



**UNIVERSITI PUTRA MALAYSIA**

**LOAD-BALANCING MODELS FOR SCHEDULING DIVISIBLE  
LOAD ON LARGE SCALE DATA GRIDS**

**MONIR ABDULLAH ABDUH KAID**

**FSKTM 2009 8**

**LOAD-BALANCING MODELS FOR SCHEDULING DIVISIBLE  
LOAD ON LARGE SCALE DATA GRIDS**

**MONIR ABDULLAH ABDUH KAID**

**DOCTOR OF PHILOSOPHY  
UNIVERSITI PUTRA MALAYSIA**

**2009**



**LOAD-BALANCING MODELS FOR SCHEDULING DIVISIBLE LOAD ON  
LARGE SCALE DATA GRIDS**

By

**MONIR ABDULLAH ABDUH KAID**

**Thesis Submitted to the School of Graduate Studies, Universiti  
Putra Malaysia, in Fulfilment of the Requirements for  
Degree of Doctor of Philosophy**

**June 2009**



**TO MY PARENTS, MY WIFE, MY KIDS, AND TO ALL  
MY BROTHERS AND SISTERS**



Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment  
of the requirement for the degree of Doctor of Philosophy

**LOAD-BALANCING MODELS FOR SCHEDULING DIVISIBLE LOAD ON  
LARGE SCALE DATA GRIDS**

By

**MONIR ABDULLAH ABDUH KAID**

**June 2009**

**Chairman: Mohamed Othman, PhD**

**Faculty: Computer Science and Information Technology**

In many data grid applications, data can be decomposed into multiple independent sub datasets and distributed for parallel execution. This property has been successfully employed using Divisible Load Theory (DLT), which has been proven to be a powerful tool for modeling divisible load problems in large scale data grid. Load balancing in such environment plays a critical role in achieving high utilization of resources to schedule the applications efficiently through joint consideration of communication and computation time. There are some scheduling models, which have been studied, such as Constraint DLT (CDLT), Task Data Present (TDP) and Genetic Algorithm (GA). However, there has been no optimal solution reached. At the same time, effective schedulers are not only required to minimize the maximum completion time (*makespan*) of the jobs, but also the execution time of the schedulers.



This thesis proposes several load balancing models for scheduling divisible load on large scale data grids, when both processor and communication link speed are heterogeneous. The proposed models can be decomposed into three stages. The first stage is to develop new DLT based models for multiple sources scheduling. Closed form solutions for the load allocation are derived. The new models are called Adaptive DLT (ADLT) and A<sup>2</sup>DLT models. In the second stage, an Iterative DLT (IDLIT) model is proposed. Recursive numerical equations are derived to find the optimal workload assigned to the grid node. The closed form solutions are derived for the optimal load allocation. Although the IDLT model is proposed for single source, it has been applied in the case of multiple sources. The third stage integrates the proposed DLT based models with GA algorithm to solve the time consuming problem. In addition, the integration of the proposed DLT model with Simulated Annealing (SA) algorithm has been also developed.

The experimental results have proven that the proposed models yield better performance than previous models in terms of *makespan* and scheduler execution time. The ADLT and A<sup>2</sup>DLT models have reduced the *makespan* by 21% and 37% respectively compared to CDLT model. The IDLT model is capable of producing almost optimal solution for single source scheduling with low time complexity. In addition, the integration of the proposed DLT model with GA and SA algorithms has also significantly improved the performance. The SA is 64.70% better than GA in terms of *makespan*. Thus, the proposed models can balance the processing loads efficiently so that they can be integrated in the existing data grid schedulers to improve the performance.

Abstrak thesis yang diserahkan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

**MODEL KESEIMBANGAN MUATAN UNTUK MENJADUAL MUATAN  
YANG BOLEH DIBAHAGI PADA GRID DATA BERSKALA BASAR**

Oleh

**MONIR ABDULLAH ABDUH KAID**

**June 2009**

**Pengerusi: Mohamed Othman, PhD**

**Fakulti: Sains Komputer dan Teknologi Maklumat**

Dalam kebanyakan aplikasi grid data, data boleh diuraikan kepada pelbagai sub dataset bebas dan disebarkan untuk dilaksanakan secara selari. Ciri-ciri ini berjaya diguna-pakai melalui penggunaan Teori Muatan Mampu-Dibahagi (Divisible Load Theory - DLT) yang telah terbukti sebagai alat yang berkuasa untuk memodelkan masalah muatan yang boleh dibahagi pada grid data berskala besar. Keseimbangan muatan dalam persekitaran ini memainkan peranan yang kritikal bagi mencapai kekerapan penggunaan yang tinggi bagi sesuatu sumber untuk penjadualan aplikasi yang berkesan dengan mengambil kira komunikasi dan tempoh penaksiran. Terdapat beberapa model penjadualan seperti DLT Desakan (Constraint DLT - CDLT), Kehadiran Data Bertugas (Task Data Present - TDP), dan Algoritma Genetik (Genetic Algorithm - GA). Walaubagaimanapun, tiada penyelesaian berkesan dapat dicapai. Pada masa yang sama, penjadualan yang berkesan tidak hanya memerlukan pengu-



rangan masa penyelesaian maksima bagi sesuatu tugas, tetapi juga meminimakan masa pelaksanaan bagi penjadualan tersebut.

Kajian ini mencadangkan beberapa model muatan seimbang untuk menjadualkan muatan yang boleh dibahagi pada grid data berskala besar apabila kelajuan pemproses dan talian komunikasi adalah berlainan. Model yang dicadangkan dapat diuraikan kepada tiga tahap. Tahap pertama ialah mengembangkan model DLR baru untuk menjadualkan pelbagai sumber. Kaedah berdasarkan pendekatan tertutup bagi pembahagian beban telah dihasilkan. Pada tahap ini, dua model baru dikenali sebagai DLT Adaptasi (Adaptive DLT - ADLT) dan A<sup>2</sup>DLT dicadangkan. Pada tahap kedua, model DLT Iteratif (Iterative DLT - IDLT) telah dicadangkan. Persamaan ber-angka berulang-semula dihasilkan bagi mencari muatan tugas yang optima berhubung dengan nod grid. Kaedah berdasarkan pendekatan tertutup ini dihasilkan untuk mencari penempatan muatan tugas yang optima. Walaupun model IDLT ini dicadangkan untuk sumber tunggal, ia turut digunakan bagi kes-kes sumber yang pelbagai. Tahap ketiga pula menyatupadukan model DLT dan GA untuk menyelesaikan masalah pengiraan masa. Sehubungan itu, integrasi antara model DLT yang dicadangkan dengan algoritma Penguatan Di-simulasi (Simulated Annealing - SA) juga telah dibangunkan.

Keputusan eksperimen menunjukkan model yang dicadangkan telah memperoleh pencapaian yang lebih baik berbanding dengan model terdahulu dalam konteks masa penyelesaian maksima dan masa pelaksanaan oleh penjadual. Model ADLT dan



A<sup>2</sup>DLT telah mengurangi masa penyelesaian maksima masing-masing antara 21% dan 37% berbanding dengan model CDLT. Model IDLT pula berkeupayaan untuk menghasilkan penyelesaian paling baik bagi penjadualan sumber tunggal dengan kerumitan masa yang rendah. Selain itu, integrasi antara model DLT yang dicadangkan dengan GA dan algoritma SA telah meningkatkan prestasi secara signifikan. Algoritma SA adalah 64.70% lebih baik berbanding GA dari segi masa penyelesaian maksima. Oleh yang demikian, model yang dicadangkan mampu mengimbangi muatan pemprosesan dengan berkesan seterusnya dapat disatupadukan dalam penjadualan grid data yang sedia ada bagi memperbaiki prestasinya.

## ACKNOWLEDGMENTS

All Praise is due to Almighty ALLAH as He is all Mercifull, Most Gracious and Most Compassionate who gathered all knowledge for eternity. May ALLAH praise our Prophet Muhammad, and render him and his household safe and secure from all evil. All Grace and Thanks belongs to Almighty Allah.

Many special thanks and gratitude go to my supervisor Associate Professor Dr. Mohamed Othman, for his extensive help and valuable comments and advice at every stage of my research.

I would like to take this opportunity to express my sincere appreciation and thanks to the supervisory committee members, Associate Professor Dr. Hamidah Ibrahim and Dr. Shamala Subramaniam for their advices and comments during the completion of this thesis.

Thanks to every person who has supported me to produce my thesis. I am very grateful to the Faculty of Computer Science and Information Technology and the staff of Postgraduate office, Library and Universiti Putra Malaysia, for providing a studying and research environment.

Finally, many thanks to my parents, my wife, my kids, my brothers, my sister, all the family members and friends for their love, constant support and encouragement in all my endeavors.

## LIST OF TABLES

2.1	Comparison of Various Divisible Load Scheduling Models	35
3.1	Notations and Definitions	40
3.2	Experiment Parameters and their Range of Values	54
4.1	Processing Time of CDLT and ADLT Models (in Seconds)	75
4.2	<i>Makespan</i> Improvements of ADLT and A <sup>2</sup> DLT Models against CDLT Model	78
4.3	<i>Makespan</i> Improvements of ATDP and A <sup>2</sup> TDP Models against TDP Model	80
4.4	<i>Makespan</i> Improvements of ADLT and A <sup>2</sup> DLT Models against CDLT Model (with different <i>oiRatio</i> )	89
5.1	<i>Makespan</i> of the IDLT Model with Five Processing Nodes	104
5.2	Percentage of <i>Makespan</i> Improvements of IDLT against CDLT, ADLT and A <sup>2</sup> DLT Models for Single Source	113
6.1	Percentage of <i>Makespan</i> Improvements of AGA against GA Models	149
6.2	Execution Time vs. No. of Processing for GA and AGA Models	151
6.3	Percentage of <i>Makespan</i> Improvements of SA-based Algorithm against other Models (in Seconds)	154
6.4	Percentage of <i>Makespan</i> Improvements of SA against GA, AGA and IDLT Models for Multiple Sources ( <i>ccRatio</i> =1000)	159
6.5	Percentage of <i>Makespan</i> Improvements of SA-based Algorithm against GA, AGA and IDLT Models ( <i>oiRatio</i> =0)	163
6.6	Percentage of <i>Makespan</i> Improvements of SA-based Algorithm against GA, AGA and IDLT Models ( <i>oiRatio</i> >0.5)	163
6.7	Execution Time vs. No. of Sources for AGA and SA Models	165



# LIST OF FIGURES

1.1	Research Scope	8
2.1	Processing Loads Classification	17
2.2	Load Distribution and Computation in DLT [16]	22
2.3	Simple Structure of the GA	33
3.1	Grid Model [58]	41
3.2	Data Decomposition and their Processing	42
3.3	Communication and Computation of the Single Source within the System in Optimal Case	45
3.4	Communication and Computation of Multiple Sources Within the System in Optimal Case	46
3.5	The General Stages of the Research Framework	47
4.1	Timing Diagram of the Distribution Strategy with $M$ Processing Nodes in CDLT Model	58
4.2	The Framework of CDLT model	59
4.3	The Framework of TDP Model	61
4.4	The Framework of ADLT Model	62
4.5	Timing Diagram of the Distribution Strategy with $N$ Nodes and $M$ Data Files in ADLT Model	67
4.6	The Framework of ATDP Model	68
4.7	Framework of A <sup>2</sup> DLT Model	70
4.8	Framework of A <sup>2</sup> TDP Model	73
4.9	An Example of a System with One Data Source and Two Processing Nodes	74
4.10	<i>Makespan</i> of the CDLT, ADLT and A <sup>2</sup> DLT Models ( $M=100$ , $N=100$ )	77
4.11	<i>Makespan</i> of the TDP, ATDP and A <sup>2</sup> TDP Models ( $M=100$ , $N=100$ )	79
4.12	<i>Makespan</i> vs. Data File Size of CDLT, ADLT and A <sup>2</sup> DLT Models ( $N=100$ , $M=100$ and $ccRatio=0.001$ )	81
4.13	<i>Makespan</i> vs. Data file Size of CDLT, ADLT and A <sup>2</sup> DLT Models ( $N=100$ , $M=100$ and $ccRatio=1$ )	82
4.14	<i>Makespan</i> vs. Data file Size for CDLT, ADLT and A <sup>2</sup> DLT Models ( $N=100$ , $M=100$ and $ccRatio=1000$ )	83
4.15	<i>Makespan</i> vs. Data file Size of TDP, ATDP and A <sup>2</sup> TDP Models ( $N=100$ , $M=100$ and $ccRatio=0.001$ )	84



4.16	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT and A <sup>2</sup> DLT Models ( <i>ccRatio</i> =0.001)	85
4.17	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT and A <sup>2</sup> DLT Models ( <i>ccRatio</i> =1)	86
4.18	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT and A <sup>2</sup> DLT Models ( <i>ccRatio</i> =1000)	87
4.19	<i>Makespan</i> vs. No. of Nodes of TDP, ATDP and A <sup>2</sup> els ( <i>ccRatio</i> =0.001)	87
4.20	The Impact of Output Data Size to Input Data Size (a) <i>oiRatio</i> = 0 : No output or small size of output (b) <i>oiRatio</i> > 0.5	88
5.1	Proposed Framework of the Iterative Model	92
5.2	Data Decomposition and their Processing in Single Source	93
5.3	IDLT Framework for Single Source Scheduling	95
5.4	Simple Idea of IDLT Model for <i>M</i> Processing Nodes	99
5.5	Timing Diagram of the Distribution Strategy with Single Source and <i>N</i> Nodes (a) Initial Solution (b) Intermediate Solution (c) Final Solution	100
5.6	Data Decomposition and their Processing in Multiple Sources	101
5.7	Flowchart of IDLT Model for Multiple Sources	102
5.8	Comparison of Various Iterations in IDLT Model (a) Initial Step (b) and (c) Intermediate Steps (d) Final Step	105
5.9	<i>Makespan</i> vs. <i>ccRatio</i> of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =10)	106
5.10	<i>Makespan</i> vs. <i>ccRatio</i> of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =50)	107
5.11	<i>Makespan</i> vs. <i>ccRatio</i> of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =100)	107
5.12	<i>Makespan</i> vs. Data File Size of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>N</i> =100 and <i>ccRatio</i> =0.001)	108
5.13	<i>Makespan</i> vs. Data File Size of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>N</i> =100 and <i>ccRatio</i> =1)	109
5.14	<i>Makespan</i> vs. Data File Size for CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>N</i> =100 and <i>ccRatio</i> =1000)	109
5.15	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =50, <i>ccRatio</i> =0.001)	110
5.16	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =50, <i>ccRatio</i> =1)	111
5.17	<i>Makespan</i> vs. No. of Nodes of CDLT, ADLT, A <sup>2</sup> DLT and IDLT ( <i>M</i> =50, <i>ccRatio</i> =1000)	111
5.18	The Impact of Output Data Size to Input Data Size (a) <i>oiRatio</i> > 0.5 (b) <i>oiRatio</i> = 0 : No Output or Small Size of Output	112

5.19	<b>Convergence of IDLT Model for Single Source (<math>ccRatio=1000</math>)</b>	114
5.20	<i>Makespan vs. <math>ccRatio</math> of CDLT, ADLT, A<sup>2</sup>DLT and IDLT models (<math>N=100</math> and <math>M=100</math>)</i>	115
5.21	<i>Makespan vs. Data File Size for CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math>, <math>M=100</math> and <math>ccRatio=0.001</math>)</i>	117
5.22	<i>Makespan vs. Data File Size of CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math>, <math>M=100</math> and <math>ccRatio=1</math>)</i>	118
5.23	<i>Makespan vs. Data File Size of CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math>, <math>M=100</math> and <math>ccRatio=1000</math>)</i>	118
5.24	<i>Makespan vs. No. of Processing Nodes of CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math> and <math>ccRatio=0.001</math>)</i>	119
5.25	<i>Makespan vs. No. of Processing Nodes of CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math> and <math>ccRatio=1</math>)</i>	120
5.26	<i>Makespan vs. No. of Processing Nodes of CDLT, ADLT, A<sup>2</sup>DLT and IDLT (<math>N=100</math> and <math>ccRatio=1000</math>)</i>	120
5.27	<b>The Impact of Output Data Size to Input Data Size (a) <math>oiRatio = 0</math> : No Output or Small Size of Output (b) <math>oiRatio &gt; 0.5</math></b>	121
6.1	<b>Chromosome Representation</b>	126
6.2	<b>Proposed Framework of the AGA Model</b>	131
6.3	<b>Sub-gene Representation and Processing (a) in GA Model (b) in AGA Model</b>	133
6.4	<b>AGA Algorithm to the Problem</b>	135
6.5	<b>Simple Structure of the SA</b>	136
6.6	<b>Proposed Framework of the Integrated SA with IDLT Model</b>	140
6.7	<b>Sorted Vertice</b>	141
6.8	<b>Inversion Move Set</b>	142
6.9	<b>Inversion_Move_Set Procedure</b>	142
6.10	<b>SA-based for the Problem</b>	144
6.11	<b>Sorted Vertice</b>	145
6.12	<b>IDLT Model in SA Algorithm</b>	146
6.13	<i>Makespan vs. <math>ccRatio</math> of Different Models</i>	148
6.14	<i>Makespan of of the GA and AGA Models</i>	148
6.15	<b>Execution Time vs. No. of Sources of GA and AGA Models (<math>N=100</math>)</b>	150
6.16	<i>Makespan of IDLT and SA Models for Single Source</i>	153
6.17	<i>Makespan of GA, AGA, IDLT and SA Models</i>	154
6.18	<i>Makespan vs. No. of Nodes for GA, AGA, IDLT and SA Models (<math>ccRatio=0.001</math>)</i>	156
6.19	<i>Makespan vs. No. of Nodes of IDLT and SA Models (<math>ccRatio=0.001</math>)</i>	157

6.20 <i>Makespan vs. No. of Nodes of GA, AGA, IDLT and SA Models (<math>ccRatio=1</math>)</i>	157
6.21 <i>Makespan vs. No. of Nodes of GA, AGA, IDLT and SA Models (<math>ccRatio=1000</math>)</i>	158
6.22 <i>Makespan vs. No. of Nodes of IDLT and SA Models (<math>ccRatio=1000</math>)</i>	158
6.23 <i>Makespan vs. Data File Size of All Models (<math>ccRatio=0.001</math>)</i>	160
6.24 <i>Makespan vs. Data File Size of All Models (<math>ccRatio=1</math>)</i>	161
6.25 <i>Makespan vs. Data File Size of All Models (<math>ccRatio=1000</math>)</i>	161
6.26 <i>The Impact of Output Data Size to Input Data Size (<math>oiRatio = 0</math>)</i>	162
6.27 <i>The Impact of Output Data Size to Input Data Size (<math>oiRatio &gt; 0.5</math>)</i>	163
6.28 <i>Execution Time vs. No. of Sources of AGA and SA Models (<math>N=100</math>)</i>	165

# LIST OF ABBREVIATIONS

ADLT	Adaptive Divisible Load Theory
ATDP	Adaptive Task Data Present
CDLT	Constraint Divisible Load Theory
CERN	Conseil European pour la Recherche Nuclaire
CMS	Compact Muon Solenoid
DIANE	Distributed Analysis Environment
DLT	Divisible Load Theory
DS	Dataset Scheduler
EGEE	Enabling Grids for E-science
ES	External Scheduler
GA	Genetic Algorithm
HEP	High Energy Physics
IDLT	Iterative Divisible Load Theory
LAN	Local Area Network
LHC	Large Hadron Collider
LS	Local Scheduler
Mbps	Mega byte per second
NAREGI	National Research Grid Initiative
NP	None Polynomial
RADIS	Resource-Aware Dynamic Incremental Scheduling
SA	Simulated Annealing
TDP	Task Data Present
TSP	Traveling Salesperson Problem
VO	Virtual Organization
WAN	Wide Area Network



# TABLE OF CONTENTS

<b>DEDICATION</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>ABSTRAK</b>	<b>v</b>
<b>ACKNOWLEDGMENTS</b>	<b>viii</b>
<b>APPROVAL</b>	<b>ix</b>
<b>DECLARATION</b>	<b>xi</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF FIGURES</b>	<b>xiii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xvii</b>
<b>CHAPTER</b>	<b>xxi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Background	1
1.1.1 Data Grid Computing	2
1.1.2 Load Balancing in Data Grid	3
1.1.3 Divisible Load Theory	4
1.2 Problem Statement	4
1.3 Research Objectives	6
1.4 Research Scope	7
1.5 Research Significance	8
1.6 Thesis Organisation	9
<b>2 LITERATURE REVIEW</b>	<b>11</b>
2.1 Introduction	11
2.2 Data Grid	12
2.3 Scheduling in Data Grid	12
2.4 Resource Scheduling	14
2.4.1 Computation Resource	14
2.4.2 Communication Resource	15
2.5 Processing Loads Classification	16
2.5.1 Indivisible Load	17
2.5.2 Divisible Load	17
2.6 Divisible Load Theory	18
2.6.1 DLT Model Properties	19
2.6.2 DLT Scheduling Model	23
2.7 Related Works	26
2.7.1 Single Source Scheduling	27



2.7.2	Multiple Sources Scheduling	28
2.7.3	Genetic Algorithm based Model	33
2.8	Limitations of Related Works	35
2.9	Summary	37
<b>3</b>	<b>RESEARCH METHODOLOGY</b>	<b>38</b>
3.1	Introduction	38
3.2	General Description of the Scheduling Problem	39
3.2.1	Notations and Definitions	40
3.2.2	Problem Formulation	41
3.2.3	Cost Model	43
3.2.4	Optimality Criterion	44
3.3	Research Framework	47
3.3.1	Problem Formulation	48
3.3.2	Previous Models Implementation	48
3.3.3	Experiments	48
3.3.4	Analysis of Results	50
3.4	Suitability of DLT for the Problem	50
3.5	Suitability of GA and SA Algorithms for the Problem	51
3.6	The Experiment Environments	51
3.6.1	Computer Resources	52
3.6.2	Experimental Setup	52
3.7	Performance Metrics	54
3.7.1	Maximum Completion Time ( <i>Makespan</i> )	54
3.7.2	Scheduling Execution Time	55
3.8	Summary	56
<b>4</b>	<b>ADAPTIVE DIVISIBLE LOAD MODELS</b>	<b>57</b>
4.1	Introduction	57
4.2	Limitations of the Previous DLT Models	57
4.3	Proposed Adaptive DLT (ADLT) Model	62
4.3.1	Computation Time Fraction	63
4.3.2	Communication Time Fraction	64
4.3.3	The ADLT Closed Form Solution	66
4.3.4	Adaptive TDP (ATDP) Model	68
4.4	Proposed A <sup>2</sup> DLT Model	69
4.4.1	Computation Time Fraction	71
4.4.2	Communication Time Fraction	71
4.4.3	A <sup>2</sup> DLT Closed Form Solution	72
4.4.4	A <sup>2</sup> TDP Closed Form Solution	73

4.5	Results and Discussion	74
4.5.1	The Effect of Application Types	76
4.5.2	The Effect of Data File Sizes	81
4.5.3	Effect of Number of Nodes	84
4.5.4	The Effect of Ratio of Output Data to Input Data	88
4.6	Summary	90
<b>5</b>	<b>ITERATIVE DIVISIBLE LOAD MODELS</b>	<b>91</b>
5.1	Introduction	91
5.2	Limitations of the Adaptive DLT Models	92
5.3	Proposed IDLT Model for Single Source	93
5.3.1	Initial Solution	94
5.3.2	The Iterative Model	94
5.4	Proposed IDLT Model for Multiple Sources	100
5.4.1	The Multiple Sources Model	101
5.4.2	The IDLT Algorithm for Multiple Sources	101
5.5	Results and Discussion	103
5.5.1	IDLT for Single Source	103
5.5.2	IDLT for Multiple Sources	115
5.6	Summary	122
<b>6</b>	<b>INTEGRATED SCHEDULING MODELS</b>	<b>123</b>
6.1	Introduction	123
6.2	Limitations of the Previous DLT Models	124
6.3	Genetic Algorithm-Based Model	125
6.3.1	Chromosome Representation	125
6.3.2	Initial Population	126
6.3.3	Selection Methods	127
6.3.4	Crossover Operation	128
6.3.5	Mutation Operation	128
6.3.6	Stop Conditions	129
6.4	Proposed Adaptive GA Scheduling Model	130
6.4.1	New Gene Representation	132
6.4.2	Adaptive Initial Population	133
6.4.3	AGA to the Problem	135
6.5	Simulated Annealing (SA) Algorithm	135
6.5.1	Related Works	136
6.5.2	SA Parameters	137
6.6	Proposed SA based for the Problem	139
6.6.1	Initial Solution	141

	<b>xxi</b>
6.6.2 Move Sets in SA Algorithm	141
6.6.3 Application of SA to the Problem	143
6.6.4 IDLT Model in SA	146
6.7 Results and Discussion	147
6.7.1 AGA-based Scheduling Algorithm	147
6.7.2 SA-based Scheduling Algorithm	151
6.8 Summary	166
<b>7 CONCLUSIONS AND FUTURE WORKS</b>	<b>167</b>
7.1 Conclusion	167
7.2 Future Works	170
<b>REFERENCES</b>	<b>171</b>
<b>BIODATA OF THE STUDENT</b>	<b>177</b>
<b>LIST of PUBLICATIONS</b>	<b>178</b>



# CHAPTER 1

## INTRODUCTION

### 1.1 Background

Grid technology promises geographically distributed scientists to the access and sharing of physically distributed resources such as computing, communication and storage, and most importantly, data collections for large scale data intensive problems [1]. During the early stages, grid computing had research focused mainly on coordination of geographically distributed computing resources for high performance. However, in many areas of science and engineering such as the High-Energy Physics (HEP) and aerospace, requirements have emerged for collaborating and sharing huge amount of geographically distributed data as well as sharing high performance computing resources. For example, in Large Hadron Collider (LHC) experiment at Conseil Européen pour la Recherche Nucléaire (CERN), huge amounts of data in terabyte or petabyte scale are collected by observing particle collisions, and analyzed through different levels of data processing operations.

Because of the massive size and distributed nature of data in data grid, scheduling data grid applications must simultaneously consider communication and computation time to achieve high performance. One issue of the data grid system is the development of effective load balancing and scheduling techniques for the distribution of multiple processes on multiple processors. At the same time, these schedulers



should maintain the expected performance goals such as minimising execution time and communication delays, which will lead to maximization of resource utilisation. As the maximum completion time (*makespan*) minimization problem is known as NP-complete when the node and communication link speeds are considered [53, 65], this problem can only heuristically dealt with [66].

This thesis focuses on load scheduling and balancing on large scale data grid. Specifically, it focuses on issues associated with scheduling and the distribution of arbitrarily divisible loads among processors in such environment. This work is an extension of the existing DLT based model for scheduling divisible load in large scale data grid. The main objective in the research of the DLT based model is to determine the optimal fractions of the entire load to be assigned to each processor so that the maximum completion time of the entire load is at the lowest point possible.

### 1.1.1 Data Grid Computing

A classical grid is a collection of connected resources that are available for an application to perform a task. Nodes are often geographically distributed and heterogeneous in nature. Administrative domains are well connected through links of various performance and reliability.

A data grid is a system that deals with the control of sharing and managing large amounts of distributed data. Many scientific and engineering applications require