

# A Parallel Algorithm Synthesis Procedure for High-Performance Computer Architectures

# SERIES IN COMPUTER SCIENCE

Series Editor: Rami G. Melhem

*University of Pittsburgh  
Pittsburgh, Pennsylvania*

---

## ENGINEERING ELECTRONIC NEGOTIATIONS

A Guide to Electronic Negotiation Technologies for the Design and Implementation of Next-Generation Electronic Markets—Future Silkroads of eCommerce

Michael Ströbel

## FUNDAMENTALS OF X PROGRAMMING

Graphical User Interfaces and Beyond

Theo Pavlidis

## INTRODUCTION TO PARALLEL PROCESSING

Algorithms and Architectures

Behrooz Parhami

## OBJECT-ORIENTED DISCRETE-EVENT SIMULATION WITH JAVA

A Practical Introduction

José M. Garrido

## A PARALLEL ALGORITHM SYNTHESIS PROCEDURE FOR HIGH-PERFORMANCE COMPUTER ARCHITECTURES

Ian N. Dunn and Gerard G. L. Meyer

## PERFORMANCE MODELING OF OPERATING SYSTEMS USING OBJECT-ORIENTED SIMULATION

A Practical Introduction

José M. Garrido

## POWER AWARE COMPUTING

Edited by Robert Graybill and Rami Melhem

## THE STRUCTURAL THEORY OF PROBABILITY

New Ideas from Computer Science on the Ancient Problem of Probability Interpretation

Paolo Rocchi

# A Parallel Algorithm Synthesis Procedure for High-Performance Computer Architectures

Ian N. Dunn

*Mercury Computer Systems, Inc.  
Chelmsford, Massachusetts*

and

Gerard G. L. Meyer

*Johns Hopkins University  
Baltimore, Maryland*

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-4658-6      ISBN 978-1-4419-8650-4 (eBook)  
DOI 10.1007/978-1-4419-8650-4

©2003 Springer Science+Business Media New York  
Originally published by Kluwer Academic/Plenum Publishers, New York in 2003  
Softcover reprint of the hardcover 1st edition 2003

<http://www.wkap.nl>

10 9 8 7 6 5 4 3 2 1

A C.I.P. record for this book is available from the Library of Congress

All rights reserved

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording, or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

# Contents

List of Figures	vii
List of Tables	ix
Acknowledgments	xi
1. INTRODUCTION	1
1 Notation and Conventions	2
2 Chapter Organization	3
2. PARALLEL COMPUTING	5
1 Architectures	6
2 Programming Models	9
3 Performance Metrics	10
3. PARALLEL ALGORITHM SYNTHESIS PROCEDURE	13
1 Architectural Model for Algorithm Synthesis	14
2 Synthesis Procedure	15
3 Related Work	25
4. REVIEW OF MATRIX FACTORIZATION	29
1 Givens-based Solution Procedures	31
2 Householder-based Solution Procedures	36
5. CASE STUDY 1: PARALLEL FAST GIVENS QR	41
1 Parallel Fast Givens Algorithm	42
2 Communication Procedures	51
3 Related Work	59
4 Experimental Results	60

6. CASE STUDY 2: PARALLEL COMPACT WY QR	75
1 Parallel Compact WY Algorithm	77
2 Related Work	83
3 Experimental Results	83
7. CASE STUDY 3: PARALLEL BIDIAGONALIZATION	89
1 Parallel Matrix Bidiagonalization Algorithm	90
2 Related Work	93
3 Experimental Results	95
8. CONCLUSION	101
References	103
Index	107

# List of Figures

2.1	A fat tree interconnection network	8
2.2	A hypercube interconnection network	9
2.3	Parallel latency $\mathcal{L}^P$ and sequential latency $\mathcal{L}$	11
2.4	Throughput time $\mathcal{T}_k$	12
3.1	Architectural model for DSM architectures	14
3.2	Source and sink primitives for SH algorithm	16
3.3	Ordering scheme 1 for SH algorithm	20
3.4	Ordering scheme 2 for SH algorithm	21
3.5	Ordering scheme parameterized by $\rho$ and $\psi$ for SH algorithm	22
3.6	Ordering scheme parameterized by $h$ for SH algorithm	23
3.7	Ordering scheme parameterized by $w$	25
4.1	Source data for line fitting example	30
4.2	Line fit using QR factorization	31
4.3	Dependency graph for SFG	34
4.4	Ordering scheme for SFG algorithm	35
4.5	Dependency graph for SH algorithm	38
4.6	Ordering scheme for SH algorithm	39
5.1	PFG: Dependency graph parameterized by $\psi$ and $\rho$	42
5.2	PFG: Parameterized superscalar ordering	43
5.3	PFG: Two adjoining groups of rotations parameterized by $\psi$ and $\rho$	44
5.4	PFG: Superscalar block and parameterization	45
5.5	PFG: Memory hierarchy parameterization	46
5.6	PFG: Synchronization and task indices	48
5.7	PFG: Execution times as a function of $P$ on the HP	65

5.8	PFG: Execution times as a function of $P$ on the SGI	66
6.1	PCWY: Task dependency graph	77
6.2	PCWY: Partitioning and sequencing strategy	80
6.3	PCWY: Partitioning and sequencing strategy	81
7.1	PMB: Phase 1 zero ordering	94
7.2	PMB: Phase 2 zero ordering	95
7.3	PMB: Phase 2 rotation sequence	96
7.4	PMB: Execution times as a function of $h$ on SGI Origin 2000	97
7.5	PMB: Execution times as a function of $h$ on HP V2500	98
7.6	PMB: Execution times as a function of $P$ on HP V2500	99
7.7	PMB: Execution times as a function of $P$ on SGI Origin 2000	99



# List of Tables

5.1	PFG: Minimum execution times on the HP	62
5.2	PFG: Optimal parameter setting on the HP	62
5.3	PFG: Minimum execution times on the SGI	63
5.4	PFG: Optimal parameter settings on the SGI	64
5.5	PFG: Sensitivity to parameter settings on the HP	66
5.6	PFG: Sensitivity to parameter settings on the SGI	67
5.7	PFG: Execution times as a function of $P$ for $m = 3000$ and $n = 1500$	68
5.8	PFG: Execution times as a function of $P$ for $m = 1500$ and $n = 1500$	69
5.9	PFG: Execution times as a function of $P$ for $m = 1500$ and $n = 500$	70
5.10	PFG: Execution times as a function of $P$ for $m = 500$ and $n = 500$	71
5.11	PFG: Execution times for various blends on the HP	72
5.12	PFG: Execution times for various blends on the IBM	72
5.13	PFG: Execution times for various blends on the SGI	73
6.1	PCWY: Execution times for $h = h^*$ and $P = P^*$	84
6.2	PCWY: Experimentally determined optimal values of $h$ and $P$	85
6.3	PCWY: Execution times for $h = 8$ and $P = P^*$	85
6.4	PCWY: Execution times for $h = 12$ and $P = P^*$	86
6.5	PCWY: Execution times as a function $P$	86

# Acknowledgments

This book grew out of research at the Parallel Computing and Imaging Laboratory. Much of the material in this book builds upon research conducted by students and members of the laboratory staff including Yosef Brandriss, Dr. James Carrig, Dr. Mike Pascale, Thomas Steck, and Dakasorn Ubol. Dr. Pascale introduced QR factorization and the critical role it plays in adaptive beamforming (Meyer and Pascale, 1995). Dr. Carrig developed a strategy for designing register and cache efficient linear algebra algorithms. His methodology led to the development of two new QR factorization algorithms (Carrig and Meyer, 1997; Carrig and Meyer, 1999). We express our sincere appreciation to Yosef Brandriss, Thomas Steck, and Dakasorn Ubol for their invaluable assistance in the development of this material as well.

This book references the research of a number of luminaries in the field of linear algebra including Christian Bischof, James Demmel, Jack Dongarra, Kyle Gallivan, Gene Golub, Richard Hanson, William Kahan, Charles Lawson, Robert Schreiber, and Charles Van Loan. Their work created a solid mathematical foundation on which to develop the material in this book.

Finally, we thank our families for their constant encouragement and patience.