

TASK SCHEDULING FOR PARALLEL SYSTEMS



THE WILEY BICENTENNIAL—KNOWLEDGE FOR GENERATIONS

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

WILLIAM J. PESCE
PRESIDENT AND CHIEF EXECUTIVE OFFICER

PETER BOOTH WILEY
CHAIRMAN OF THE BOARD

TASK SCHEDULING FOR PARALLEL SYSTEMS

Oliver Sinnen

Department of Electrical and Computer Engineering
The University of Auckland
New Zealand



WILEY-INTERSCIENCE
A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2007 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

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, Inc., 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400, fax 978-646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Wiley Bicentennial Logo: Richard J. Pacifico

Library of Congress Cataloging-in-Publication Data:

Sinnen, Oliver, 1971-

Task scheduling for parallel systems / by Oliver Sinnen.

p. cm.

Includes bibliographical references and index.

978-0-471-73576-2

1. Parallel processing (Electronic computers)
2. Multitasking (Computer science)
3. Computer scheduling. I. Title.

QA76.58.S572 2007

044'.35—dc22

2006052157

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

à Patrícia

CONTENTS

Preface	xi
Acknowledgments	xiii
1. Introduction	1
1.1 Overview	1
1.2 Organization	5
2. Parallel Systems and Programming	7
2.1 Parallel Architectures	7
2.1.1 Flynn's Taxonomy	7
2.1.2 Memory Architectures	9
2.1.3 Programming Paradigms and Models	11
2.2 Communication Networks	13
2.2.1 Static Networks	13
2.2.2 Dynamic Networks	18
2.3 Parallelization	22
2.4 Subtask Decomposition	24
2.4.1 Concurrency and Granularity	24
2.4.2 Decomposition Techniques	25
2.4.3 Computation Type and Program Formulation	27
2.4.4 Parallelization Techniques	28
2.4.5 Target Parallel System	28
2.5 Dependence Analysis	29
2.5.1 Data Dependence	29
2.5.2 Data Dependence in Loops	32
2.5.3 Control Dependence	35
2.6 Concluding Remarks	36
2.7 Exercises	37
3. Graph Representations	40
3.1 Basic Graph Concepts	40

3.1.1	Computer Representation of Graphs	43
3.1.2	Elementary Graph Algorithms	46
3.2	Graph as a Program Model	49
3.2.1	Computation and Communication Costs	50
3.2.2	Comparison Criteria	50
3.3	Dependence Graph (DG)	51
3.3.1	Iteration Dependence Graph	53
3.3.2	Summary	55
3.4	Flow Graph (FG)	56
3.4.1	Data-Driven Execution Model	60
3.4.2	Summary	61
3.5	Task Graph (DAG)	62
3.5.1	Graph Transformations and Conversions	64
3.5.2	Motivations and Limitations	68
3.5.3	Summary	69
3.6	Concluding Remarks	69
3.7	Exercises	70
4.	Task Scheduling	74
4.1	Fundamentals	74
4.2	With Communication Costs	76
4.2.1	Schedule Example	81
4.2.2	Scheduling Complexity	82
4.3	Without Communication Costs	86
4.3.1	Schedule Example	87
4.3.2	Scheduling Complexity	88
4.4	Task Graph Properties	92
4.4.1	Critical Path	93
4.4.2	Node Levels	95
4.4.3	Granularity	101
4.5	Concluding Remarks	105
4.6	Exercises	105
5.	Fundamental Heuristics	108
5.1	List Scheduling	108
5.1.1	Start Time Minimization	111
5.1.2	With Dynamic Priorities	114
5.1.3	Node Priorities	115
5.2	Scheduling with Given Processor Allocation	118
5.2.1	Phase Two	119

5.3	Clustering	119
5.3.1	Clustering Algorithms	121
5.3.2	Linear Clustering	124
5.3.3	Single Edge Clustering	128
5.3.4	List Scheduling as Clustering	135
5.3.5	Other Algorithms	138
5.4	From Clustering to Scheduling	139
5.4.1	Assigning Clusters to Processors	139
5.4.2	Scheduling on Processors	141
5.5	Concluding Remarks	141
5.6	Exercises	142
6.	Advanced Task Scheduling	145
6.1	Insertion Technique	145
6.1.1	List Scheduling with Node Insertion	148
6.2	Node Duplication	150
6.2.1	Node Duplication Heuristics	153
6.3	Heterogeneous Processors	154
6.3.1	Scheduling	157
6.4	Complexity Results	158
6.4.1	$\alpha \beta \gamma$ Classification	158
6.4.2	Without Communication Costs	165
6.4.3	With Communication Costs	165
6.4.4	With Node Duplication	168
6.4.5	Heterogeneous Processors	170
6.5	Genetic Algorithms	170
6.5.1	Basics	171
6.5.2	Chromosomes	172
6.5.3	Reproduction	177
6.5.4	Selection, Complexity, and Flexibility	180
6.6	Concluding Remarks	182
6.7	Exercises	183
7.	Communication Contention in Scheduling	187
7.1	Contention Awareness	188
7.1.1	End-Point Contention	189
7.1.2	Network Contention	190
7.1.3	Integrating End-Point and Network Contention	192
7.2	Network Model	192
7.2.1	Topology Graph	192
7.2.2	Routing	198
7.2.3	Scheduling Network Model	202

7.3	Edge Scheduling	203
7.3.1	Scheduling Edge on Route	204
7.3.2	The Edge Scheduling	208
7.4	Contention Aware Scheduling	209
7.4.1	Basics	209
7.4.2	NP-Completeness	211
7.5	Heuristics	216
7.5.1	List Scheduling	216
7.5.2	Priority Schemes—Task Graph Properties	219
7.5.3	Clustering	220
7.5.4	Experimental Results	221
7.6	Concluding Remarks	223
7.7	Exercises	224
8.	Processor Involvement in Communication	228
8.1	Processor Involvement—Types and Characteristics	229
8.1.1	Involvement Types	229
8.1.2	Involvement Characteristics	232
8.1.3	Relation to LogP and Its Variants	236
8.2	Involvement Scheduling	238
8.2.1	Scheduling Edges on the Processors	240
8.2.2	Node and Edge Scheduling	246
8.2.3	Task Graph	247
8.2.4	NP-Completeness	248
8.3	Algorithmic Approaches	250
8.3.1	Direct Scheduling	251
8.3.2	Scheduling with Given Processor Allocation	254
8.4	Heuristics	257
8.4.1	List Scheduling	257
8.4.2	Two-Phase Heuristics	261
8.4.3	Experimental Results	263
8.5	Concluding Remarks	264
8.6	Exercises	265
	Bibliography	269
	Author Index	281
	Subject Index	285

PREFACE

Even though the area of parallel computing has existed for many decades, programming a parallel system is still a challenging problem, much more challenging than programming a single processor system. With the current dual-core and multicore processors from IBM, AMD, Intel, and others, mainstream PCs have entered the realm of parallel systems. The investigation and understanding of the foundations of parallel computing is therefore more important than ever.

One of these foundations is task scheduling. To execute a program consisting of several tasks on a parallel system, the tasks must be arranged in space and time on the multiple processors. In other words, the tasks must be mapped to the processors and ordered for execution. This so-called task scheduling is a very complex problem and crucially determines the efficiency of the parallel system. In fact, task scheduling is an NP-hard problem; that is, an optimal solution generally cannot be found in polynomial time (unless $P = NP$). This has been motivating the development of many heuristics for its near optimal solution.

This book is devoted to task scheduling for parallel systems. Anyone who gets involved for the first time in task scheduling is overwhelmed by the enormous wealth of heuristics, models, and methods that have been contributed during the last decades. One of my main objectives for this book is to bring order into this jungle of task scheduling. However, the book does not simply categorize and order scheduling heuristics. Instead, it investigates and presents task scheduling by extracting and discussing common models, methods, and techniques, and by setting them into relation. Hence, this book is not a mere survey of scheduling algorithms, but rather an attempt at a consistent and unifying theoretical framework.

Another objective I have with this book is to go beyond the classic approach to task scheduling by studying scheduling under more advanced and accurate system models. These system models consider heterogeneity, contention for communication resources, and involvement of the processor in communication. For efficient and accurate task scheduling, a realistic system model is most crucial. This book is the first publication that discusses advanced system models for task scheduling in a comprehensive form.

Task Scheduling for Parallel Systems is targeted at practicing professionals, researchers, and students. For those who are new to task scheduling, the first chapters carefully introduce parallel systems and their programming, setting task scheduling into the context of the program parallelization process. Practitioners involved in parallel programming will gain an understanding of fundamental aspects of the parallelization process. This knowledge will help them to write more efficient code.

Compiler and parallelization tool developers will benefit from a deeper understanding of the scheduling problem, which is also a generalization of many other problems they face (e.g., loop scheduling). A chapter on graph models promotes the understanding of these relations. For task scheduling researchers, this book serves as a comprehensive reference, based on a unifying framework. The research community will especially value the later chapters on advanced scheduling and sophisticated scheduling models. Graduate students of parallel computing and compiler courses can use this book to thoroughly study task scheduling, which is supported by the exercises at the end of each chapter. The extensive index and the large number of bibliographic references make this book a valuable tool for everybody interested in task scheduling.

For a brief introduction to task scheduling and an overview of this book, including a short summary of each chapter, refer to Chapter 1.

OLIVER SINNEN
The University of Auckland
New Zealand

ACKNOWLEDGMENTS

This book has its roots in my PhD thesis, which I presented at the Instituto Superior Técnico, Technical University of Lisbon in Portugal. I therefore want to sincerely thank Prof. Leonel Sousa again for his academic advice, the fruitful discussions, and his criticism during that time. In this context I also thank Prof. Frode Sandnes, Oslo University College, Norway, who introduced me to genetic algorithms and has been an invaluable research colleague for many years.

I would like to thank The University of Auckland for the opportunity to write this book while being a lecturer in the Department of Electrical and Computer Engineering.

Cover design: Digital image of a painting created by Patrícia Raquel de Vasconcelos da Silva (2006).

OLIVER SINNEN
The University of Auckland
New Zealand