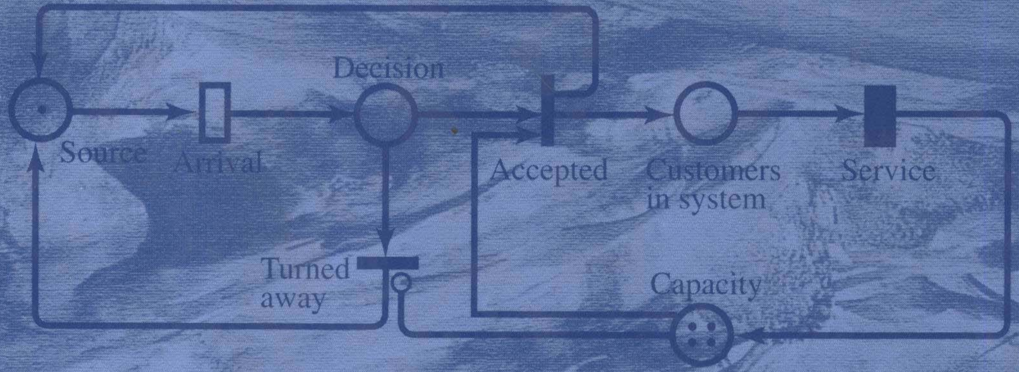
WILEY

Performance Modelling with Deterministic and Stochastic Petri Nets



CHRISTOPH LINDEMANN

Performance Modelling with Deterministic and Stochastic Petri Nets

Christoph Lindemann
GMD Research Institute FIRST, Berlin, Germany

JOHN WILEY AND SONS
Chichester · New York · Brisbane · Toronto · Singapore

National 01243 779777
International (+44) 1243 779777

e-mail (for orders and customer service enquiries): cs-books@wiley.co.uk.
Visit our Home Page on http://www.wiley.co.uk
or http://www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

WILEY-VCH Verlag GmbH, Pappelallee 3,
D-69469 Weinheim, Germany

Jacaranda Wiley Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01,
Jin Xing Distripark, Singapore 129809

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0 471 97646 6

Produced from camera-ready copy supplied by the author
Printed and bound in Great Britain by Bookcraft (Bath) Ltd
This book is printed on acid-free paper responsibly manufactured from sustainable forestation, for which at least two trees are planted for each one used for paper production.

Performance Modelling with Deterministic and Stochastic Petri Nets

Preface

This text provides an up-to-date treatment of the fundamental techniques and algorithms for numerical analysis of deterministic and stochastic Petri nets, a particular stochastic modelling formalism, and the application of this modelling formalism to performance analysis of parallel computer architectures. The material grew from work conducted in the last six years at the Technische Universität Berlin, at the GMD Institute for Computer Architecture and Software Technology (GMD FIRST) in Berlin, and at the IBM Almaden Research Center in San Jose, CA.

Performance evaluation is an applied discipline of computer science which has its roots in queuing theory and teletraffic engineering. A discussion on the evolution of the discipline performance evaluation between 1965 and 1990 appeared in 1991 as an editorial of the journal *Performance Evaluation* [Rei91]. The following historical review continues in this light with emphasis on newly proposed modelling formalisms and the evolution of their numerical solution techniques in the last fifteen years.

Methodological research work in the 1970s emphasised numerical algorithms for steady-state analysis of queuing networks with product-form solution. Typically, such algorithms are based on simple iterative schemes. The first class of such schemes successively considers individual queues of the queuing network, i.e., they exploit the property that the solution has a product form. The second class of such simple iterative schemes considers the entire network starting with a single customer and iterates on the number of customers until the desired customer population is reached. At that time computer equipment was a mainframe connected to terminals or a punchcard reader. Since main memory was quite limited in these computer systems, the amenability of numerical solution methods was restricted to the class of queuing networks with product-form solution. Major results in this decade include the BCMP theorem named after its authors, Baskett, Chandy, Muntz and Palacios [BCM+75], Norton's theorem for queuing networks [CHW75], and the Mean Value Analysis algorithm [RL80].

In the 1980s, theoretically known numerical methods for transient and stationary analysis of Markov chains were extended and refined in order to put them to work.

Markov chains have more modelling power than product-form queuing networks, although they allow only the representation of exponentially distributed delays. Algorithms for steady-state analysis are based on direct or iterative methods for the solution of a sparse linear system of equations whereas algorithms for transient analysis employ iterative techniques for solving a system of ordinary differential equations. Computer equipment available at that time consisted of early personal computers and first-generation workstations. Such workstations could compute transient solutions of Markov chains up to a few thousands of states and the steady-state solution of Markov chains up to several tens of thousands of states. A major result in this decade was the introduction of stochastic Petri nets [Mol82] and generalised stochastic Petri nets as high-level representations of Markov chains [ABC84]. Moreover, Ajmone Marsan and Chiola proposed deterministic and stochastic Petri nets [AC87] which include exponentially distributed and deterministic delays. Further important results in that decade constituted the practical discussion of the randomisation technique [GM84] in conjunction with the development of a stable numerical method for computing Poisson probabilities [FG88]. Later, it was shown that depending on structural properties, the stochastic process underlying a deterministic and stochastic Petri net is either a Markov regenerative process [CKT94] or a generalised semi-Markov process [LS96].

Methodological research work in the 1990s is focusing on time- and memory-efficient numerical algorithms for transient and steady-state analysis of Markov regenerative processes and more general stochastic processes. Algorithms for the analysis of these stochastic processes do not only involve numerical solution of one linear system of equations. Depending on the stochastic process, their analysis also involves numerical solution of a system of ordinary differential equations and/or solution of a system of stochastic integro-differential equations. At present, powerful personal computers and very powerful workstations with several RISC processors and hundreds of Mbytes of main memory are widely available. Numerical methods have been implemented that can compute steady-state solutions of a restricted class of Markov regenerative processes with up to several hundred thousand states and moderate stiffness in the delay parameters on a modern workstation in one hour of CPU time (see Chapter 4). At present, there exists no large-scale implementations of a numerical method for more general stochastic processes. However, methodological results presented in Chapter 5 show great promise that analysis of a quite general stochastic process (i.e., the generalised semi-Markov process with exponential and deterministic events) can be conducted in parallel on a cluster of four to six modern workstations in a similar amount of time.

So far, two textbooks have been devoted to analysis and applications of generalised stochastic Petri nets: [ABC86] and [ABC+95]. Furthermore, a few recent textbooks contain a chapter dealing with some classes of stochastic Petri nets: [BDM+94], [Kan92], and [VN92]. To the best of the author's knowledge, no monograph on the analysis and application of deterministic and stochastic Petri nets has been published so far. This text tries to fill this gap. It covers methodological results on the numerical analysis of deterministic and stochastic Petri nets and their application to performance modelling of parallel computer architectures. The author considers himself as an engineer who has been fortunate to work with gifted mathematicians. As a consequence, the exposition of the text is engineering-oriented, i.e., intuitive explanations for mathematical results are provided instead of rigorous mathematical proofs. The intended audience are computer scientists, applied mathematicians, and electrical engineers who are interested in performance analysis of computer systems and communication networks. The text is accompanied by a CD-ROM containing the object code of the software package DSPNexpress for several hardware platforms and specification files of a variety of deterministic and stochastic Petri net models. Readers, who are interested in further information on DSPNexpress, are welcome to visit the DSPNexpress Web page mentioned above and to contact the author.

ACKNOWLEDGEMENTS

Several people contributed in one or more ways to this text. First, I would like to express my sincere gratitude to Professor Wolfgang Giloi for his continuous support of my career and his impact on my work. The participation in research projects he initiated since 1990 and the collaboration with him and his co-workers introduced me to the challenging and rapidly emerging field of computer architecture. I also owe a significant dept to Professor Kishor Trivedi for his interest in my research work and our joint work on numerical methods for acyclic Markov chains.

The preparation of this text has also benefited from joint work and technical discussions with a number of people. In particular, I would like to mention Heinz Beilner, Gianfranco Ciardo, Reinhard German, Peter Glynn, Gerd Heber, Holger Hermanns, Manish Malhotra, Samuel Pletner, Friedrich Schön, and Gerald Shedler. Moreover, it is my pleasure to acknowledge the contributions and help of numerous (former) students of the Technische Universität Berlin, who worked with me in the last six years. In particular, I am very grateful to Tobias Bading, Enrik Baumann, Stefan Busse, Christian Lühe, Martin Müller, and Armin Zimmermann. The software package DSPNexpress would not have become reality without their excellent programming capabilities and their dedication to this project. Furthermore, I would like to thank Michael Bäuning and Matthias Weiss for developing programs for the two examples presented in Chapter 5.

Last, but by no means least, the financial support received from the German Research Council (DFG), from the GMD Research Institute FIRST, and from the IBM Almaden Research Center is gratefully acknowledged.

This text is dedicated to
Professor Wolfgang K. Giloi
– an outstanding researcher
with whom I was fortunate to work.

Ch. L.

Contents

Part I Introduction to Performance Modelling

This introductory part states fundamental concepts of computer systems performance evaluation in order to help the reader to get an appreciation for the results presented in Parts II and III of this text. Chapter 1 describes different methods for conducting performance studies of computer systems and states their pros and cons. In particular, we strongly argue for the employment of stochastic modelling to support early design stages of computer systems. Furthermore, we emphasise the extended applicability of numerical methods for the stationary analysis of stochastic models due to the rapid progress in computer hardware technology and the availability of efficient numerical algorithms. In Chapter 2, we illustrate how to employ queuing networks, stochastic Petri nets, and stochastic process algebras, three particular stochastic modelling formalisms, for analysing the performance of a small-scale multiprocessor system.

1

Performance Evaluation of Computer Systems

1.1 WHY PERFORMANCE MODELLING?

Today's computer systems are rapidly evolving and include typically complex system features. Recent progress in semiconductor technology will soon allow the production of chips for superscalar processors with peak performances comparable to that of the bus-structured multiprocessor systems of ten years ago. As a consequence of this rapid progress, thorough understanding and evaluation of the behaviour of such complex systems is necessary to meet cost and performance requirements through the life cycle of the system. In particular, during the design and implementation phase of a computer system the evaluation of different design alternatives is essential in order to achieve optimal performance for given cost.

Figure 1.1 shows a straightforward design cycle for a computer system using a top-down approach. Starting from a high-level design, e.g., as written in a research proposal, individual system components are designed, and subsequently a detailed design of a novel computer system is derived. Without the support of performance modelling, computer architects have to rely on their experience and intuition for the development of this detailed design. Subsequently, based on the detailed design a software prototype is implemented which mimics the behaviour of the novel computer system and estimates for performance measures are derived using *trace-driven* or *execution-driven simulation*. After the hardware development for a novel computer system has been completed and the system has been installed, *measurements* can be conducted for evaluating system performance. Measurements

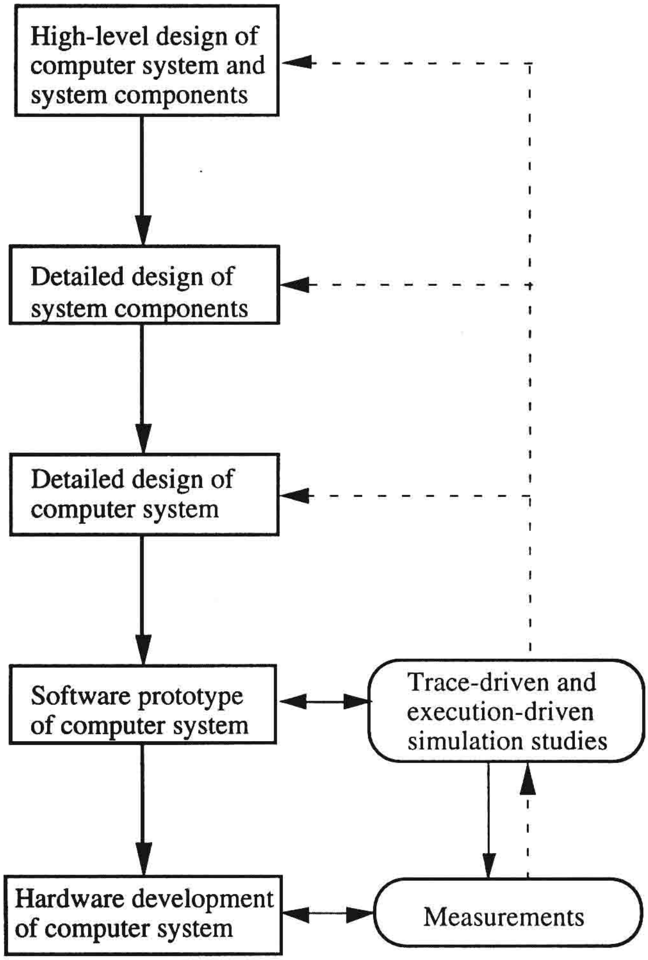


Figure 1.1. Computer system design without support of performance modelling

as well as trace-driven and execution driven simulation belong to the class of deterministic performance evaluation techniques and are briefly discussed in Section 1.2. The dashed lines in Figure 1.1 indicate the possibility of going back to early design stages and redesigning system components due to performance bottlenecks detected by trace-driven and execution-driven simulation studies on the prototype implementation.

If performance is to be considered in the early design stages of a computer system, stochastic modelling must be used because the system is not yet operational, and therefore measurements cannot be conducted. Loosely speaking, a *stochastic model* is an attempt to derive, from the mass of details of the entire system, exactly those which have significant impact on the performance measures of interest. Since a model avoids unnecessary details, it is typically easier to develop and more