results with those of load tests. This Apache performance analysis is carried out to determine the configuration of the system facilitating the identification of sources of bottlenecks.

Other studies [5], have proposed a method to conduct an analysis of Web applications. The behavior model is first built from log file after the user navigation and an expanded state diagram is extracted from this log file in order to obtain a Markov model. The five indexes that are used to measure performance are: service response time, service path length, use of services, the implementation rate of the service and the access error rates. The results of this performance analysis would have provided a suggestion for improving the design of Web applications and optimize services.

More recently, in the example of [6], Web server’s performance study remains a topic of interest. One solution is to optimize the database management system used by Web servers so that improve the performance of the latter in terms of response time. This is relatively a laborious task that the optimal values of these parameters depend on the behavior of Web users and the type and size of the requested information.

This paper examines the database server’s influences on Web server performances by using the neural network model. In fact, databases are particularly resource-intensive. In this work, we used MySQL and PostgreSQL as database servers because of their popularity and licensing [7] [8]. Section 2 concerns the analysis of the influences of the MySQL and PostgreSQL database servers on the performances of the Apache Web server in terms of mean response time. The

parameters taken into account are: the number of instances of concurrent client connections, the number of tables requested by the SQL queries of the web pages to be displayed, and the number of records to be displayed. The same database structure was used for both database server types and in total, we performed 30,000 tests (15,000 tests for each of the two types of database server) [Section 3]. The results of these tests were subsequently used as *inputs* for the neural network modeling phase using the SNNS (Stuttgart Neural Network Simulator) tool. Section 3 deals with the neural network modeling phase of Web server performance. We used the multilayer perceptron model [9]. We used supervised learning by using the Back-Propagation algorithm, and to validate the learning outcomes [1] [10], we used the cross-validation method.

The research presented in this work is divided into two main parts. The first part is about performance analysis of the Web server; and the second part is modeling the performance of the web server using Neural Networks models.

## 2. Web Server Performances Analysis

### 2.1. Experimental Environment

We performed all our experiments in localhost and no formatting was done on the web pages; in order to avoid the possible influences of the network architectures on the results and not to affect the loading time of the web pages.

During experimentations, we used an ASUS K53SV laptop that has a 4 GB RAM and a Core i5-2410M processor (2.30 GHz  $\times$  4) on which we have installed Ubuntu 14.0432 bit Linux.

For both types of database server (MySQL 5.5 and PostgreSQL 9.1), we used the “employees” database downloaded from the official MySQL site because it provides a combination of a large database of about 160 Mo spread over six separate tables with 4 million records in total. Specially designed for researchers, the “employees” database also includes a series of tests to ensure the integrity of the loaded data. In order to avoid affecting the data integrity of used databases, we only used the SELECT query that was designed to display the data.

We used Apache 2.4 as HTTP server, PHP 5.5 as programming language for Web pages, Apache Benchmark 2.3 as assessment tool and SNNS 4.2 for the neural network modelization.

We adopted the test plans that the details of which are in **Table 1**.

Where  $\beta = \mathbf{M}$  for MySQL and  $\mathbf{P}$  for PostgreSQL. For each **Test**  $\beta_{ij}$ , (where  $i$  representing the number of simultaneous connection instances of clients varies from 1 to 5 occurrences, and  $j$  which represents the number of tables affected by SQL queries varies from 1 to 3 tables). We have developed **page<sub>x</sub>.php** web pages (where  $x$  varies from 1 to 1000 according to the number of records to be displayed for each test).

We have also developed Shell scripts to plan the execution of all these tests and collect all the 30,000 results, which in turn have served as inputs for the modeling phase using neural networks.

**Table 1.** Testing plan.

		Database server $\beta$				
		Number of simultaneous connection occurrences of clients				
		1	2	3	4	5
Number of tables affected by SQL queries	1	Test $\beta_{11}$ Nb of record 1 to 1000	Test $\beta_{12}$ Nb of record 1 to 1000	Test $\beta_{13}$ Nb of record 1 to 1000	Test $\beta_{14}$ Nb of record 1 to 1000	Test $\beta_{15}$ Nb of record 1 to 1000
	2	Test $\beta_{21}$ Nb of record 1 to 1000	Test $\beta_{22}$ Nb of record 1 to 1000	Test $\beta_{23}$ Nb of record 1 to 1000	Test $\beta_{24}$ Nb of record 1 to 1000	Test $\beta_{25}$ Nb of record 1 to 1000
	3	Test $\beta_{31}$ Nb of record 1 to 1000	Test $\beta_{32}$ Nb of record 1 to 1000	Test $\beta_{33}$ Nb of record 1 to 1000	Test $\beta_{34}$ Nb of record 1 to 1000	Test $\beta_{35}$ Nb of record 1 to 1000

## 2.2. Experimental Results for MySQL Database Server

**Figure 1** shows the result during  $M_{1j}$  tests, when the occurrences number increases from 4 to 5 occurrences in  $M_{14}$  and  $M_{15}$  experiences, we realize that the Web server requires considerably more time to fulfill requests at querying the unique table “employees”. While the server means response time for  $M_{14}$  test is equal to **2.75 ms**, it is **3.66 ms** for  $M_{15}$ .

**Figure 2** records the averages of server response time for tests  $M_{21}$ ,  $M_{22}$ ,  $M_{23}$ ,  $M_{24}$  and  $M_{25}$  which are respectively **1.40**, **1.73**, **2.01**, **2.72** and **4.07 ms**.

We have summarized the results of experiments  $M_{3j}$  in **Figure 3**. Note that the behavior of the Web server during these experiments is divided into three parts. Indeed, the mean response times for  $M_{33}$  and  $M_{34}$  is almost doubled than that of  $M_{31}$  and  $M_{32}$ . This means response time for  $M_{35}$  tests stays still nearly double that of  $M_{33}$  and  $M_{34}$ .

## 2.3. Experimental Results for PostgreSQL Database Server

It is observed in **Figure 4** that the connections occurrences are between 1 and 4 occurrences, the Web server mean response time stays constant because they are between **5** and **10 ms**. In contrary, when the number of occurrence is 5, the observed mean response time of the server is becoming more and more slow to satisfy the requests, and we record response times between **11** and **17 ms**.

In **Figure 5**, the server behavior doesn’t change in terms of response time. Indeed, these response times is generally comprising between **5** and **9 ms**. In contrary, the server behavior tends to be considerably slow down during the  $P_{24}$  test, because when the number of recording passes of **900** to **1000 records**, the server response time varies from **8** to **12 ms**. These mean responses time have almost doubled during the  $P_{25}$  test because if the average response time is equal to **7.44 ms** for  $P_{24}$  tests, and it is **14.60 ms** for  $P_{25}$  experiments.

In **Figure 6**, it is noticed that the Web server behavior is divided into two parts. The first part corresponds to  $P_{31}$ ,  $P_{32}$ ,  $P_{33}$  and  $P_{34}$  tests, where the mean server response times are generally between **4** and **9 ms**. The second part is  $P_{35}$

experiments where response times are between **12** and **22 ms** with an average of **16.07 ms**.

In this Section 1, we presented an analysis of Web server performance in terms of mean response time where MySQL and PostgreSQL here used as the

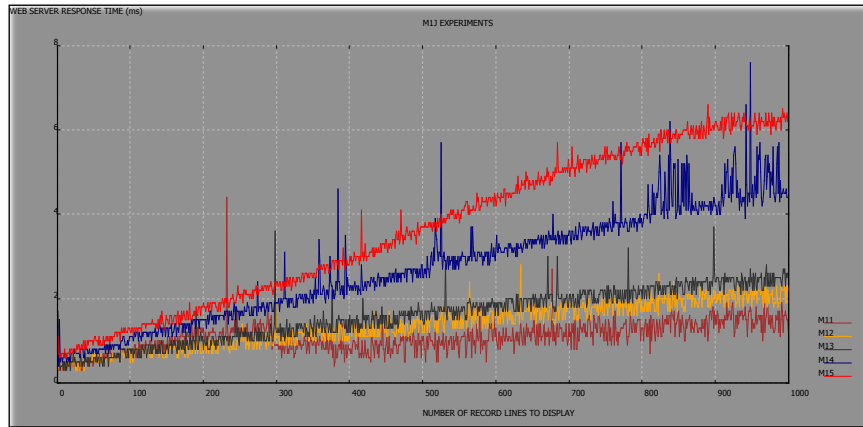


Figure 1. Apache Web Server means response time during  $M_{1j}$  experimentations.

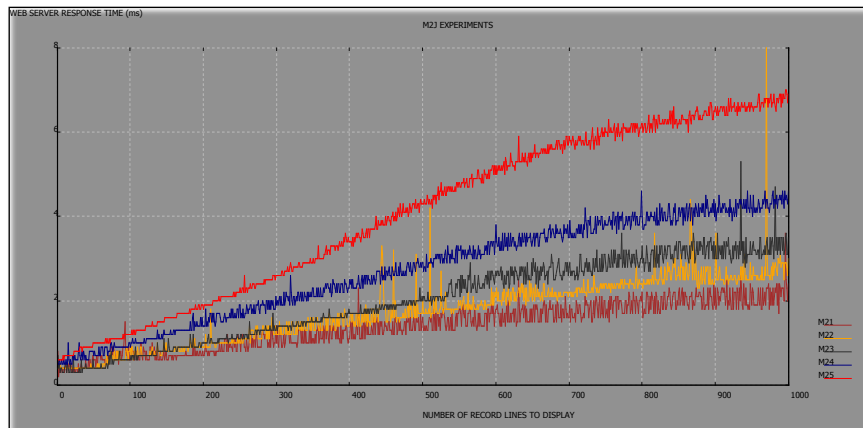


Figure 2. Apache Web Server means response time during  $M_{2j}$  experimentations.

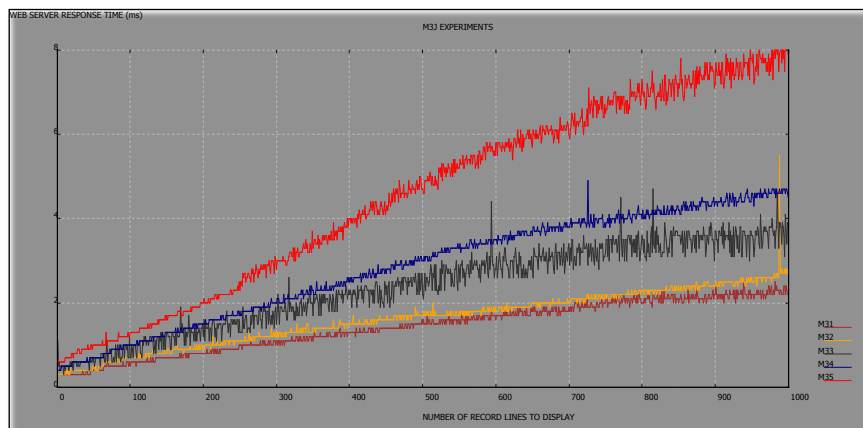


Figure 3. Apache Web Server means response time during  $M_{3j}$  experimentations.

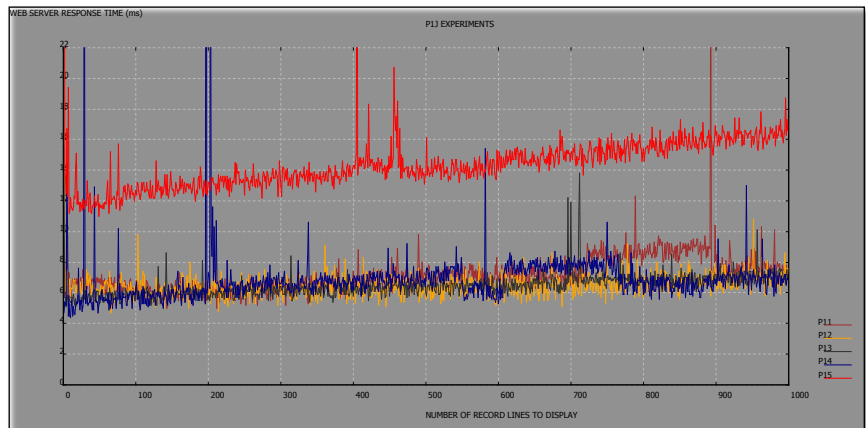


Figure 4. Apache Web Server means response time during  $P_{1j}$  experimentations.

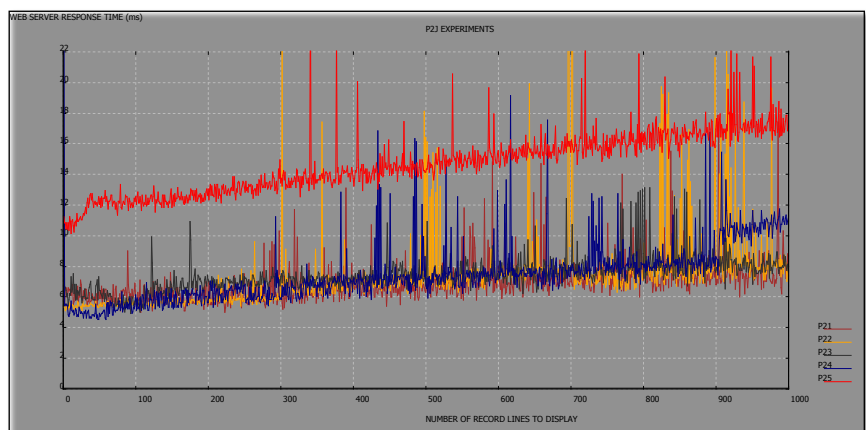


Figure 5. Apache Web Server means response time during  $P_{2j}$  experimentations.

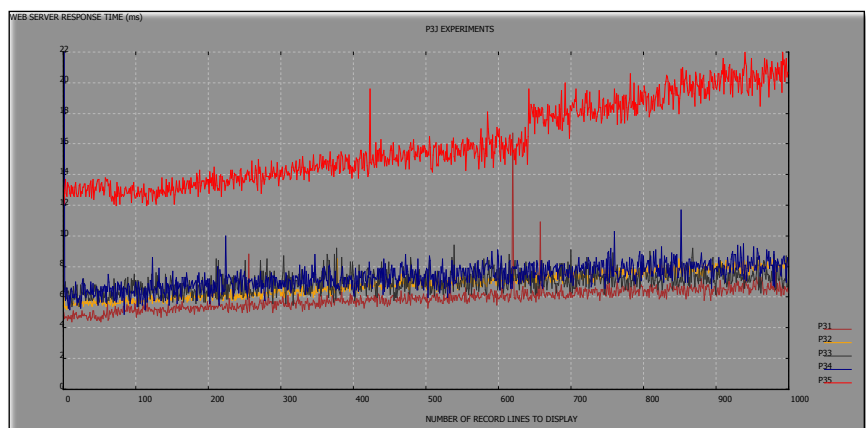


Figure 6. Apache Web Server means response time during  $P_{3j}$  experimentations.

database server. The Apache Benchmark tool which is an HTTP server performance testing tool has allowed us to the custom simulations during the various experiments carrying out the examination of the Web server behavior. It was equally possible to analyze the impact of some SQL queries settings on the Web

server behavior. Indeed, changes in these parameters affect a *non-linear* web server mean response time. The modeling of these non-linear characteristics of the relationship between the parameters and server response time requires “*smart*” mechanisms; hence the concept of “*smart system*”. We’ll treat in the Section 2, the *neural networks modeling* for predicting the average response times from the Web server performances based on the number of clients, number of queried tables by SQL queries, and the number of records to display.

### 3. Neural Network Modeling

This second section addresses the phases of modeling Web server performance in terms of average response time using neural network models. We used SNNS version 4.2 [11] which is a neural network simulator originally developed at the University of Stuttgart previously designed for Unix workstations. The simulator core operates on the internal data structures of the neural network, and performs all learning operations. SNNS can be expanded by the user to set the activation functions, input functions, learning functions that are written in C program, and subsequently linked to the simulator core. SNNS can generate C code from the trained network.

As a neural network architecture, we used a MLP (or *MultiLayer Perceptron*) model with 3 layers. An input layer with 3 neurons that correspond to the 3 parameters taken into account during the performance analyzes in Section 2. Then, a hidden layer with 2, 4, 8, 16 and 32 neurons. And an output layer of a neuron corresponding also to the average response times of the Web server obtained during the performance analyzes of the previous section. We used *Supervised Learning* [1], by using the *Back-Propagation* algorithm; and to validate the learning outcomes, we used the *cross-validation* method [12].

We performed much training with many possible combinations for the choices parts of the learning basis. In our case, so we calculated  $C \frac{8}{10} = 45$  independent learning for the evaluation of the generalization error. The *MSE* or “*Mean Squared Error*” was used to estimate the neural models performance. In this experiment, each learning basis contains 12,000 examples for MySQL and PostgreSQL database servers, and 3000 examples for each validations.

#### Training Results

Both simulation results obtained with the SNNS tool are presented in this section. The first simulation corresponds to the response time of the MySQL database server, while the second when using PostgreSQL. The three parameters of the input vector are: number of occurrence of simultaneous client connections (*nb\_occ*); the number of tables queried by SQL (*nb\_tbl*); and the number of records to display on Web pages (*nb\_rec*). We have only one output, the Web server mean response time:  $\theta_m$  for MySQL and  $\theta_p$  for PostgreSQL. It is therefore to learn the relationships:  $(nb\_occ, nb\_tbl, nb\_rec) \rightarrow \theta_m$  for MySQL and

$(nb\_occ, nb\_tbl, nb\_enreg) \rightarrow \theta_p$  for PostgreSQL.

The learning outcomes of the Web server mean response time using the MySQL and PostgreSQL servers database for MLP with 2, 4, 8, 16 and 32 hidden neurons are summarized in **Table 2**. The “Architecture” columns show the number of parameters of the architecture of the MLP. The MSE columns provide the MSE estimation measured on the 45 independent learning of cross-validation.

When learning networks using the foundations of examples from the performance test results of the server using MySQL and PostgreSQL, it is found that the network whose architecture is 3/16/1 turns out the most efficient. Indeed, 3/16/1 architecture yields sufficiently small MSE: 0.0015108 for MySQL and 0.44738 for PostgreSQL.

#### 4. Discussion

The various research works related to Web server performance are in different aspects. If [1] focused on studying the effects of various parameters of a Web server on its performance metrics, [13] analyzed the effect of the use of a dynamic workload on Web performance indicators. To do this, they have evaluated an e-commerce site with typical scenarios using different workload levels. [14] conducted the modeling and control of a server studying the influence of the configuration of a server on the contradictory aspects of performance and service availability. They have developed two admission control laws designed to optimize server configuration while ensuring a stress on performance. [4] has modeled the web application performance using the queue theory. For this, they have used a Web application using Apache and PHP and PostgreSQL as its database. The performance evaluation is done by comparing the model results with the results of the load test.

In our case, we examined the influence of the MySQL and PostgreSQL database servers on Apache Web server performance using neural network modeling. To do this, we started by analyzing the mean response time of the Web server based on: the number of simultaneous connection instances of clients, the number of tables requested by SQL queries, and the number of records to display. Then, we used the results obtained for the neural modeling phase with the “Stuttgart Neural Network Simulator” tool.

**Table 2.** MSE of different models.

MySQL		PostgreSQL	
<i>Architecture</i>	<i>MSE</i>	<i>Architecture</i>	<i>MSE</i>
03/02/2001	0.012828	03/02/2001	0.014817
03/04/2001	0.003253	03/04/2001	0.005559
03/08/2001	0.001689	03/08/2001	0.004564
3/16/1	0.001511	3/16/1	0.004474
3/32/1	0.012828	3/32/1	0.014817



## 5. Conclusions and Perspective

This research work has been specially oriented to the evaluation problem and neural network modeling of the Web server performance using MySQL and PostgreSQL. It was possible to study the influence of some parameters of the database management system used by the Web server; in particular, according to the number of customers, the size of the information requested and the number of records to display. We created a learning base from obtained data during the Web server performance analysis. The SNNS (Stuttgart Neural Network Simulator) as neural network simulation tool was used for the construction of two neural networks for the prediction of the Web server response time. The first network is for predicting the web server mean response time using MySQL, while the second, for a Web server using PostgreSQL.

The interpretations of the results obtained allowed us to state that for the two types of database server used, the average response times observed are much faster for MySQL than for PostgreSQL to a few tens of milliseconds. At the same time, we observed significant variations in these average response times for MySQL. While for PostgreSQL, the variation of these average response times of the Web server is more stable.

It would be interesting to increase the values of the parameters that we have taken into account in a very significant way in order to broaden our scope of analysis. The Apache Web server and the two database servers we used, were used with the default configuration values. Indeed, the proposal of a model taking into account the different configurations of the Web server, the database servers, etc. remains a challenge.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Rafamantanantsoa, F. (2009) Etude des performances d'un serveur web et d'un réseau local sans fil en utilisant les techniques neuronales. Université Blaise Pascal-Clermont-Ferrand II, Français, 121 p.
- [2] Kamaruddzaman, K.A. (2009) Phase II: Comparison Report on MySQL and PostgreSQL. Malaysian Public Sector Open Source Software Programme, OSCC Report, 67 p.
- [3] Gaspard, G., Jachniewicz, R., Lacava, J. and Meslard, V. (2009) Équilibrage de Charge et Haute Disponibilité pour applications Web Ruby On Rails. LIP Asrall, 47 p.
- [4] Shoaib, Y. and Das, O. (2011) Web Application Performance Modelling Using Layered Queuing Networks. *Electronic Notes in Theoretical Computer Science*, **275**, 123-142. <https://doi.org/10.1016/j.entcs.2011.09.009>
- [5] Waswani, V. (2005) PHP and PostgreSQL. Developer Shed, Melonfire, 20 p.
- [6] Walker, J.D. and Chapra, S.C. (2014) A Client-Side Web Application for Interactive Environmental Simulation Modeling. *Environmental Modelling & Software*, **55**,

---

49-60. <https://doi.org/10.1016/j.envsoft.2014.01.023>

- [7] Mauchle, F. (2008) Database Replication with MySQL and PostgreSQL Software and Systems. University of Applied Sciences Rapperswil, Switzerland, 11 p.
- [8] Conrad, T. (2004) Postgresql vs. Mysql vs. Commercial Databases: It Is All about What You Need. 5 p.
- [9] Parizeau, M. (2012) Le perceptron multicouche et son algorithme de rétro propagation des erreurs. Département de génie électrique et de génie informatique, Université Laval, Québec, 8 p.
- [10] Cornec, M. and Bertail, P. (2009) Validation Croisée et Modèles Statistiques Appliquées. Thèse Université Paris X, Nanterre, 47 p.
- [11] Zell, A., Mamier, G., Vogt, M., Mache, N., Hübner, R., Döring, S., Herrmann, K.U., Soyez, T., Schmalzl, M., Sommer, T., Hatzgeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., Wieland, J. and Gatter, J. (2008) Stuttgart Neural Network Simulator User Manual, Version 4.2. 350 p.
- [12] Chiquet, J. Validation Croisée Pour le Choix de Paramètre de Méthodes de Régularisation. Module MPR—Option Modélisation, 8 p.
- [13] Peña-Ortiz, R., Gil, J.A., Sahuquillo, J. and Pont, A. (2013) Analyzing Web Server Performance under Dynamic User Workloads. *Computer Communications*, **36**, 386-395. <https://doi.org/10.1016/j.comcom.2012.11.005>
- [14] Malrait, L., Bouchenak, S. and Marchand, N. (2009) Modélisation et contrôle d'un serveur. RenPar'/SympA'13/CFSE'7, Toulouse, 12 p.