Editorial

Tools for Performance Evaluation

Corresponding Author:

Maria Calzarossa

Dipartimento di Informatica e Sistemistica

Università di Pavia

Via Ferrata 1

I–27100 Pavia, Italy

Phone: +39 382 505522 or 505350

Fax: +39 382 505373

E–mail: mcc@alice.unipv.it

Editorial

Tools for Performance Evaluation

Maria Calzarossa
Dipartimento di Informatica e Sistemistica
Università di Pavia
Via Ferrata 1
I–27100 Pavia, Italy
mcc@alice.unipv.it

Raymond Marie
IRISA
Université de Rennes 1
Campus Universitaire de Beaulieu
35042 Rennes cedex, France
marie@irisa.fr

Performance evaluation is a multi–faced problem involving the various components which contribute in different ways to the overall performance of a system. Designers and capacity planners have to meet more demanding customer expectations of performance. A systematic approach towards performance evaluation is based on the construction of a system model which in turn relies on a workload model. This means that measurement techniques have to be applied together with modeling technique.

However, performance evaluation is not an easy task if the designer does not have software tools or at least some toolkit to help her/him. Having tools available for performance evaluation increases the productivity not only by saving time dedicated to the task itself but also by decreasing the production error rate. So developing software tools for performance evaluation seems of high benefit.

When looking for the trends of this activity, we can discern two new challenges; the first one relies on the fact that it is necessary to be able to model more and more sophisticated architectures such as fully distributed systems or statistical multiplexing of high speed networks. Also, as this is now (often) the case for the production of the documentation in software engineering, computer scientists would like to obtain the production of the performance model as a by–product of the project specifications; and research has moved to this direction. However, if we leave everything to the automatic model production, we can take the risk of generating only huge simulation models which are not usually the best models to capture the main features of the behavior of the system.

Therefore, the second challenge is to be still able to realize good aggregated models. For example, going back to the 70's, if we assume the availability of tools and of computation power,

can we imagine how an automated process would have modeled complex mainframes by simple product form queueing network models of three or four stations?

A possible solution to be able to cope with the increasing complexity of the systems will be to develop user friendly tools which allow the analysis of large quantities of measurements coming from today's production environments (e.g., Web servers, parallel systems) and which implement more and more hierarchical approaches for system modeling.

Moreover, we believe that performance evaluation is more art than science, and developing this art requires lot of practice. Hence, whatever powerful the tools will be, we will always need the human talent to introduce the part of the art which is inherent in workload characterization and system modeling and to provide new methods and theories to respond to new applications and to new complexities.

To be more specific, this Special Issue, which contains the extended versions of the best papers of the "9th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation" (Saint–Malo, France, 3-6 June 1997)[1], covers the hot topics in the field of performance evaluation. One of the four articles presented here is primarily concerned with Web server performance, one with workload characterization of I/O intensive parallel applications, and two with tools for performance/availability modeling and for solving large Markov models.

The paper "Web Server Performance Measurement and Modeling Techniques" analyzes the performance of large–scale commercial Internet and Intranet Web servers with the objective of creating a model to be used for capacity planning. The approach adopted in the study is based on the analysis of a large quantity of measurements collected from Web servers over many months. Custom instrumentation has been incorporated into the Web server process in order to derive metrics, such as, server response and service demand, and client residence time at the server. The predictions of client response times under different Web server and network configurations are based on Layered Queueing Network models.

The paper "Workload Characterization of Input/Output Intensive Parallel Applications" deals with the study of parallel I/O behavior and access patterns of five scientific applications. The applications have been executed on a 512-node Intel Paragon equipped with two different I/O configurations. The study shows that, although the applications are algorithmically different, there

[1] R. Marie, B. Plateau, M. Calzarossa, and G. Rubino, editors. *Computer Performance Evaluation – Modelling Techniques and Tools*, volume 1245 of *Lecture Notes in Computer Science*. Springer, 1997.

are a few common characteristics in their I/O activity. Parallel I/O is typically bursty. The access patterns range from simple sequential to interleaved. The size of the I/O requests can differ dramatically from few bytes to several kilobytes each. All these observations provide good insights for improving the performance of parallel file systems.

The paper "Performance/Availability Modeling with the TANGRAM-II Modeling Environment" presents a modeling environment tool for the analysis of performance and availability metrics. The system being modeled is represented by a set of objects which interact by sending and receiving messages. Techniques for both steady state and transient solutions are provided within TANGRAM-II. The modeling process implemented by TANGRAM-II is presented on three examples, dealing with a jitter control mechanism for a communication network, a multiplexor for voice and data and a dependability model, respectively.

The paper "An Efficient Disk-Based Tool for Solving Very Large Markov Models" describes a tool for solving general large Markov processes which result when modeling real computer systems and networks. The tool implements a Block Gauss Seidel solver and stores the state transition rate matrix of the Markov process on disk. The tool architecture is based on two cooperating processes, one schedules disk I/O and the other one does computation. Two different implementations of the solver are presented, a disk–based Block Gauss Seidel solver and a similar solver which uses file compression. Two large models, dealing with a Kanban manufacturing system and a Courier protocol stack executing on a VME bus–based multiprocessors, are presented to illustrate the benefits of using the tool. The implementation based on file compression, although less general, seems faster than any other existing tools.

Finally, as Guest Editors of this Special Issue, we would like to thank the authors of these papers for their valuable contribution in the field of performance tools and for the effort they put in the preparation of their manuscripts.