

Wolfgang W. Osterhage

# Mathematical Theory of Advanced Computing

 Springer Vieweg

---

# Mathematical Theory of Advanced Computing

---

Wolfgang W. Osterhage

# Mathematical Theory of Advanced Computing

 Springer Vieweg

Wolfgang W. Osterhage  
Wachtberg-Niederbachem, Germany

ISBN 978-3-662-60358-1      ISBN 978-3-662-60359-8 (eBook)  
<https://doi.org/10.1007/978-3-662-60359-8>

© Springer-Verlag GmbH Germany, part of Springer Nature 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer Vieweg imprint is published by the registered company Springer-Verlag GmbH, DE, part of Springer Nature.

The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

---

# Preface

This book is the English translation of Wolfgang W. Osterhage's *Mathematische Algorithmen und Computer-Performance*, Springer Vieweg, Heidelberg, 2016. It has been enhanced by the following items and sections:

- Chapter 5: Jump Transformations: Source code for computer art graphs and additional material in electronic form of 35 realizations of computer art
- Chapter 6: Data Management
- Chapter 7: Quantum Computers.

Initially this book originated from a collection of disparate papers all related to computer performance. Since a comprehensive work concerning performance had already been published (s. References), the challenge was to sort of compile a later supplement of these aspects and deliver it in an acceptable form. There are some special features in this book relating to two new algorithms and database theory, which have not yet been put to the test in the field. We hope to have called forth the interest of software and hardware architects alike.

On this occasion, I would like to express my special thanks to Martin Börger and Sophia Leonhard and their team for their patient support for this project.

Wachtberg-Niederbachem, Germany

Wolfgang W. Osterhage

---

# Contents

<b>1</b>	<b>Introduction</b> . . . . .	1
1.1	Terminology . . . . .	2
1.2	Three Levels . . . . .	2
<b>2</b>	<b>Performance Theory</b> . . . . .	7
2.1	System Performance . . . . .	7
2.1.1	Hardware Parameters . . . . .	8
2.1.2	CPU . . . . .	14
2.1.3	Main Memory . . . . .	19
2.1.4	Disks . . . . .	24
2.1.5	I/O . . . . .	26
2.1.6	Operating System Parameters . . . . .	29
2.2	Application Performance . . . . .	30
2.2.1	Application Parameters . . . . .	30
	Bibliography . . . . .	32
<b>3</b>	<b>Test-Automatisation</b> . . . . .	33
3.1	Introduction . . . . .	33
3.2	Why Automatisation? . . . . .	33
3.3	Comparison Between Manual and Automated Testing . . . . .	37
3.4	Symbolic Execution . . . . .	38
3.4.1	Challenges and Solutions . . . . .	40
3.4.2	Tools . . . . .	41
3.5	Search-Based Automated Structural Testing . . . . .	42
3.5.1	Genetic Algorithm . . . . .	42
3.5.2	Limits . . . . .	43
3.6	Combination of SE and SBST . . . . .	44
3.7	Conclusion . . . . .	44
	Bibliography . . . . .	45

<b>4</b>	<b>Preservation Numbers: A New Approach in Soft Computing . . . . .</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Calculations and Observations . . . . .	48
4.3	The Economy of Calculations . . . . .	49
4.4	Preservation Numbers . . . . .	50
4.5	Definitions . . . . .	50
	4.5.1 Accuracy . . . . .	50
	4.5.2 Efficiency . . . . .	51
4.6	Relationships . . . . .	52
	4.6.1 Accuracy and Efficiency . . . . .	52
	4.6.2 Accuracy, Efficiency and Preservation Numbers . . . . .	53
4.7	Examples . . . . .	54
4.8	Chains of Calculations . . . . .	54
4.9	Logic . . . . .	55
4.10	Applications . . . . .	56
	Appendix . . . . .	58
	References . . . . .	61
<b>5</b>	<b>Jump Transformations . . . . .</b>	<b>63</b>
5.1	Introduction . . . . .	63
5.2	Jump Transformations . . . . .	64
5.3	n-Tuple Transformations . . . . .	64
	5.3.1 Definitions and Rules . . . . .	65
	5.3.2 Characteristics . . . . .	66
	5.3.3 Formalizations . . . . .	67
	5.3.4 Examples . . . . .	68
	5.3.5 Inter-relationships . . . . .	70
5.4	Random Trigger Transformations . . . . .	72
	5.4.1 Definitions and Rules . . . . .	72
	5.4.2 Example . . . . .	73
5.5	Inter-transformations . . . . .	73
	5.5.1 Inter-relationships Between Two Random Triggers . . . . .	73
	5.5.2 Transformation Between n-Tuple and Random Triggers . . . . .	75
5.6	Applications . . . . .	75
	5.6.1 Sets and Address Spaces . . . . .	75
	5.6.2 Cryptography . . . . .	76
	5.6.3 Computer Art . . . . .	77
	5.6.4 Other . . . . .	78
	Appendix . . . . .	79
	Reference . . . . .	86

---

<b>6</b>	<b>Data Management</b> . . . . .	87
6.1	Artificial Neural Networks . . . . .	87
6.1.1	Introduction . . . . .	87
6.1.2	The Model for Artificial Neural Networks . . . . .	87
6.1.3	Applications . . . . .	89
6.1.4	Reliability . . . . .	90
6.1.5	Multi-layer Neural Networks . . . . .	90
6.2	Data Management Concepts . . . . .	91
6.2.1	Technical Conditions . . . . .	93
6.2.2	Accesses . . . . .	96
6.3	Knowledge Bases and Neuronal Tools . . . . .	97
6.3.1	Introduction . . . . .	97
6.3.2	Knowledge Bases and AI . . . . .	97
6.3.3	Advanced Options for Data Base Management . . . . .	98
6.3.4	Conclusion . . . . .	101
<b>7</b>	<b>Quantum Computers</b> . . . . .	103
7.1	What Is Intelligence? . . . . .	103
7.2	Quantum Computers . . . . .	105
7.3	Implementation . . . . .	106
7.4	Intelligence of Quantum Systems . . . . .	107
	<b>Index</b> . . . . .	109



# Introduction

# 1

This book has to do with performance, but not as a general subject and not in completeness. It is about selected methods and how under certain conditions specific performance aspects can be improved. This means that—contrary to other publications (s. References at the end of the book)—for example considerations concerning process performance do not figure at all. The book is rather technical and partly speculative.

After an introductory chapter, which deals with the basics of computer performance and provides a general overview, there are six chapters dedicated each to specific algorithms and techniques, which may play a role under a performance point of view:

- Symbolic Execution (SE)
- Search-Based Automated Structural Testing (SBST)
- Preservation Numbers as an Instrument of Soft Computing
- Jump Transformations
- Data Management
- Quantum Computers.

SE and SBST are established methods in automatic testing of complex software. They are relevant to ensure and optimize performance during extensive tests. The concept of Preservation Numbers constitutes a new approach in soft computing and could be used in complex mathematical applications, for example for numerical modeling or simulations in the technical-scientific area. Jump Transformations present again a new algorithm, which may enable to generate overlapping numerical spaces on the basis of transformation boundaries, possible to be applied in the design of main memories to create competing address spaces within one and the same memory for example.

In the chapter about Data Management it is proposed to combine advanced data management techniques with artificial neural network technology.

Finally the concept of quantum computers and first realization attempts are introduced against the backdrop of an intelligence definition culminating in the speculation, whether quantum systems as such carry intelligence at all.

---

## 1.1 Terminology

The subject of performance optimization can be divided into three main parts of consideration:

- system performance
- application performance and
- process performance.

All three areas are subject to:

- theory
- measurement
- analysis and
- optimization.

---

## 1.2 Three Levels

When talking about performance people usually refer to system performance only—or to simplify it yet again: to the power of hardware, i.e. processor and main memory. This is the reason, why performance has been neglected during the last decades. At some stage hardware became so cheap that optimizing by programming techniques for example did not seem worthwhile, since manpower became more and more expensive. Hardware was bought and its extensions and as a result systems were running faster again. Or existing systems were configured so comfortably that performance problems just did not happen.

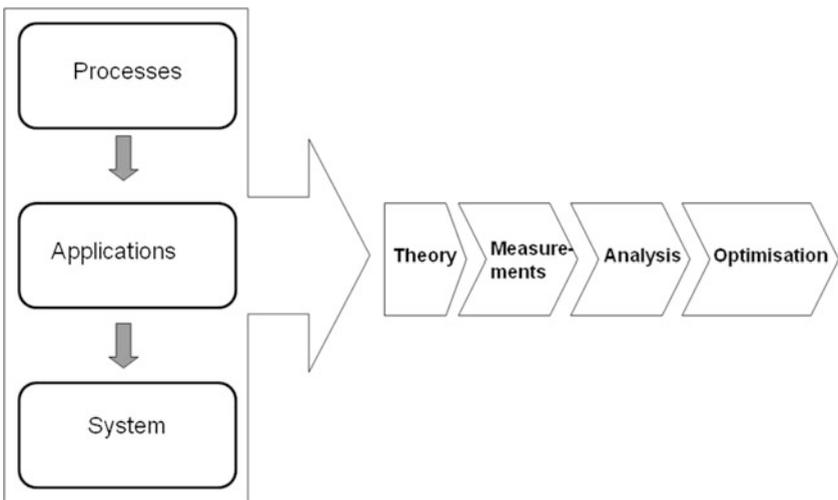
However, end user experience spoke a different language. Negative perceptions of response times did not only play a psychological role but also affected throughput in daily business. However, the ratio between hardware investments to optimization

remained more or less constant over time. The reason for this is that generous hardware resources are exploited equally generously.

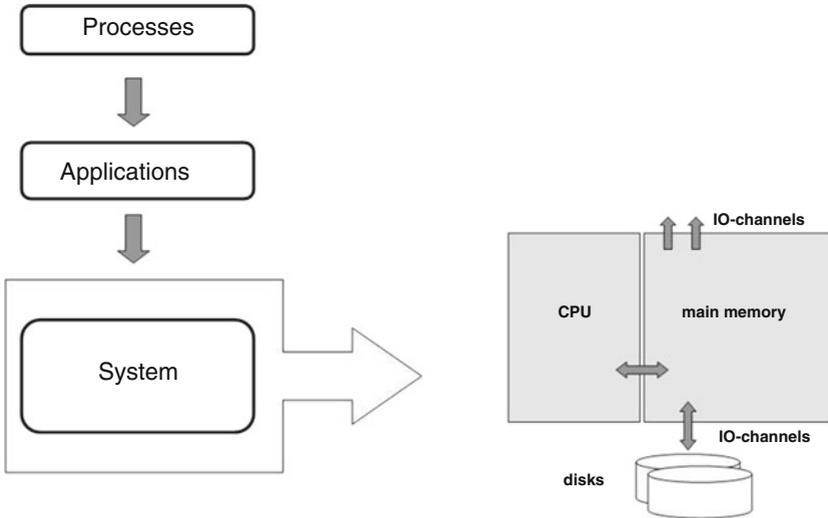
Only 50 years ago no one could afford memory allocations for blanks or binary zeros. Already at the variable declaration level and successively at the address level every single byte had to be considered deliberately. Otherwise major applications would never have taken off the ground. And finally, after the introduction of graphical user interfaces, C++ and Java and their derivatives structured programming in the classically sense was abandoned. Demands on ease of use, end user queries etc. played their part to resurrect old bottlenecks in new garments. In this way the performance debate has become newsworthy again—and this time not only restricted to systems and hardware alone. Although according to Fig. 1.1 the three levels

- system performance
- application performance
- process performance

are addressed one by one, the terminology in use initially refers to system and application performance only.



**Fig. 1.1** Performance dimensions



**Fig. 1.2** System performance elements

System performance includes (Fig. 1.2)

- hardware utilization (memory, processor)
- configuration of system tables
- I/Os

with all procedures and parameters relevant to system management.

Concerning application performance and its analysis one has to take into account the interplay with system resources in detail, considering system resource calls, data management and I/Os for example. Quite generally speaking, if you want to execute applications you need systems to run them. Figure 1.3 shows the overall context. Thus, in performance considerations these two levels cannot be regarded as separate or just existing side by side.

Those elements playing the most important roles when analyzing applications are:

- program structure
- data management philosophy
- GUIs