

Contents

Certificate	iii
Declaration by the Candidate	v
Copyright Transfer Certificate	vii
Acknowledgements	ix
Contents	xi
List of Figures	xv
List of Tables	xix
Abbreviations	xxiii
Symbols	xxvii
Preface	xxix
1 Introduction	1
1.1 Background	3
1.1.1 Distributed Systems	3
1.1.2 Distributed Computing versus Parallel Computing	5
1.1.3 Task Scheduling in Distributed Computing	5
1.2 Motivation	7
1.3 Objective	7
1.4 Thesis Contributions	8
1.5 Thesis Organization	8

2	Preliminaries and Literature Review	13
2.1	Preliminaries	13
2.1.1	Assumptions	13
2.1.2	Definitions	14
2.1.3	Task Graph	17
2.1.4	Task Scheduling Problem	18
2.1.5	Real-world Application Graphs	19
2.1.5.1	Gaussian Elimination	19
2.1.5.2	Fast Fourier Transform	19
2.1.5.3	Systolic Array	20
2.1.5.4	Montage Workflow	21
2.1.5.5	Epigenomics Workflow	22
2.2	Literature Review	23
2.2.1	List Scheduling Algorithms	23
2.2.2	Duplication-based Scheduling Algorithms	27
2.2.3	Clustering-based Scheduling Algorithms	31
2.2.4	Outcome of Literature Review	34
2.3	Summary	35
3	Benchmarking Task Scheduling Algorithms and a Possible Framework	37
3.1	Benchmarking	38
3.2	Metrics	39
3.2.1	Schedule Length Ratio	39
3.2.2	Speedup	40
3.2.3	Efficiency	40
3.2.4	Frequency of best schedules	40
3.2.5	Slack	40
3.3	Algorithms considered for Benchmarking	41
3.3.1	HEFT Algorithm	41
3.3.2	PETS Algorithm	42
3.3.3	LDCP Algorithm	43
3.3.4	Lookahead Algorithm	44
3.3.5	CEFT Algorithm	45
3.3.6	PEFT Algorithm	46
3.4	Experimental Results and Discussion	47
3.4.1	Randomly Generated Application Graphs	47
3.4.2	Real-world Application Graphs	53
3.4.2.1	Gaussian Elimination	54
3.4.2.2	Fast Fourier Transform	56
3.4.2.3	Montage Workflow	58
3.4.2.4	Epigenomics Workflow	61

3.5	A Possible Framework for Benchmarking	65
3.5.1	Advantages and Limitations of the Proposed Framework . . .	67
3.6	Summary	68
4	An Edge Priority-based Scheduling Algorithm	71
4.1	The EPS Algorithm	72
4.1.1	Edge Prioritization	74
4.1.2	Clustering	74
4.1.3	The Algorithm	75
4.1.4	Algorithm Complexity Analysis	76
4.1.5	An Illustrative Example	77
4.2	Performance Metrics	80
4.2.1	Normalized Schedule Length	80
4.2.2	Speedup	80
4.2.3	CCR	80
4.2.4	Percentage Improvement in NSL	81
4.2.5	Percentage Improvement in Speedup	81
4.3	Experimental Results and Discussion	81
4.3.1	Randomly Generated Task Graphs	82
4.3.2	Real-World Application Graphs	92
4.3.2.1	Gaussian Elimination	92
4.3.2.2	Fast Fourier Transform	103
4.4	Summary	112
5	Effective Critical Path Scheduling Algorithm	115
5.1	The ECP Algorithm	116
5.1.1	Critical Path Computation and Edge Selection	117
5.1.2	Clustering	118
5.1.3	The ECP Algorithm	119
5.1.4	Algorithm Complexity Analysis	120
5.1.4.1	Analysis of Algorithm 2	120
5.1.4.2	Analysis of Algorithm 3	121
5.1.4.3	Analysis of Algorithm 4	121
5.1.5	An Illustrative Example	121
5.2	Experimental Results and Discussion	124
5.2.1	Randomly Generated Task Graphs	124
5.2.2	Real-World Application Graphs	126
5.2.2.1	Gaussian Elimination	126
5.2.2.2	Fast Fourier Transform	131
5.2.2.3	Systolic Array	136
5.3	Summary	144

6	An Energy-aware Task Scheduling Algorithm	147
6.1	Related Work	148
6.2	System Model and Problem Formulation	149
6.2.1	Application Model	150
6.2.2	Processor Model	150
6.2.3	Energy Model	151
6.2.4	Problem Formulation	152
6.3	The EAEPS Algorithm	152
6.3.1	Priority Function for the Edges	153
6.3.2	Voltage Scaling for Non-critical Tasks	153
6.3.3	The Algorithm	154
6.4	Experimental Results	155
6.5	Summary	156
7	Conclusions and Future Directions	157
	Bibliography	161
	List of Publications	173