INFORMATIONAL DISPERSAL AND PARALLEL PROCESSING

Cambridge International Series on Parallel Computation: 3

INFORMATION DISPERSAL AND PARALLEL COMPUTATION

YUH-DAUH LYUU

NEC Research Institute
Princeton, New Jersey

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York NY 10011–4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 1992
First paperback edition 2004

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

Library of Congress Cataloguing-in-Publication Data
Lyuu, Yuh-Dauh.
Information dispersal and parallel computation / Yuh-Dauh Lyuu.
p. cm. – (Cambridge international series on parallel
computation : 3)
Includes bibliographical references (p.) and index.
ISBN 0 521 43226 X hardback
1. Parallel processing (Electronic computers) 2. Computer
algorithms. 3. Fault-tolerant computing. I. Title. II. Series.
QA76.58.L98 1992
004′.35–dc20 92-29635
CIP

ISBN 0 521 43226 X hardback
ISBN 0 521 60279 3 paperback

To the memory of CHRISTINE LYUU

Contents

List of Figures **xi**

Preface **xiii**

Acknowledgments **xv**

Glossary of Notations **xvii**

1 Introduction **1**
- 1.1 The von Neumann Machine Paradigm 2
- 1.2 Issues in Parallel Processing 3
- 1.3 Overview of the Book 6

2 Information Dispersal **8**
- 2.1 Introduction 9
- 2.2 Results from Algebra and Number Theory 11
- 2.3 The Vandermonde Matrix 12
  - 2.3.1 Relation with polynomial interpolation 12
  - 2.3.2 Methods for interpolating polynomials 13
  - 2.3.3 The discrete Fourier transform 13
- 2.4 An IDA Schema 14
- 2.5 Fast Information Dispersal Algorithms 16
  - 2.5.1 Choosing a commutative ring 18
  - 2.5.2 Comparison with Rabin's IDA 19
  - 2.5.3 Variations on a theme 20
  - 2.5.4 Related work 22
- 2.6 Application to Voting 22

3 Interconnection Networks **26**
3.1 Introduction 27
3.2 Basic Terminology for Networks 27
3.2.1 Interconnection and node structures 27
3.2.2 Switching 30
3.2.3 Universality 31
3.3 Graph-Theoretical Modeling 32
3.4 Sample Networks 32
3.4.1 The hypercube network 32
3.4.2 The de Bruijn network 33
3.4.3 The butterfly network 33
3.4.4 Other networks 34
3.5 Some Desirable Properties for Networks 36

4 Introduction to Parallel Routing **41**
4.1 Introduction 42
4.2 The Parallel Routing Problem 42
4.3 Outline of Approach 44
4.4 Summary of Results 45
4.5 Other Work on IDA-Based Routing Schemes 47
4.6 Previous Work on Parallel Routing 47

5 Fault-Tolerant Routing Schemes and Analysis **50**
5.1 Parallel Communication Scheme 51
5.2 Basic Terminology and Conventions 52
5.3 Routing on the Fault-Free Hypercube 55
5.3.1 Definitions 55
5.3.2 SRA: description and preliminary analysis 56
5.3.3 Analysis of SRA 58
5.3.4 Remarks on the tightness of bounds 61
5.4 Routing on the Hypercube with Faults 62
5.4.1 Definitions 62
5.4.2 Description of FSRA 63
5.4.3 Analysis of FSRA under the fault-free model 64
5.4.4 Analysis of FSRA under random fault models 67
5.5 Routing on the de Bruijn Network 71
5.5.1 Definitions 71
5.5.2 RABN: description and preliminary analysis 72
5.5.3 Analysis of RABN under the fault-free model 74

5.5.4 Analysis of RABN under a random fault model 78
5.6 On Time Bounds and Models 81
5.7 A General Framework for IDA-Based Schemes 82
5.8 FSRA under a More Realistic Fault Model 83

6 Simulation of the PRAM **86**
6.1 Introduction 87
6.2 The PRAM model 88
6.2.1 Simulation of the PRAM 89
6.2.2 Previous work on simulation of PRAMs 90
6.2.3 Combining in the priority model 91
6.2.4 A digression on "hot spot" contention 92
6.3 Special Case Simulation 94
6.3.1 The simulation algorithm and results 95
6.3.2 Analysis of size of regular buffers 96
6.3.3 Analysis of size of wait buffers 100
6.3.4 Analysis of slowdown 106
Appendix: A Universal Simulation Scheme 107

7 Asynchronism and Sensitivity **110**
7.1 Introduction 111
7.2 Terminology and Definitions 112
7.3 Asynchronous Routing Using Local Synchronization 112
7.4 Run-Time Sensitivity to Component Speeds 116
7.4.1 A general theorem 117
7.4.2 Sensitivity analysis of FSRA 119
7.5 Communication Complexity and Sensitivity 121

8 On-Line Maintenance **123**
8.1 Introduction 124
8.2 On-Line Wire Maintenance on Hypercubes with FSRA 124
8.2.1 Definitions and outline of approach 125
8.2.2 Partition $\mathcal{P}$ 126
8.2.3 Analysis of partition $\mathcal{P}$ 127

9 A Fault-Tolerant Parallel Computer **130**
9.1 Introduction 131
9.2 Introductory Fault-Tolerant Computing 131
9.2.1 Techniques 131

9.2.2 Samples 133
9.3 HPC: Approach and Building Blocks 134
9.3.1 Cluster: fault containment and communication 135
9.3.2 HPC-Routing: analysis of run-time 139
9.4 Conclusions 139

Bibliography **141**

Index **171**

Cambridge University Press
0521602793 - Information Dispersal and Parallel Computation
Yuh-Dauh Lyuu
Frontmatter
More information

List of Figures

2.1 The Working of IDA. 9
2.2 Idea of Voting. 23
2.3 An IDA-Based Voting Scheme FILTERING. 23

3.1 Two Ways to Organize Processors and Memory Modules. 28
3.2 Direct (Communication) Network Architecture. 29
3.3 Sample Networks. 38
3.4 Sample Networks (continued). 39
3.5 Sample Networks (continued). 40

4.1 Routing in a Direct Network. 43

5.1 SRA ("Subcube Routing Algorithm"). 57
5.2 The Working of SRA on C_4. 59
5.3 A Routing Scheme without the Subcube Concept. 61
5.4 FSRA ("Fault-Tolerant Subcube Routing Algorithm"). 63
5.5 The Working of FSRA on C_6. 65
5.6 RABN ("Routing Algorithm for the de Bruijn Network"). 73
5.7 The Working of RABN on $N_{3,3}$. 75
5.8 Rabin's Paradigm. 83
5.9 Unsuccessful Routing Under $O(1)$ Link Failure Probability. 85

6.1 The PRAM Model of Computation. 88
6.2 Tree Saturation. 93
6.3 How Pieces Got Combined in Subphase 2.1. 98
6.4 Combining and Wait Buffers in Subphase 2.1. 102

7.1 How SA Works. 114
7.2 Properties of the l-Graph. 118
7.3 Sensitivity of SRA on C_2. 120

9.1 The Clusters. 136
9.2 How HPC-Routing Works. 137
9.3 The HPC-Routing Algorithm. 138

Preface

It has long been recognized that computer design utilizing more than one processor is one promising approach — some say the only approach — toward more powerful computing machines. Once one adopts this view, several issues immediately emerge: how to connect processors and memories, how do processors communicate efficiently, how to tolerate faults, how to exploit the redundancy inherent in multiprocessors to perform on-line maintenance and repair, and so forth.

This book confronts the above-mentioned issues with two keys insights. There exist error-correcting codes that generate redundancy which is efficient in terms of the number of bits. Such redundancy is used to correct errors and erasures caused by component failures and resource limitations (such as limited buffer size). This insight comes from Michael Rabin. The next insight, due to Leslie Valiant, demonstrates the criticality of randomization in achieving communication efficiency.

We intend to make this book an up-to-date account of the information dispersal approach as it is applied to parallel computation. We also discuss related work in the general area of parallel communication and computation and provide an extensive bibliography in the hope that either might be helpful for researchers and students who want to explore any particular topic. Although materials in this book extend across several disciplines (algebra, coding theory, number theory, arithmetics, algorithms, graph theory, combinatorics, and probability), it is, the author believes, a self-contained book; adequate introduction is given and every proof is complete.

Technical Summary

Efficient schemes are presented for the following problems: fast parallel communication, low congestion, fault tolerance, simulation of ideal parallel computation models, synchronization in asynchronous networks with low sensitivity to variations in component speed, and on-line maintenance. All of the schemes either employ, or have inspirations from, Rabin's information dispersal idea. We also describe efficient information dispersal algorithms (IDAs) and their application to the enforcement of regions of fault containment.

Let N denote the size of the hypercube network. We present a randomized communication scheme, FSRA ("Fault-Tolerant Subcube Routing Algorithm"), that routes in $2 \log N + 1$ time using only constant size buffers and with probability of success $1 - N^{-\Theta(\log N)}$. FSRA furthermore tolerates $O(N)$ random link failures with high probability. Similar results are also proved for the de Bruijn network. A general framework for fault-tolerant routing is described for the large class of node-symmetric networks (such as the hypercube).

FSRA is employed to simulate, without using hashing, a class of CRCW PRAM ("Concurrent-Read Concurrent-Write Parallel Random Access Machine") programs with a slowdown of $O(\log N)$ with almost certainty if combining is used. Fault-tolerant simulation schemes for general CRCW programs are also presented.

A simple acknowledgment synchronizer can make the routing schemes in this book run on asynchronous networks without loss of efficiency. We further show that speed of any component — be it a processor or a link — has only linear impact on the run-time of FSRA; that is, the extra delay in its run-time is only proportional to the drift in the component's delay and is independent of the size of the network.

On-line maintainability makes the machine more available to the user. We show that, under FSRA, a constant fraction of the links can be disabled with essentially no impact on the routing performance. This result immediately suggests several efficient maintenance and repair procedures.

Based on the above results, a fault-tolerant parallel computing system, called HPC ("hypercube parallel computer"), is sketched at the end of this book.

Acknowledgments

The study of information dispersal and its various applications was initiated by my thesis advisor at Harvard University, Prof. Michael Rabin, under whose supervision and encouragement I completed my thesis *An Information Dispersal Approach to Issues in Parallel Processing* and several other papers [233, 234, 235, 236, 237]. The architecture of that thesis is largely preserved in this book. His ability to approach seemingly difficult problems with clear but at first not so obvious insights compels me to strive for simplicity and precision. He also raised the sensitivity issue for asynchronous algorithms.

I thank Prof. Meichun Hsu and Prof. Les Valiant for serving on my thesis committee. Prof. Valiant's seminal work on parallel routing also strongly influences the direction of my research.

This book benefits from several people's comments, knowledge, and generosity. Prof. Rabin contributed to the proof of Theorem 7.1. Satish Rao first suggested the possibility of an information dispersal algorithm (IDA) based on the Fourier transform. After such an IDA had been developed (see Chapter 2), Prof. Krizanc of Rochester University pointed out that the scheme had largely been anticipated by Preparata [280]. Prof. Beaver of Pennsylvania State University also pointed out the connection between my scheme and Shamir's secret-sharing algorithm and subsequent idea advanced by Ben-Or, Goldwasser, and Wigderson [46] in that setting. (It turns out that all of them are Reed-Solomon codes.) Joe Kilian and Prof. Rabin improved the phrasing of Theorem 8.3. Prof. Tsantilas of Columbia University has always been a generous source on the literature of, and ideas about, routing. With the help of Bill Gear of NEC Research Institute, we showed that a global minimum is indeed produced by Eq. (6.8), correcting an error in the original proof of Lemma 6.15. Harry Bochner and Joanne Klys of Aiken Computation Laboratory, Harvard University, provided valuable assistance on the use of graphical tools and subtle features of LaTeX. Part of Chapter 3 was written at the Department of Computer Science and Information Engineering, National Taiwan University, under Kuo-Liang Chung's computer account. Jehoshua Bruck of IBM, Yonatan Aumann of the Hebrew University of Jerusalem, and Peter Mysliwietz of the University of Paderborn suggested relevant literature. Prof. McColl of Oxford University not only suggested the title, but also made the publication possible.

I am indebted to NEC Research Institute for the unmatched generosity and the vision that autonomy is essential to scientific inquiry. The research reported in my original dissertation was generously supported by National Science Foundation Grant MCS-8121431 at Harvard University.

It has been a pleasure to work with Dr. Alan Harvey, Editor of Mathematical Sciences, and Lauren Cowles, Editor of Mathematics and Computer Science, both at Cambridge University Press. My thanks are due to them and their staff.

The constant supports from my wife, Chih-Lan, made the writing experience a pleasant one. Although my son, Raymond, has increased in age since his birth when I was writing my thesis *with* him, I again had to write this book with him. Fortunately, now, as before, the computer has made the destruction of manuscripts much less life-threatening.

Glossary of Notations

notation	page number	notation	page number
$\approx$	13	D	96
$\lvert A \rvert$	11	D_1	97
A_{ij}	11	D_2	99
$A_t(y)$	97	$dim(i)$	126
A^T	11	$E(G)$	52
A^*	11	$E(i,l,t)$	126
A^{-1}	11	F	14
$\mathcal{A}$	14	$\lvert F \rvert$	9
$\tilde{\mathcal{A}}$	15	$F_{n,n}$	13
$\mathcal{A}_{\mathrm{FFT}}$	16	$G_1 \subseteq G_2$	52
$\mathcal{A}_{\mathrm{V}}$	16	$G_1 \cap G_2 = \emptyset$	52
a_t	97	$GF(p)$	11
$\overline{a}$	34	$G(V,E)$	32
$[\, a_{ij} \,]_{\substack{1 \le i \le n \\ 1 \le j \le m}}$	11	I	11
B	14	$\mathcal{I}_x$	135
$BF(x)$	52	$IDA(n,m)$	14
$B(m,N,q)$	54	L	14 and 53
$B_t(y)$	97	$L(i,j)$	117
$B_t^i(y)$	100	$L_{i,l}$	126
$bits_k(i)$	56	M	90
$bits_3(i)$	126	M_i	14
$\beta_x(l)$	96	m_0	53
C_n	55	$N_{d,n}$	71
$C_{n-k}(y)$	56	$N_{d,n}\,(i:l)$	72
$C_{n-k-1}(i:l)$	62	$nodes(l)$	126
$\mathcal{C}(x)$	135	n_0	53
γ_t	103	n_x	96

notation	page number
$P_{\vec{a}}(x)$	12
$\mathcal{P}$	127
$P(i)$	73
P_x	53
$P_x(i)$	53
$P_x(i) \in V$	53
p_0	60
π	42
$R(+, \cdot, 0, 1)$	11
$T(i)$	117
Δt	116
(u, v)	32
$V(G)$	52
$v_x(i)$	96
$x[\,i:j\,]$	55
$x(i:l)$	63
$x^{(i)}$	63 and 135
$x//i$	55
$x \circ y$	56
$X^*(t)$	103
$\mathbf{Z}_q$	11
$Z_{t,x,l}$	101
ω	13