

Scheduling Parallel Applications on Heterogeneous Distributed Systems

Guoqi Xie • Gang Zeng • Renfa Li • Keqin Li

Scheduling Parallel Applications on Heterogeneous Distributed Systems

 Springer

Guoqi Xie
College of Computer Science
and Electronic Engineering
Hunan University
Changsha, Hunan, China

Gang Zeng
Graduate School of Engineering
Nagoya University
Nagoya, Aichi, Japan

Renfa Li
Key Laboratory for Embedded and
Cyber-Physical Systems of Hunan Province
Hunan University
Changsha, Hunan, China

Keqin Li
Department of Computer Science
State University of New York
New Paltz, NY, USA

ISBN 978-981-13-6556-0

ISBN 978-981-13-6557-7 (eBook)

<https://doi.org/10.1007/978-981-13-6557-7>

© Springer Nature Singapore Pte Ltd. 2019

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, express 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 imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Foreword

Today, heterogeneous distributed systems exhibit diversification and variation. These systems can be large enough, such as cloud computing systems, or small enough, such as embedded computing systems. Cyber-physical systems (CPS) are emerged heterogeneous distributed systems, which have created one of the hottest research fields in recent years. Automotive CPS (ACPS) are further development of automotive embedded systems. Similarly, cyber-physical cloud systems (CPCS) are further development of cloud computing systems. At present, the relevant results of CPS have gradually been reflected in practical applications. The scheduling problems of parallel applications have been studied for a long time, but for the abovementioned heterogeneous distributed systems, there is an urgent need for customized scheduling algorithms to deal with their individual characteristics and challenges. The publication of this book satisfies this need in a timely manner.

This book first introduces different forms of today's heterogeneous distributed systems. Then, it presents energy-efficient real-time scheduling algorithms for real-time parallel applications on heterogeneous distributed embedded systems, reliability-aware fault-tolerant scheduling algorithms for parallel applications on heterogeneous distributed cloud systems, and high-performance real-time scheduling algorithms for multiple parallel applications on heterogeneous distributed embedded systems. The book further discusses scheduling algorithms and applications on ACPS and CPCS, respectively, which are the two types of CPS.

This book introduces different types of heterogeneous distributed systems and proposes a variety of advanced scheduling algorithms for parallel applications on these systems. This book is rich in content and detailed in graphics. For each proposed algorithm, the book uses corresponding motivational examples to explain clearly and achieve the easy-to-understand purpose; such an approach is a major and unique feature of this book. The book contains the basic knowledge and the latest research development of scheduling theory and methods on heterogeneous distributed systems and is indeed a significant contribution to the field of heterogeneous distributed systems.

This book is a joint effort and creation of four scholars from three countries with dedication and distinction. The authors have published very extensively in

the last few years in the fields of embedded computing, cloud computing, and cyber-physical systems and are undoubtedly the leading scholars in scheduling parallel applications on heterogeneous distributed systems. Finally, I would like to congratulate the authors on a job well done, and I look forward to see the book in print.

Chair Professor of High Performance Computing
and Networking
School of Computer Science
University of Sydney
Sydney, NSW, Australia

Albert Y. Zomaya

Preface

Motivation of the Book

Scheduling parallel applications on heterogeneous distributed systems is a classic research area in computer science and engineering. In recent years, with the emergence and development of embedded computing, cloud computing, and cyber-physical systems (CPS), this research area has shown renewed vitality, challenges, and breakthroughs. Today, heterogeneous distributed embedded systems (e.g., automotive embedded systems) and heterogeneous distributed cloud systems (e.g., cloud-based services, such as Amazon EC2) are typical scenarios of heterogeneous distributed systems. As advanced heterogeneous distributed systems, CPS further enhance the existing embedded and cloud systems. Specifically, automotive CPS (ACPS) and cyber-physical cloud systems (CPCS) are two types of CPS applied to the areas of embedded computing and cloud computing, respectively.

These new distributed systems not only have their own characteristics but also raise new and special requirements for scheduling parallel applications. Parallel application scheduling in the context of energy-efficient scheduling, reliability-aware scheduling and high-performance real-time scheduling, is worthy of investigation on heterogeneous distributed embedded systems, heterogeneous distributed cloud systems, ACPS, and CPCS. The motivation of this book is to provide a comprehensive coverage of our research results in the aforementioned topics in recent years.

Summary of Contents

Chapter 1 introduces heterogeneous distributed systems, including heterogeneous distributed embedded systems and heterogeneous distributed cloud systems. At the same time, ACPS and CPCS are taken as examples to introduce the advanced heterogeneous distributed systems of CPS. Then, this chapter explains the characteristics

of parallel applications and introduces the automotive and workflow applications as examples. For scheduling parallel applications on heterogeneous distributed systems, this chapter analyzes the necessity and challenges of energy-efficient scheduling, reliability-aware scheduling, high performance real-time scheduling, scheduling on ACPS, and scheduling on CPCS. Finally, this chapter lists the outline of this book.

Chapter 2 discusses energy-efficient real-time scheduling for parallel applications. First, this chapter presents an effective energy consumption minimization method for real-time parallel applications on heterogeneous distributed embedded systems implemented by combining the presented non-DVFS and global DVFS-enabled energy-efficient scheduling algorithms. Second, this chapter presents two energy-aware processor merging algorithms for real-time parallel applications in heterogeneous distribution cloud systems by turning off partial processors from an energy-saving perspective.

Chapter 3 studies reliability-aware fault-tolerant scheduling for parallel applications. First, this chapter presents enough and heuristic replication algorithms, respectively, to minimize the redundancy for a parallel application on heterogeneous distributed cloud systems. Second, this chapter further presents quantitative fault-tolerant scheduling algorithms with minimum execution costs and shortest schedule lengths, respectively, for a parallel application on heterogeneous distributed cloud systems. Third, this chapter proposes reliability-aware fault-tolerant scheduling toward resource cost optimization for heterogeneous embedded systems, such as ACPS, during the design phase.

Chapter 4 focuses on high-performance real-time scheduling for parallel applications. First, this chapter presents mixed real-time scheduling of multiple parallel applications on heterogeneous distributed embedded systems. Then, this chapter further presents high-performance mixed-criticality real-time scheduling of multiple parallel applications on heterogeneous distributed embedded systems.

Chapter 5 considers the application on ACPS. This chapter presents fairness-based and adaptive dynamic scheduling algorithms, respectively, on mixed-criticality ACPS, which aims to minimize individual schedule lengths of applications with short overall schedule length of ACPS from a high-performance perspective. The adaptive dynamic scheduling algorithm can respond autonomously to the joint challenges of heterogeneity, dynamics, and parallelism of ACPS.

Chapter 6 covers the application on CPCS. This chapter presents an effective energy management algorithm for multiple real-time parallel workflow applications on CPCS. First, the algorithm can achieve lower deadline missed ratios (DMRs) than the state-of-the-art algorithm. Second, this algorithm can save as much energy as possible with lower DMRs.

Chapter 7 summarizes the book and mentions future research.

Audience and Readership

This book should be a useful reference for the researchers, engineers, and practitioners interested in scheduling theory for embedded computing systems, cloud computing systems, and CPS. The book can be used as a supplement for an advanced undergraduate or a graduate course in distributed computing, embedded computing, cloud computing, and cyber-physical systems in computer science, computing engineering, and electrical engineering. By reading this book, the graduate students and doctoral students will be familiar with the new types of heterogeneous distributed systems and their features, learn a variety of scheduling algorithms, and find a source of inspiration for their own research.

Acknowledgments

The authors are grateful to Professor Albert Zomaya of the University of Sydney in Australia for writing the foreword of this book. We also thank Ms. Jasmine Dou and Mr. Ganesan Senthil Kumar of Springer for their efforts and support in helping to publish this book.

Changsha, Hunan, China
Nagoya, Aichi, Japan
Changsha, Hunan, China
New Paltz, NY, USA

Guoqi Xie
Gang Zeng
Renfa Li
Keqin Li

Contents

1	Introduction	1
1.1	Heterogeneous Distributed Systems	1
1.1.1	Heterogeneous Distributed Embedded Systems	1
1.1.2	Heterogeneous Distributed Cloud Systems	2
1.2	Cyber-Physical Systems	3
1.2.1	Automotive Cyber-Physical Systems	3
1.2.2	Cyber-Physical Cloud Systems	4
1.3	Parallel Applications	4
1.3.1	Parallel Automotive Applications	4
1.3.2	Parallel Workflow Applications	5
1.4	Scheduling	6
1.4.1	Energy-Efficient Scheduling	6
1.4.2	Reliability-Aware Scheduling	6
1.4.3	High Performance Real-Time Scheduling	7
1.4.4	Scheduling on ACPS	8
1.4.5	Scheduling on CPCS	9
1.5	Outline of the Book	10
1.6	Concluding Remarks	10
2	Energy-Efficient Real-Time Scheduling	13
2.1	Introduction	13
2.1.1	Scheduling in Embedded Systems	13
2.1.2	Scheduling in Cloud Systems	14
2.2	Related Work	15
2.3	Models and Preliminaries	17
2.3.1	System Architecture	18
2.3.2	Parallel Application Model	19
2.3.3	Power and Energy Models	20
2.3.4	Problem Statement	23
2.4	Non-DVFS Energy-Efficient Scheduling	23
2.4.1	Lower Bound	24

- 2.4.2 Deadline Slack 26
- 2.4.3 Variable Deadline Slack 29
- 2.4.4 The NDES Algorithm..... 30
- 2.5 Global DVFS-Enabled Energy-Efficient Scheduling 33
 - 2.5.1 Existing EES Algorithm 33
 - 2.5.2 Revision of Time Calculations 34
 - 2.5.3 Energy-Efficient Dynamic Energy Consumption..... 35
 - 2.5.4 The GDES Algorithm..... 37
- 2.6 Energy-Efficient Processor Merging 40
 - 2.6.1 A New Example for Processor Merging 40
 - 2.6.2 Processor Merging 41
 - 2.6.3 Energy-Efficient Processor Merging 43
 - 2.6.4 Quick Energy-Efficient Processor Merging..... 46
- 2.7 Experiments for Energy-Efficiency Algorithm NDES&GDES 47
 - 2.7.1 Experimental Metrics 49
 - 2.7.2 Fast Fourier Transform Application..... 49
 - 2.7.3 Gaussian Elimination Application 52
 - 2.7.4 Randomly Generated Parallel Applications 54
 - 2.7.5 Real Platform Verification 56
- 2.8 Experiments for Energy-Aware Algorithms EPM and QEPM 58
 - 2.8.1 Performance Evaluation Metrics 60
 - 2.8.2 Fast Fourier Transform Application..... 60
 - 2.8.3 Diamond Graph Application 63
 - 2.8.4 Gaussian Elimination Application 65
 - 2.8.5 Randomly Generated Parallel Application..... 66
 - 2.8.6 Summary of Experiments..... 69
- 2.9 Concluding Remarks 69
- 3 Reliability-Aware Fault-Tolerant Scheduling 71**
 - 3.1 Introduction 72
 - 3.1.1 Redundancy Optimization 72
 - 3.1.2 Schedule Length and Cost Optimizations..... 74
 - 3.2 Related Work 76
 - 3.2.1 Reliability-Aware Scheduling 76
 - 3.2.2 Response Time Aware Scheduling 78
 - 3.3 Models and Problem Statement 79
 - 3.3.1 Reliability Model 79
 - 3.3.2 Cost Model 80
 - 3.3.3 Fault-Tolerant Scheduling 81
 - 3.4 Enough Replication for Redundancy Minimization 81
 - 3.4.1 Problem Statement 82
 - 3.4.2 Lower Bound on Redundancy 82
 - 3.4.3 The LBR Algorithm 83
 - 3.4.4 Time Complexity of the LBR Algorithm 84
 - 3.4.5 An Example of The LBR Algorithm 84

- 3.4.6 Enough Replication 84
- 3.4.7 The ERRM Algorithm 86
- 3.4.8 Time Complexity of the ERRM Algorithm 87
- 3.4.9 An Example of The ERRM Algorithm 87
- 3.5 Heuristic Replication for Redundancy Minimization 89
 - 3.5.1 Upper Bound on Reliability Requirement 89
 - 3.5.2 The HRRM Algorithm 90
 - 3.5.3 Time Complexity of the HRRM Algorithm 91
 - 3.5.4 An Example of The HRRM Algorithm 92
- 3.6 Quantitative Fault-Tolerance with Minimum Execution Cost 93
 - 3.6.1 Problem Statement 93
 - 3.6.2 Execution Cost Computation 93
 - 3.6.3 The QFEC Algorithm 94
 - 3.6.4 The QFEC+ Algorithm 96
- 3.7 Quantitative Fault-Tolerance with Shortest Schedule Length 99
 - 3.7.1 Problem Statement 99
 - 3.7.2 The QFSL Algorithm 99
- 3.8 Resource Cost Fault-Tolerant Optimization 101
 - 3.8.1 Verifying Functional Safety Requirement 101
 - 3.8.2 Problem Statement 102
 - 3.8.3 Satisfying Reliability Requirement 102
 - 3.8.4 Minimizing Response Time 104
 - 3.8.5 The VFSR Algorithm 105
 - 3.8.6 Time Complexity of the VFSR Algorithm 106
 - 3.8.7 An Example of the VFSR Algorithm 107
- 3.9 Resource Cost-Aware Fault-Tolerance Optimization 108
 - 3.9.1 Satisfying Reliability Requirement 108
 - 3.9.2 Satisfying Response Time Requirement 109
 - 3.9.3 Optimizing Resource Cost 111
 - 3.9.4 The RCFO Algorithm 113
- 3.10 Experiments for Redundancy Optimization Algorithms 115
 - ERRM and HRRM 115
 - 3.10.1 Performance Evaluation Metrics 115
 - 3.10.2 Fast Fourier transform application 116
 - 3.10.3 Gaussian Elimination Application 119
 - 3.10.4 Randomly Generated Parallel Application 121
 - 3.10.5 Summary of Experiments for ERRM and HRRM 123
- 3.11 Experiments for Execution Cost Optimization Algorithms 123
 - QFEC, QFEC+, QFSL and QFSL+ 123
 - 3.11.1 Performance Evaluation Metrics 124
 - 3.11.2 Low-Parallelism Workflows 125
 - 3.11.3 High-Parallelism Workflows 127
 - 3.11.4 Workflows Statistics 128
 - 3.11.5 Summary of Experiments For QFEC, QFEC+, QFSL and QFSL+ 129

- 3.12 Experiments for Resource Cost Optimization Algorithms
 - VFSR and RCFO 130
 - 3.12.1 Performance Evaluation Metrics 131
 - 3.12.2 Real-Life End-to-End Distributed Automotive Function .. 132
 - 3.12.3 Synthetic End-to-End Distributed Function 140
- 3.13 Concluding Remarks 144
- 4 High-Performance Real-Time Scheduling 147**
 - 4.1 Introduction 147
 - 4.1.1 Scheduling with Priority 147
 - 4.1.2 Scheduling with Criticality 148
 - 4.2 Related Work 149
 - 4.2.1 High Performance Scheduling..... 149
 - 4.2.2 Real-Time Scheduling 149
 - 4.3 Modeling with Priority 150
 - 4.3.1 System Model 150
 - 4.3.2 Motivating Example 151
 - 4.4 Fairness Scheduling 153
 - 4.4.1 Lower Bound and Deadline 153
 - 4.4.2 Scheduling Framework 153
 - 4.4.3 The F_MHEFT Algorithm 155
 - 4.5 Whole Priority Scheduling 156
 - 4.5.1 The WP_MHEFT Algorithm 156
 - 4.5.2 An Example of the WP_MHEFT Algorithm 157
 - 4.6 Partial Priority Scheduling 158
 - 4.6.1 The PP_MHEFT Algorithm 158
 - 4.6.2 An Example of the PP_MHEFT Algorithm 159
 - 4.6.3 Summarization of the Three Algorithm..... 162
 - 4.7 Mixed-Criticality Scheduling 162
 - 4.7.1 System Architecture 162
 - 4.7.2 Mixed-Criticality Model 163
 - 4.7.3 Motivating Example 164
 - 4.7.4 Deadline Slack 165
 - 4.7.5 The D_MHEFT Algorithm 165
 - 4.8 Experiments For Fairness and Priority Scheduling Algorithm
 - PP_MHEFT 170
 - 4.8.1 Performance Evaluation Metrics 170
 - 4.8.2 Experimental Results 171
 - 4.9 Experiments For Mixed-Criticality Scheduling Algorithms
 - F_MHEFT and D_MHEFT 175
 - 4.9.1 Performance Evaluation Metrics 175
 - 4.9.2 Experimental Results 175
 - 4.10 Concluding Remarks 178

- 5 Application on Automotive Cyber-Physical Systems** 181
 - 5.1 Introduction 181
 - 5.2 Related Work 182
 - 5.3 Models 184
 - 5.3.1 Architecture 184
 - 5.3.2 Mixed-Criticality Dynamic Model 185
 - 5.3.3 Motivating Example 185
 - 5.3.4 Problem Statement 187
 - 5.4 Fairness-Based Dynamic Scheduling 187
 - 5.4.1 Lower Bound and Deadline 187
 - 5.4.2 Dynamic Scheduling Framework 188
 - 5.4.3 Fairness-Based Dynamic Scheduling Algorithm 190
 - 5.4.4 An Example of the FDS_MIMF Algorithm 192
 - 5.5 Adaptive Dynamic Scheduling 195
 - 5.5.1 Absolute Deadline 195
 - 5.5.2 The ADS_MIMF Algorithm 195
 - 5.5.3 An Example of the ADS_MIMF Algorithm 198
 - 5.6 Experiments For Dynamic Scheduling Algorithms
FDS_MIMF and ADS_MIMF 201
 - 5.6.1 Performance Evaluation Metrics 201
 - 5.6.2 Experimental Analysis 202
 - 5.7 Concluding Remarks 205
- 6 Application on Cyber-Physical Cloud Systems** 207
 - 6.1 Introduction 207
 - 6.2 Related Work 208
 - 6.3 Models 209
 - 6.3.1 CPCS Architecture 209
 - 6.3.2 Parallel Workflow Application Model 211
 - 6.3.3 Power and Energy Models 213
 - 6.3.4 Lower Bound and Deadline 213
 - 6.3.5 Problem Statement 214
 - 6.4 Proposed Energy Management Algorithms 215
 - 6.4.1 Existing DEWTS for Single Real-Time Parallel
Workflow Application 215
 - 6.4.2 Reusable DEWTS for Multiple Real-Time Parallel
Workflow Applications 217
 - 6.4.3 Minimizing DMR for Multiple Real-Time Parallel
Workflow Applications 218
 - 6.4.4 Minimizing Energy Consumption for Multiple
Real-Time Parallel Workflow Applications 221
 - 6.4.5 Summary of Algorithms 227

- 6.5 Experiments for Multiple Parallel Application Energy Optimization Algorithm DPMMW&GESMW 228
 - 6.5.1 Performance Evaluation Metrics 228
 - 6.5.2 Small-Scale CPCS with Multiple Workflows..... 228
 - 6.5.3 Large-Scale CPCS with Multiple Workflows..... 229
 - 6.5.4 Mixed-Scale CPCS with Multiple Workflows 230
 - 6.5.5 Summary of Experiments..... 231
- 6.6 Concluding Remarks 232
- 7 Summary and Future Research 233**
 - 7.1 Summary 233
 - 7.2 Future Research..... 235
- References 237**