Contents

Acknowledgements **xi**

1 Introduction **1**

1.1 Motivation 1
1.2 Parallel programming models 4
1.3 Synchronization Architecture 6
1.4 Problem statement and thesis 9
1.5 Approach 10
1.6 Outline 11

2 Conceptual approach **13**

2.1 Parallel models and definitions 14
2.1.1 Parallelism and parallel computing 14
2.1.2 Modeling 15
2.1.3 PCMs and PMMs 15
2.1.4 Detail levels 18
2.1.5 Requirements of PPMs 18
2.2 Synchronization architecture 20
2.2.1 Types of synchronization: CS, ME 20
2.2.2 Mutual exclusion, mapping and bounded resources 21
2.2.3 Classification criteria for SAs 24
2.2.4 Condition synchronization: CS classes 26
2.2.5 Mutual exclusion: ME classes 30
2.2.6 Mutual exclusion vs. condition synchronization 32
2.2.7 Data-Dependency: DS, NDS classes 33
2.3 Execution-level models 35
2.3.1 SA class of machine models 36
2.3.2 Conclusions about execution models 39
2.4 Bridging models and cost models 40
2.4.1 Class (SP,NME,NDS): PRAM 40
2.4.2 Class (SP,ME,NDS): BSP 46

2.4.3 Class (SP,ME,NDS): QSM 53
2.4.4 Class (NSP,ME,DS): LogP 55
2.4.5 Conclusions about PCMs SA 59
2.5 Parallel programming languages and models 61
2.5.1 Class (SP,NME,DS): Pure nested parallelism 61
2.5.2 Class (SP,ME,NDS): Nested parallelism with ME 62
2.5.3 Class(NSP,NME,NDS): Mapping oriented models 65
2.5.4 Class (NSP,ME,DS): Message passing 68
2.5.5 Class (NSP,ME,DS): Maximum abstraction 69
2.5.6 Conclusions about PPLs/PPMs SA 71
2.6 Synchronization architecture of applications 73
2.6.1 Class (SP,ME,NDS/DS) 74
2.6.2 Class (NSP,ME,NDS/DS) 75
2.6.3 Class (SP,NME,NDS) 75
2.6.4 Class (SP, NME, DS) 76
2.6.5 Class (NSP,NME,NDS) 77
2.6.6 Class (NSP,NME,DS) 79
2.6.7 Conclusions about applications SA 80
2.7 Summary 81

3 Theoretical approach 83
3.1 Graph preliminaries 84
3.1.1 Basic graph concepts and notations 85
3.1.2 Transitivities 88
3.1.3 Topological graph parameters 89
3.1.4 Task graphs 90
3.2 Series-parallel graphs 92
3.2.1 Definitions 92
3.2.2 Distance from NSP to SP graphs 96
3.3 Transformation problem (NSP to SP) 98
3.3.1 SP-ization 98
3.3.2 Local resynchronization 101
3.3.3 Combinations of NSP problems 104
3.3.4 Simple SP-ization techniques 109
3.4 Algorithm 1: Local exploration 112
3.4.1 Notations 112
3.4.2 SP-ization technique 113
3.4.3 JF combinations 116
3.4.4 Mixing problems through JF combinations 117
3.4.5 Example 118
3.4.6 Complexity 120
3.5 Algorithm 2: Local layering technique 123

3.5.1 Notations 123
3.5.2 Algorithm description 124
3.5.3 Example 127
3.5.4 Correctness 131
3.5.5 Critical path property for UTC graphs 132
3.5.6 Complexity 133
3.5.7 Implementation 134
3.5.8 Improvements 136
3.6 Measuring the SP-ization impact 136
3.6.1 Potential performance impact 137
3.6.2 Structural impact 140
3.6.3 Algorithms comparison 143
3.6.4 Analytical models 145
3.6.5 Conclusions about SP-ization impact 147
3.7 Summary 148

4 Experimental study 151
4.1 Synthetic graphs 152
4.1.1 Workload modeling 153
4.1.2 Random sample of the graph space 155
4.1.3 Meshes 160
4.1.4 Conclusions about synthetic graph results 178
4.2 Real Applications 180
4.2.1 Experiments design 180
4.2.2 Application cost models at programming level 188
4.2.3 Application cost models at implementation level 198
4.2.4 Static applications results 207
4.2.5 Dynamic applications results 215
4.2.6 Conclusions about real application results 226
4.3 Summary 226

5 Conclusion 229
5.1 Contributions 230
5.2 Conclusions 231

Bibliography 233