Workflow Scheduling on Computing Systems

Kenli Li, Xiaoyong Tang, Jing Mei, Longxin Zhang, Wangdong Yang, and Keqin Li
Contents

List of Figures xiii
List of Tables xvii
Foreword xxi
Author Bios xxiii
Preface xxvii

CHAPTER 1 • Computing Systems 1

1.1 COMPUTING SYSTEMS RESOURCE MANAGEMENT 1
1.2 THE WELL KNOWN SYSTEMS 2
  1.2.1 SLURM 2
  1.2.2 PBS 5
  1.2.3 YARN 7
1.3 PARALLEL APPLICATIONS 11
  1.3.1 Workflow Applications 12
  1.3.2 Classical Tasks DAG Model 12
1.4 SOME REAL-WORLD WORKFLOW APPLICATIONS 13
  1.4.1 Montage 14
  1.4.2 Broadband 16
  1.4.3 Epigenomics 17
  1.4.4 LIGO Inspiral Analysis 18
1.5 OUTLINE OF THE BOOK

CHAPTER 2 • Classical Workflow Scheduling

2.1 TASK SCHEDULING

2.2 SCHEDULING CHALLENGES
   2.2.1 Energy-Efficient Scheduling
   2.2.2 Reliability-Aware Scheduling
   2.2.3 High Performance Real-Time Scheduling

2.3 SCHEDULING ALGORITHMS CLASSIFICATION
   2.3.1 Local versus Global
   2.3.2 Static versus Dynamic
   2.3.3 Optimal versus Suboptimal
   2.3.4 Approximate versus Heuristic
   2.3.5 Centralized versus Distributed

2.4 SEVERAL HEURISTIC WORKFLOW SCHEDULING ALGORITHMS
   2.4.1 DLS
   2.4.2 MCP
   2.4.3 HEFT

2.5 SUMMARY

CHAPTER 3 • Stochastic Task Scheduling on Grid Computing Systems

3.1 INTRODUCTION

3.2 THE GRID SCHEDULING ARCHITECTURE

3.3 STOCHASTIC SCHEDULING PROBLEM
   3.3.1 The Random Variable Approximate Weight
   3.3.2 Stochastic Scheduling Attributes
CHAPTER 5 • Reliability-Energy-Aware Scheduling algorithm

5.1 INTRODUCTION

5.2 SYSTEM MODELS
5.2.1 Task Scheduling Architecture
5.2.2 Heterogeneous Computing Systems
5.2.3 Parallel Application Workflow DAG
5.2.4 Energy Consumption Model

5.3 SYSTEM RELIABILITY ANALYSIS
5.3.1 Single Processor Failure Rate
5.3.2 Application Reliability Analysis

5.4 THE RELIABILITY-ENERGY AWARE SCHEDULING ALGORITHM
5.4.1 Task Priorities Phase
5.4.2 Task Assignment Phase
5.4.3 Slack Reclamation

5.5 EXPERIMENTAL RESULTS AND DISCUSSION
5.5.1 Simulation Environment
5.5.2 Randomly Generated Application
5.5.3 Various Weight \( \theta \) of REAS Algorithm
5.5.4 The Real-World Applications Results

5.6 SUMMARY

CHAPTER 6 • Energy Consumption and Reliability Bi-objective Workflow Scheduling

6.1 INTRODUCTION

6.2 MODELS AND PRELIMINARIES
6.2.1 Workflow Model
6.2.2 System Model 96
6.2.3 Energy Model 96
6.2.4 Reliability Model 98
6.2.5 Problem Definition 98

6.3 MULTI-OBJECTIVE OPTIMIZATION AND A MOTIVATIONAL EXAMPLE 99
6.3.1 Multi-Objective Optimization Problem Overview 99
6.3.2 A Motivational Example 101

6.4 ALGORITHMS 102
6.4.1 Encoding 102
6.4.2 Initial Population 103
6.4.3 Fitness Measure 103
6.4.4 Selection 103
6.4.5 Two-Point Crossover 104
6.4.6 Mutation 107
6.4.7 The Main Algorithm 109

6.5 PERFORMANCES EVALUATION 111
6.5.1 Performance Metrics 111
6.5.2 Experimental Setting 111
6.5.3 Real World Application Graphs 113
   6.5.3.1 Three Kinds of Classic DAG Graphs 113
   6.5.3.2 Molecular Dynamic Code 116
6.5.4 Randomly Generated Application Graphs 117

6.6 SUMMARY 119

CHAPTER 7 Interconnection Network Energy-Aware Scheduling Algorithm 121

7.1 INTRODUCTION 121
8.2.3 Performance Measures 147

8.3 RESOURCE-AWARE SCHEDULING ALGORITHM WITH DUPLICATION MINIMIZATION (RADMS) 149

8.3.1 Task Prioritization Stage 149
8.3.2 Task Mapping Stage 150
8.3.3 Redundancy Deletion Stage 153
8.3.4 A Scheduling Example 156

8.4 DUPLICATION OPTIMIZING SCHEME 159

8.4.1 Analysis on Generation of Redundancy 159
8.4.2 Strategies of Redundancy Exploitation 159
  8.4.2.1 Move Tasks to the LFT 160
  8.4.2.2 Move Tasks to the EST 162
  8.4.2.3 Migrate Tasks among Processors 164

8.5 EXPERIMENTAL RESULTS AND ANALYSIS 168

8.5.1 Experimental Metrics 168
8.5.2 Parameter Settings 169
8.5.3 Experimental Results and Analysis 170
  8.5.3.1 Effect of Task Number 170
  8.5.3.2 Effect of Processor Number 171
  8.5.3.3 Effect of Parallelism Factor 172
  8.5.3.4 Effect of CCR 173
  8.5.3.5 Makespan Improvement 173

8.6 SUMMARY 174

CHAPTER 9 • Contention-Aware Reliability Efficient Scheduling 175

9.1 INTRODUCTION 175

9.2 MODELS AND PRELIMINARIES 176
  9.2.1 Application Model 176
Contents

9.2.2 Communication Contention Model 177
9.2.3 Energy Model 179
9.2.4 Reliability Model 179

9.3 PRELIMINARIES 180
9.3.1 Task Priority 180
9.3.2 Problem Description 181
9.3.3 Motivational Example 181

9.4 CONTENTION-AWARE RELIABILITY MANAGEMENT SCHEME 182

9.5 EXPERIMENTS 184
9.5.1 Performance Metrics 186
  9.5.1.1 Scheduling Length Ratio (SLR) 186
  9.5.1.2 Energy Consumption Ratio (ECR) 186
  9.5.1.3 POF 187
9.5.2 Randomly Generated DAG 187
9.5.3 Effect of Random Applications 188
9.5.4 Real-World Application DAG 190
  9.5.4.1 LU Decomposition 191
  9.5.4.2 Fast Fourier Transform 192
  9.5.4.3 Molecular Dynamic Code 194

9.6 SUMMARY 196

Bibliography 197
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The main architecture of SLURM</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>SLURM entities</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>PBS structure</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>The structure of YARN</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>The application execution on YARN</td>
<td>10</td>
</tr>
<tr>
<td>1.6</td>
<td>An example of workflow application DAG model</td>
<td>14</td>
</tr>
<tr>
<td>1.7</td>
<td>Montage workflow</td>
<td>15</td>
</tr>
<tr>
<td>1.8</td>
<td>Broadband workflow</td>
<td>16</td>
</tr>
<tr>
<td>1.9</td>
<td>Epigenomics workflow</td>
<td>18</td>
</tr>
<tr>
<td>1.10</td>
<td>LIGO workflow</td>
<td>18</td>
</tr>
<tr>
<td>2.1</td>
<td>A hierarchical taxonomy for task scheduling</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>DAG task diagram example</td>
<td>30</td>
</tr>
<tr>
<td>2.3</td>
<td>A simple arbitrary processor network topology diagram</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>Grid scheduling architecture</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Experimental results of 100 tasks. (a) makespan; (b) speedup; (c) makespan standard deviation</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Experimental results of 200 tasks. (a) makespan; (b) speedup; (c) makespan standard deviation</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Experimental results of 300 tasks. (a) makespan; (b) speedup; (c) makespan standard deviation</td>
<td>49</td>
</tr>
</tbody>
</table>
4.1 A workflow application with normal distribution 56
4.2 Stochastic workflow DAG series and parallel model 61
4.3 The operator $\mathcal{R}$ 68
4.4 An illustration example. (a) DLS (makespan = 15.33); (b) SDLS (makespan = 14.46) 69
4.5 Examples of workflow DAGs. (a) A low parallelism degree application; (b) A high parallelism degree application 70

5.1 The reliability-energy aware workflow scheduling architecture 76
5.2 The example of workflow application DAG graph 77
5.3 The experimental results of real-world DSP problem. (a) schedule length; (b) energy consumption; (c) reliability 90

6.1 A simple DAG 96
6.2 Multi-objective optimization 100
6.3 The selection procedure 105
6.4 Two-point crossover 106
6.5 Mutation 108
6.6 Comparisons of Gauss-Jordan (the number of processors equals 3 with CCR = 1.0) 114
6.7 Comparisons of Laplace (the number of processors equals 3 with CCR = 1.0) 114
6.8 Comparisons of LU (the number of processors equals 3 with CCR = 1.0) 115
6.9 A molecular graph 116
6.10 Comparisons of molecular graph (the number of processors equals 6 with CCR = 1.0) 117
6.11 Comparisons of randomly generated DAG graph (the number of task graphs equals to 100, the number of processors equals 6 with CCR = 0.5) 118
6.12 Comparisons of randomly generated DAG graph (the number of task graphs equals to 100, the number of processors equals 8 with CCR = 1.0) 118
6.13 Comparisons of randomly generated DAG graph (the number of task graphs equals to 100, the number of processors equals 6 with CCR = 5.0) 119

7.1 A heterogeneous computing systems fat-tree architecture 124
7.2 The illustration of extended DAG mode 125
7.3 Task scheduling across computing nodes problem 127
7.4 Network routing chip data communication time 128
7.5 The results of varying CCR. (a) LIGO; (b) Montage 137
7.6 The results of varying deadline. (a) LIGO; (b) Montage 139

8.1 Heterogeneous distributed system architecture 145
8.2 An example of DAG 146
8.3 A duplication-based schedule of the example DAG 148
8.4 Determining the most-suitable time slot to duplicate $t_j$ 153
8.5 Schedule of tasks $t_1, t_2, t_3, t_4, t_6, t_8, t_7,$ and $t_5$ 156
8.6 Schedule after deleting redundancy of $t_1$ 158
8.7 Schedule of tasks $t_9, t_{10}, t_{11}, t_{12},$ and $t_{13}$ 158
8.8 Schedule after deleting redundancy of $t_2$ and $t_8$ 158
8.9 Schedule after tasks moving to the LFT and redundancy deletion 162
8.10 Schedule after tasks moving to the EST and redundancy deletion 165
8.11 Schedule after tasks migration 168
8.12 Effect of task number on performance 170
8.13 Effect of processor number on performance 171
8.14 Effect of parallelism factor on performance 172
8.15 Effect of CCR on performance 173

9.1 Simple DAG 178
9.2 Link model 178
9.3 Scheduling of task graph in Figure 9.1. (a) schedule without contention; (b) schedule under CARMED with contention 182
9.4 Effect of varying task number for CCR = 0.5 188
9.5 Effect of varying task number for CCR = 1.0 189
9.6 Effect of varying task for CCR = 10 189
9.7 Effect of varying task number for CCR = 5 and DAGsize = 100 189
9.8 LU-decomposition task graph 191
9.9 Effect of varying CCR the LU decomposition task graph 192
9.10 FFT with four points 193
9.11 Effect of varying CCR for the FFT task graph 194
9.12 Effect of varying CCR for the molecular dynamics code task graph 195
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Taxonomy of task scheduling strategies</td>
<td>25</td>
</tr>
<tr>
<td>2.2</td>
<td>The attribute value in Figure 2.2</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>Application task execution size and edge data communication on Figure 1.6</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Some symbols used in this chapter</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Performance impact of 10 machines for makespan</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance impact of 10 machines for speedup</td>
<td>51</td>
</tr>
<tr>
<td>3.5</td>
<td>Performance impact of 10 machines for makespan standard deviation</td>
<td>51</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance impact of 16 machines for makespan</td>
<td>52</td>
</tr>
<tr>
<td>3.7</td>
<td>Performance impact of 16 machines for speedup</td>
<td>52</td>
</tr>
<tr>
<td>3.8</td>
<td>Performance impact of 16 machines for makespan standard deviation</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>The deterministic scheduling of Figure 4.1</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>The processing time of Figure 4.1 on cluster systems</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Notations and definitions</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>The \textit{sblevel} of sample DAG tasks in Figure 4.1</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>The special workflow DAG experimental results about makespan with Figure 4.5(a)</td>
<td>71</td>
</tr>
<tr>
<td>4.6</td>
<td>The special workflow DAG experimental results about speedup with Figure 4.5(a)</td>
<td>71</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.7</td>
<td>The special workflow DAG experimental results about makespan with Figure 4.5(b)</td>
<td>71</td>
</tr>
<tr>
<td>4.8</td>
<td>The special workflow DAG experimental results about speedup with Figure 4.5(b)</td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>The parameters of computing systems processors</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>The task estimation execution matrix $[w_{i,k,h}]$</td>
<td>78</td>
</tr>
<tr>
<td>5.3</td>
<td>The estimation data communication matrix $[a_{i,j}]$</td>
<td>78</td>
</tr>
<tr>
<td>5.4</td>
<td>The workflow DAG task $b_{level}$ value</td>
<td>86</td>
</tr>
<tr>
<td>5.5</td>
<td>The schedule length of REAS algorithm with various weight $\theta$</td>
<td>89</td>
</tr>
<tr>
<td>5.6</td>
<td>The energy consumption of REAS algorithm with various weight $\theta$</td>
<td>90</td>
</tr>
<tr>
<td>5.7</td>
<td>The reliability of REAS algorithm with various weight $\theta$</td>
<td>90</td>
</tr>
<tr>
<td>6.1</td>
<td>Voltage-relative frequency combinations</td>
<td>97</td>
</tr>
<tr>
<td>6.2</td>
<td>Computation costs on different processors</td>
<td>102</td>
</tr>
<tr>
<td>6.3</td>
<td>Selected workflow models</td>
<td>112</td>
</tr>
<tr>
<td>7.1</td>
<td>Tasks $t_{level}$, $d_{element}$, and $d_{li}$ of Figure 7.2</td>
<td>130</td>
</tr>
<tr>
<td>7.2</td>
<td>The adjusted unscheduled tasks for Figure 7.2</td>
<td>131</td>
</tr>
<tr>
<td>8.1</td>
<td>Notations used in this chapter</td>
<td>144</td>
</tr>
<tr>
<td>8.2</td>
<td>WCETs of tasks on different processors</td>
<td>146</td>
</tr>
<tr>
<td>8.3</td>
<td>The upward ranks of tasks in the motivation application</td>
<td>150</td>
</tr>
<tr>
<td>8.4</td>
<td>A comparison of makespan for random DAGs</td>
<td>174</td>
</tr>
<tr>
<td>9.1</td>
<td>Notations used in this chapter</td>
<td>177</td>
</tr>
<tr>
<td>9.2</td>
<td>Computation costs on different processors</td>
<td>181</td>
</tr>
</tbody>
</table>
9.3 Parameter configuration for the LU task graphs 191
9.4 Parameter configuration for the FFT task graphs 193
Foreword

In recent years, with the popularity of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components, it is possible to construct large-scale parallel and distributed computing systems, such as cluster systems, supercomputers, grid computing, cloud computing, and edge/fog computing. These technical opportunities enable the sharing, selection, and aggregation of geographically distributed heterogeneous resources to solve science, engineering, and commerce problems. Resource management plays a key role in improving the performance of these systems, and especially effective and efficient scheduling methods are fundamentally important. However, the systems face a lot of challenging problems, such as energy consumption, reliability, resource utilization, cost, instability, and resource contention. Workflow scheduling aims at meeting user demands and resource provider management indicators, while maintaining a good overall performance or throughput for computing systems. The publication of this book satisfies this need in a timely manner.

This book offers a systematic presentation of workflow scheduling, which encompasses the systems architecture, scheduling model, energy consumption, reliability, resource utilization, problem formulation, billing mechanisms, and the detailed discussion of the theoretical underpinnings, design methodology, and practical implementation. This book is rich in content and detailed in graphics. For each presented algorithm, the book uses corresponding motivational examples to explain clearly and achieve the easy-to-understand purpose. In particular, the book:
Offers a comprehensive overview of computing systems workflow scheduling techniques about systems, scheduling architecture, energy consumption, reliability, resource utilization, problem formulation, billing mechanism, methods, design considerations, and practical implementation.

Presents the design principles necessary for analyzing the computing systems requirements, objectives, time complexity and constraints, that will guide engineering students and engineers toward achieving high-performance, low-cost, and efficient resource management systems.

Demonstrates the practical implementation of workflow scheduling and their design guidelines and optimizations that can be directly adopted in engineering application and research work.

Provides a complete perspective on workflow scheduling that hopefully can inspire appreciation and better understanding of the subject matter.

It is a great pleasure to introduce this Workflow Scheduling on Computing Systems, which is a joint effort and creation of six scholars with dedication and distinction. The authors have published very extensively in the fields of grid computing systems, cluster systems, cloud computing, and are undoubtedly the leading scholars in scheduling workflow parallel applications on computing systems. Finally, I would like to congratulate the authors on their excellent work, and I look forward to see the publication of this book.

Kai Hwang
Presidential Chair Professor
Chinese University of Hong Kong
Shenzhen, China
Author Bios

Kenli Li (Senior Member, IEEE) received his PhD in computer science from the Huazhong University of Science and Technology, China, in 2003. He was a visiting scholar at the University of Illinois at Urbana-Champaign, Champaign, Illinois from 2004 to 2005. He is currently a full professor of computer science and technology at Hunan University, China, and deputy director of National Supercomputing Center in Changsha. His major research areas include parallel computing, high-performance computing, grid and cloud computing. He has published more than 130 research papers in international conferences and journals such as the IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Signal Processing, Journal of Parallel and Distributed Computing, ICPP, and CCGrid. He is an outstanding member of CCF. He is serves on the editorial board of the IEEE Transactions on Computers.

Xiaoyong Tang received his M.S. and Ph.D. degrees from Hunan University, China, in 2007 and 2013, respectively. He is currently Professor at the School of Computer and Communication Engineering, Changsha University of Science and Technology. His research interests include parallel computing, cloud computing, big data, modeling and scheduling of distributed computing systems, distributed system reliability, and parallel algorithms. He has published more than 50 technique papers in international journals and conferences. He holds over 10 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the World’s Top 2%
Scientists. He is the reviewers of TC, TPDS, TII, JPDC, FGCS, CPE, JS, and so on.

**Jing Mei** received her Ph.D in computer science from Hunan University, China, in 2015. She is currently an assistant professor in the College of Information Science and Engineering in Hunan Normal University. Her research interests include parallel and distributed computing, cloud computing, edge computing etc. She has published 16 research articles in international conference and journals, such as IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed System, IEEE Transactions on Service Computing, Cluster Computing, Journal of Grid Computing, and Journal of Supercomputing.

**Longxin Zhang** received his Ph.D. in computer science from Hunan University, China, in 2015. He is currently Associate Professor of computer science at the Hunan University of Technology. He is also a Visiting Scholar of the University of Florida. His major research interests include modeling and scheduling for distributed computing systems, distributed system reliability, parallel algorithms, cloud computing, and deep learning. He has published more than 20 technique papers in international journals and conferences. He is the reviewer of IEEE TPDS, ACM TIST,IEEE TII, IEEE IOT, INS, ASC, and so on.

**Wangdong Yang** received his PhD in computer science and technology from Hunan University, China. He is a professor of computer science and technology in Hunan University, China. His research interests include modeling and programming for heterogeneous computing systems and parallel algorithms. He has published more than 30 papers in international conferences and journals such as the IEEE Transactions on Computers, the IEEE Transactions on Parallel and Distributed Systems, etc.

**Keqin Li** is a a SUNY Distinguished Professor of computer science at the State University of New York. He is also a National Distinguished Professor of Hunan University, China. His current research interests include cloud computing, fog computing...
and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored more than 840 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds over 60 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world’s top 5 most influential scientists in distributed computing based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the ACM Computing Surveys and the CCF Transactions on High Performance Computing. He has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.
MOTIVATION OF THE BOOK

In the past few years, with the rapid development of IT technology, computing systems have become the core infrastructure of social economy. However, with the exponential growth of computing and data storage requirements, computing systems are facing with a lot of challenging problems, such as energy consumption, reliability, resource utilization, cost, stochastic computation, and resource contention. Workflow scheduling aims at meeting user demands and resource provider management indicators while maintaining a good overall performance or throughput for such systems.

With the increasingly prominent role of workflow scheduling on computing systems, it is timely to introduce the workflow scheduling technology, including the basic concept of workflow scheduling, stochastic tasks scheduling, reliability-driven scheduling, reliability-energy-aware scheduling, interconnection network-aware scheduling, and resource-aware duplication optimization scheduling. To the best of our knowledge, although many books about job or task scheduling already exist, these books lack to provide a comprehensive review and thorough discussion of workflow scheduling. Educating and imparting the holistic understanding of workflow scheduling on computing systems has laid a strong foundation for postgraduate students, research scholars, and practicing engineers in generating and innovating solutions and products for a broad range of applications.

In recognition of this, the book *Workflow Scheduling on Computing Systems* is intended to provide a coverage on the
theoretical and practical aspects of the subject matter, which includes not only the conventional workflow scheduling but also the systems challenging problems, such as energy consumption, reliability, resource utilization, cost, and all of which stem from the authors’ own research work.

SUMMARY OF CONTENTS

This book focuses on workflow scheduling on computing systems. The main contents are summarized as follows.

Chapter 1 introduces the working principle of resource management and some typical resource managements (such as SLURM, PBS, YARN) in computing systems. Then, this chapter presents the practical application of workflow DAG model and real-world workflow applications.

In Chapter 2, we introduce the scheduling problems, workflow task scheduling, scheduling challenges, and the classification of scheduling algorithms. We also list several typical heuristic workflow scheduling algorithms such as DLS, MCP, HEFT.

Chapter 3 focuses on the stochastic scheduling problem on grid computing systems. In order to effectively scheduling precedence constrained stochastic tasks, this chapter present a stochastic heterogeneous earliest finish time scheduling algorithm, which incorporate the stochastic attribute, such as expected value and variance, of task processing time and edge communication time into scheduling.

Chapter 4 emphasizes the scheduling stochastic parallel applications with precedence constrained tasks on heterogeneous cluster systems. It formulates the stochastic task scheduling model and develops effective methods to deal with the normally distributed random variables. This chapter also describes a stochastic dynamic level scheduling algorithm SDLS, which employs stochastic bottom level and stochastic dynamic level to produce schedules of high quality.
In Chapter 5, we first build a reliability and energy-aware task scheduling architecture including precedence-constrained parallel applications, energy consumption model on heterogeneous systems. Then, we present the single processor failure rate model based on Dynamic Voltage and Frequency Adjustment (DVFS) technique and deduce the application reliability of systems. Finally, to provide an optimum solution for this problem, a heuristic reliability-energy aware scheduling algorithm is presented.

Chapter 6 addresses a bi-objective genetic algorithm to deal with the bi-objective optimization problem of high system reliability and low energy consumption for parallel tasks. This approach offers users more flexibility when jobs are submitted to a data center.

Chapter 7 comprehensively presents the issues of heterogeneous systems, energy consumption of processors and interconnection networks, computation-intensive scientific workflow applications with deadline constraints, and task scheduling. This chapter also presents a network energy-efficient workflow task scheduling algorithm that consists of task level computing, task subdeadline initialization, dynamic adjustment, and a data communication optimization method.

In Chapter 8, we present a novel resource-aware scheduling algorithm called RADS, which searches and deletes redundant task duplications dynamically in the process of scheduling. A further optimizing scheme is designed for the schedules generated by our algorithm, which can further reduce resource consumption without degrading the makespan.

Chapter 9 presents a novel contention-aware reliability management algorithm for parallel tasks in heterogeneous systems. Given that majority of previous studies do not consider the realistic existence of contention in modern communication systems, the algorithm is presented in the current study by applying DVFS and slack reclaiming techniques.
AUDIENCE AND READERSHIP
This book should be a useful reference for researchers, engineers, and practitioners interested in scheduling theory for computing systems. The book can be used as a supplement for graduate students and system developers whose major areas of interest are in resource management of cluster, supercomputers, grid computing, cloud computing, edge/fog computing systems, and related fields, as well as engineering professionals from both academia and computing systems development companies. By reading this book, readers will be familiar with new types of computing systems and their features, will learn a variety of scheduling algorithms, and find a source of inspiration for their own research.

ACKNOWLEDGMENTS
This book is published with financial support from National Key R&D Programs of China (Grant No. 2020YFB2104000), National Natural Science Foundation of China (Grant No. 61972146), Hunan Provincial Natural Science Foundation of China (Grant No. 2020JJ4376).