

PARALLEL AND DISTRIBUTED SIMULATION SYSTEMS

WILEY SERIES ON PARALLEL AND DISTRIBUTED COMPUTING

EDITOR:

ALBERT Y. ZOMAYA

FUJIMOTO • Parallel and Distributed Simulation Systems

SAPATY • Mobile Processing in Distributed and Open Environments

XAVIER AND IYENGAR • Introduction to Parallel Algorithms

PARALLEL AND DISTRIBUTED SIMULATION SYSTEMS

Richard M. Fujimoto, PhD
Georgia Institute of Technology



A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS, INC.

New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

This book is printed on acid-free paper. ∞

Copyright © 2000 by John Wiley & Sons, Inc. All rights reserved.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ@WILEY.COM.

For ordering and customer service, call 1-800-CALL-WILEY.

Library of Congress Cataloging-in-Publication Data:

Fujimoto, Richard M.

Parallel and distributed simulation systems / Richard M.

Fujimoto.

p. cm.

"A Wiley-Interscience publication."

Includes bibliographical references.

ISBN 0-471-18383-0 (alk. paper)

1. Computer simulation. 2. Parallel processing (Electronic computers) 3. Electronic data processing—Distributed processing.

I. Title.

QA76.9.C65F84 2000

003'.3435—dc21

99-25438

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To
Jan, Emily, and Alex

CONTENTS

Preface	xv
PART I INTRODUCTION	
1 Background and Applications	3
1.1 Why Parallel/Distributed Simulation?	4
1.2 Analytic Simulations versus Virtual Environments	6
1.3 Historical Perspective	8
1.3.1 High-Performance Computing Community	8
1.3.2 Defense Community	9
1.3.3 Interactive Gaming and Internet Communities	10
1.4 Applications	11
1.4.1 Military Applications	12
1.4.2 Entertainment	13
1.4.3 Social Interactions and Business Collaborations	13
1.4.4 Education and Training	14
1.4.5 Telecommunication Networks	14
1.4.6 Digital Logic Circuits and Computer Systems	15
1.4.7 Transportation	16
1.5 Underlying Technologies	16
1.6 Hardware Platforms	17
1.6.1 Parallel versus Distributed Computers	17
1.6.2 Shared-Memory Multiprocessors	19
1.6.3 Distributed-Memory Multicomputers	20
1.6.4 SIMD Machines	21
1.6.5 Distributed Computers	22
1.7 Summary	23
1.8 Additional Readings	24
2 Discrete Event Simulation Fundamentals	27
2.1 Time	27
2.2 Real-Time, Scaled Real-Time, and As-Fast-As-Possible Execution	28
	vii

2.3	State Changes and Time Flow Mechanisms	30
2.3.1	Time-Stepped Execution	31
2.3.2	Event-Driven Execution	32
2.4	Discrete-Event Simulation Programs	34
2.5	An Example Application	36
2.6	Starting and Stopping the Simulation	39
2.7	Parallel/Distributed Simulation Example	39
2.8	World Views and Object-Oriented Simulation	41
2.8.1	Simulation Processes	41
2.8.2	Object-Based and Object-Oriented Simulations	45
2.8.3	Query Events and Push versus Pull Processing	46
2.8.4	Event Retraction	47
2.9	Other Approaches to Exploiting Concurrent Execution	48
2.10	Additional Readings	48

PART II PARALLEL AND DISTRIBUTED DISCRETE-EVENT SIMULATION

3	Conservative Synchronization Algorithms	51
3.1	Synchronization Problem	52
3.2	Deadlock Avoidance Using Null Messages	54
3.3	Lookahead and the Simulation Model	58
3.4	Deadlock Detection and Recovery	60
3.4.1	Deadlock Detection	60
3.4.2	Deadlock Recovery	63
3.5	Synchronous Execution	65
3.5.1	Centralized Barriers	66
3.5.2	Tree Barrier	67
3.5.3	Butterfly Barrier	68
3.5.4	Transient Messages	70
3.5.5	A Simple Synchronous Protocol	74
3.5.6	Distance between Logical Processes	75
3.6	Bounded Lag	79
3.7	Conditional versus Unconditional Information	81
3.8	Dynamic Processes and Interconnections	82

3.9	Repeatability and Simultaneous Events	84
3.9.1	Using Hidden Time Stamp Fields to Order Simultaneous Events	85
3.9.2	Priority Numbers	85
3.9.3	Receiver-Specified Ordering	86
3.10	Performance of Conservative Mechanisms	87
3.11	Summary and Critique of Conservative Mechanisms	91
3.12	Additional Readings	94
4	Time Warp	97
4.1	Preliminaries	98
4.2	Local Control Mechanism	98
4.2.1	Rolling Back State Variables	100
4.2.2	Unsending Messages	102
4.2.3	Zero Lookahead, Simultaneous Events, and Repeatability	106
4.3	Global Control Mechanism	108
4.3.1	Fossil Collection	110
4.3.2	Error Handling	110
4.4	Computing Global Virtual Time	112
4.4.1	Transient Message Problem	113
4.4.2	Simultaneous Reporting Problem	114
4.4.3	Samadi's GVT Algorithm	116
4.4.4	Mattern's GVT Algorithm	117
4.5	Other Mechanisms	122
4.5.1	Dynamic Memory Allocation	122
4.5.2	Infrequent State Saving	123
4.5.3	Specifying What to Checkpoint	126
4.5.4	Event Retraction	128
4.5.5	Lazy Cancellation	129
4.5.6	Lazy Re-evaluation	132
4.6	Scheduling Logical Processes	133
4.7	Summary	135
4.8	Additional Readings	135
5	Advanced Optimistic Techniques	137
5.1	Memory Utilization in Time Warp	138

5.1.1	Preliminaries: State Vectors and Message Send Time Stamps	139
5.1.2	Memory Management Mechanisms and Message Sendback	140
5.1.3	Storage Optimality	142
5.1.4	Cancelback	145
5.1.5	Artificial Rollback	145
5.1.6	Pruneback	147
5.1.7	Memory-Based Flow Control	148
5.1.8	Trading Off Performance and Memory	149
5.2	Performance Hazards in Time Warp	151
5.2.1	Chasing Down Incorrect Computations	151
5.2.2	Rollback Echoes	152
5.3	Other Optimistic Synchronization Algorithms	154
5.3.1	Moving Time Window	155
5.3.2	Lookahead-Based Blocking	156
5.3.3	Local Rollback	156
5.3.4	Breathing Time Buckets	157
5.3.5	Wolf Calls	159
5.3.6	Probabilistic Rollbacks	160
5.3.7	Space-Time Simulation	160
5.3.8	Summary	160
5.4	Putting It All Together: Georgia Tech Time Warp (GTW)	161
5.4.1	Programmer's Interface	161
5.4.2	I/O and Dynamic Memory Allocation	162
5.4.3	GTW Data Structures	163
5.4.4	Direct Cancellation	165
5.4.5	Event-Processing Loop	166
5.4.6	Buffer Management	166
5.4.7	Flow Control	168
5.4.8	GVT Computation and Fossil Collection	168
5.4.9	Incremental State Saving	169
5.4.10	Local Message Sends	169
5.4.11	Message Copying	169
5.4.12	Batch Event Processing	169
5.4.13	Performance Measurements	170
5.5	Summary	171
5.6	Comparing Optimistic and Conservative Synchronization	172
5.7	Additional Readings	174

6	Time Parallel Simulation	177
6.1	Time Parallel Cache Simulation Using Fix-up Computations	179
6.2	Simulation of an ATM Multiplexer Using Regeneration Points	183
6.3	Simulation of Queues Using Parallel Prefix	188
6.4	Summary	190
6.5	Additional Readings	191

PART III DISTRIBUTED VIRTUAL ENVIRONMENTS (DVEs)

7	DVEs: Introduction	195
7.1	Goals	195
7.2	Contrasting DVE and PDES Systems	196
7.3	Server versus Serverless Architectures	197
7.4	Distributed Interactive Simulation	199
7.4.1	DIS Design Principles	200
7.4.2	DIS PDUs	202
7.4.3	Time Constraints	203
7.5	Dead Reckoning	204
7.5.1	Dead Reckoning Models	206
7.5.2	Time Compensation	208
7.5.3	Smoothing	208
7.6	High Level Architecture	209
7.6.1	Historical Perspective	210
7.6.2	Overview of the HLA	211
7.6.3	HLA Rules	213
7.6.4	Object Models and the Object Model Template	213
7.6.5	Interface Specification	218
7.6.6	Typical Federation Execution	219
7.7	Summary	220
7.8	Additional Readings	221
8	Networking and Data Distribution	223
8.1	Message-Passing Services	223
8.1.1	Reliable Delivery	223

8.1.2	Message Ordering	224
8.1.3	Connection-Oriented versus Connectionless Communication	224
8.1.4	Unicast versus Group Communication	224
8.1.5	Examples: DIS and NPSNet	225
8.2	Networking Requirements	226
8.3	Networking Technologies	227
8.3.1	LAN Technologies	227
8.3.2	WAN Technologies	231
8.3.3	Quality of Service	233
8.4	Communication Protocols	234
8.4.1	OSI Protocol Stack	235
8.4.2	ATM Protocol Stack	238
8.4.3	Internetworking and Internet Protocols	240
8.5	Group Communication	243
8.5.1	Groups and Group Communication Primitives	244
8.5.2	Transport Mechanisms	244
8.6	Data Distribution	245
8.6.1	Interface to the Data Distribution System	245
8.6.2	Example: Data Distribution in the High Level Architecture	247
8.6.3	Implementation Issues	251
8.6.4	Dynamic Group Management	254
8.6.5	Wide-Area Viewers and Fast-Moving Entities	255
8.7	Summary	256
8.8	Additional Readings	257
9	Time Management and Event Ordering	259
9.1	The Problem	259
9.2	Message-Ordering Services	261
9.2.1	Causal Order	261
9.2.2	Causal and Totally Ordered	265
9.2.3	Delta Causality	267
9.2.4	Time Stamp Order	268
9.3	Synchronizing Wallclock Time	269
9.3.1	Time and Time Sources	270
9.3.2	Clock Synchronization Algorithms	271
9.3.3	Correcting Clock Synchronization Errors	273
9.3.4	Network Time Protocol	274

9.4	Summary	275
9.5	Additional Readings	275
References		277
Index		293

PREFACE

These are exciting times in the parallel and distributed simulation field. After many years of research and development in university and industrial laboratories, the field has exploded in the last decade and is now seeing use in many real-world systems and applications. My goal in writing *Parallel and Distributed Simulation Systems* is to give an in-depth treatment of technical issues concerning the execution of discrete event simulation programs on computing platforms composed of many processors interconnected through a network. The platform may range from tightly coupled multiprocessor computer systems confined to a single cabinet or room to geographically distributed personal computers or specialized simulators (for example, video game systems) spread across the world. This technology can be used to speed up the execution of large-scale simulations, for example simulations of the next generation of the Internet, or to create distributed synthetic environments for training or entertainment.

My goal in writing this book was to bring together into one volume the fundamental principles concerning parallel and distributed simulation systems that today are scattered across numerous journals and conference proceedings. The intended audience includes managers and practitioners involved in research and/or development of distributed simulation systems. The book can serve as a textbook for an advanced undergraduate or a graduate level computer science course. The book might be of interest in other disciplines (for example, industrial engineering or operations research) although the principal emphasis is on issues concerning parallel and distributed computation. Prior knowledge of discrete event simulation parallel, or distributed computation would be helpful, but is not essential as the book will include brief introductions to these fields.

Contents

The book is divided into three parts. The first provides an introduction to the field. Chapter 1 describes typical applications where this technology can be applied, and gives an historical perspective to characterize the communities that developed and refined this technology. Background information concerning parallel and distributed computing systems is reviewed. Chapter 2 reviews fundamental concepts in discrete event simulation to provide a common basis and terminology that is used in the remainder of the book.

The second part is primarily concerned with parallel and distributed execution of simulations, primarily for analysis applications such as to design large, complex systems. Here the goal is to use multiple processors to speed up the execution. Much of the material in these four chapters is concerned with synchronization algorithms that are used to ensure a parallel execution of the simulation yields the same results as a sequential execution, but (hopefully!) much more quickly. Two principal approaches to addressing this issue are called conservative and optimistic synchronization. Chapter 3 is concerned with the former, and Chapters 4 and 5 with the latter. Chapter 6 is concerned with an altogether different approach to parallel execution called time parallel simulation that is only suitable for certain classes of simulation problems, but can yield dramatic performance improvements when it can be applied.

The third part is concerned with distributed virtual environments (DVEs). Here the emphasis is on real-time simulations, that is, to create virtual environments into which humans may be embedded, for example, for training or entertainment. Chapter 7 gives an introduction to this area, focusing primarily on two efforts within the defense community, namely Distributed Interactive Simulation (DIS) and the High Level Architecture (HLA) where much of this technology was developed and has been applied. Chapters 8 and 9 are concerned with two specific issues in DVEs. Chapter 8 covers the problem of efficiently distributing data among the participants of the DVE. The first half of the chapter is an introduction to computer networks which provide the underlying communication support for DVEs. The second half is concerned with techniques to effectively utilize the networking infrastructure, particularly for large-scale simulations with many interacting components. Finally, Chapter 9 revisits the problem of time synchronization in DVEs as well as the problem of ensuring that the different computers participating in the simulation have properly synchronized clocks.

Part I lays the groundwork for the remainder of the book, so should be read first. Parts II and III can be read in either order. I have used this book as the text in a 10-week course in parallel and distributed simulation taught at Georgia Tech, and plan to use it when we transition to 15-week semesters. Alternatively, this book could be used for part of a course in discrete event simulation. When used in this manner, instructors may wish to skip Chapters 5 and 6, and the first half of Chapter 8 to obtain a more abbreviated treatment of the subject material.

Software

Interested readers may wish to try out some of the algorithms discussed in this book. Although software is not included with the text, it is available. In particular, the Georgia Tech Time Warp (GTW) software discussed in Chapter 5 and an implementation of a subset of the High level Architecture Run Time Infrastructure are freely available for education and research purposes. Information concerning this software is available at <http://www.cc.gatech.edu/computing/pads>. To obtain a copy of either or both of these software packages, you may contact me via electronic mail at fujimoto@cc.gatech.edu.

Acknowledgments

Obviously, this book would not be possible without the many technical contributions by numerous individuals in both academia and industry. I have attempted to recognize as many of these contributors as possible in the bibliography and references to additional reading materials. Regrettably, the field has expanded to the extent that anything approaching a complete listing of the contributors is impossible.

I am in debt to many individuals who contributed directly to the development of this book. In particular, many useful comments on early drafts were provided by students in my graduate class on parallel and distributed simulation taught at Georgia Tech. Specific detailed comments from Glenn Oberhauser and Katherine Morse are also appreciated. I am grateful to several funding agencies that sponsored my research in parallel and distributed simulation; some of the results of this work are included in this text. These agencies include the Ballistic Missile Defense Organization, the Defense Advance Research Projects Agency, the Defense Modeling and Simulation Office, the National Science Foundation, SAIC, Mitre Corporation, Bellcore, Army Research Office, the Office of Naval Research, and the Strategic Missile Defense Command.

Finally, I owe the greatest gratitude to my family, who provided continued support and understanding despite the countless evenings and weekends of dealing with my absence due to this project which came to be known simply as “the book.” This manuscript would never have been completed were it not for their love and devotion.

PARALLEL AND DISTRIBUTED SIMULATION SYSTEMS