UNIVERSITI PUTRA MALAYSIA

BATCH MODE HEURISTIC APPROACHES FOR EFFICIENT TASK SCHEDULING IN GRID COMPUTING SYSTEM

JAMILU YAHAYA MAIPAN-UKU

FSKTM 2016 23

BATCH MODE HEURISTIC APPROACHES FOR EFFICIENT TASK SCHEDULING IN GRID COMPUTING SYSTEM

By

JAMILU YAHAYA MAIPAN-UKU

Thesis Submitted to the School of Graduate Studies,Universiti Putra Malaysia, in fulfilment of the Requirements for the Degree of Master of Science

June 2016

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

BATCH MODE HEURISTIC APPROACHES FOR EFFICIENT TASK SCHEDULING IN GRID COMPUTING SYSTEM

By

JAMILU YAHAYA MAIPAN-UKU

June 2016

Chairman : Abdullah Muhammed, PhD
Faculty : Computer Science and Information Technology

The concept of grid computing originated in the early 1990s as a metaphor for making computer power as easy as accessing an electric power grid. Grid computing appears to be a promising trend due to its ability to make the computational cost more cost-effective, the use of a given amount of computer resources, as a way to solve the problems that cannot be approached without an enormous amount of computing power, and its capability of utilising the resources of many computers which are not in use for other computational tasks. For desirable use (application) of the capabilities of large distributed systems like Grid, an efficient and effective scheduling algorithm is required for reducing total completion time and advancement of load balancing. Many algorithms have been implemented to solve the grid scheduling problem. These include Min-Min and Max-Min tasks scheduling algorithms, the former finds a task with minimum execution time and assigned to a resource that is able to produce it with minimum completion time, whereas the latter finds a task with maximum execution time and assigned to a resource that is abe to produce it with minimum completion time. Min-Min task scheduling algorithm has two clear weaknesses, a high value of makespan being generated and low resource utilisation when the numbers of tasks with minimum execution time are more than the number of tasks with maximum execution time. In Max-Min algorithm, a high completion time and resource imbalance are the two issues arise when the number of tasks with maximum execution time are more than the number of tasks with minimum execution time. This is due to the nature of Max-Min algoritm (the way how it works) in which it gives more priority to the task with maximum execution time first, leaving tasks with the minimum execution time waiting longer in a queue instead of executing them concurrently. To address these problems, this research proposes three new distributed static batch mode inspired algorithms. The first (proposed) algorithm is based on Min-Min, called Min-Diff, the second algorithm is based on Max-Min, called Max-Average, and the third algorithm is to handle the load balancing, called Efficient Load Balancing (ELB). In the Min-Diff algorithm, an Initial Task Queue (ITQ) (in non-decreasing order) is generated, where the differences between maximum and minimum execution time is calculated and compared with the minimum completion time. An appropriate resource for scheduling is selected accordingly. In the Max-Average algorithm, tasks are allocated to resources on the basis of Average Completion Time (AvCT). In ELB algorithm, the tasks are distributed among resources based on their execution time range. We simulate our proposed algorithms using a Java based simulator that is purposedly built for Grid computing simulations. The performances of the algorithms are evaluated using several metrics: makespan, average resource utilisation, flow-time, fitness, and load balancing. The results of our proposed algorithms has been compared with the ones that produced by the standard benchmark algorithms (MCT, MET, Min-Min and Max-Min). Experimental results demonstrate that the proposed algorithms (Min-Diff, Max-Average and ELB) are able to produce good quality solution when compared with the existing algorithm.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk Ijazah Master Sains

PENDEKATAN HEURISTIK MOD KELOMPOK UNTUK PENJADUALAN KERJA EFISIEN DALAM PERSEKITARAN PENGKOMPUTERAN GRID

Oleh

JAMILU YAHAYA MAIPAN-UKU

Jun 2016

Pengerusi : Abdullah Muhammed, PhD
Fakulti : Sains Komputer dan Teknologi Maklumat

Konsep pengkomputeran grid yang mula diperkenalkan sejak awal 1990-an adalah merupakan suatu perumpamaan untuk menjadikan kuasa komputer semudah mengakses suatu grid kuasa elektrik. Pengkomputeran grid dilihat sebagai satu trend yang menjanjikan kerana keupayaannya menjadikan kos pengkomputasi lebih kos-efektif, penggunaan sejumlah sumber komputer sebagai satu cara untuk menyelesaikan masalah yang tidak dapat diselesaikan tanpa sejumlah kuasa pengkomputeran besar dan kemampuannya untuk memanfaatkan sumber-sumber komputer yang pelbagai yang tidak digunakan sepenuhnya (dioptimumkan) untuk tugas pengkomputeran lain. Untuk sebarang kegunaan (aplikasi) yang memerlukan keupayaan sistem teragih besar seperti Grid, suatu algoritma penjadualan yang cekap dan berkesan adalah diperlukan untuk mengurangkan jumlah masa selesai dan kemajuan pengimbangan beban. Banyak algoritma telah diimplementasikan untuk menyelesaikan masalah penjadualan grid. Ini termasuklah algoritma penjadualan tugas Min-Min dan Max- Min, di mana algoritma pertama iaitu Min-Min, mencari satu tugas dengan masa pelaksanaan minimum untuk diumpukkan kepada satu sumber yang mampu menyempurnakannya dalam masa selesai yang minimum, manakala algoritma kedua, Max-Min, mencari satu tugas dengan masa pelaksanaan yang maksimum untuk diumpukkan kepada satu sumber yang dapat menyempurnakannya dalam masa selesai yang minimum. Algoritma penjadualan tugas Min-Min mempunyai dua kelemahan ketara iaitu penghasilan makespan yang tinggi dan penggunaan sumber yang rendah (di bawah kapasiti) apabila bilangan tugas dengan masa pelaksanaan minimum melebihi bilangan tugas dengan masa pelaksanaan maksimum. Dalam algoritma Max-Min, suatu masa selesai yang tinggi dan ketidakseimbangan sumber adalah merupakan dua isu yang timbul apabila bilangan tugas dengan masa pelaksanaan maksimum melebihi bilangan tugas dengan masa pelaksanaan minimum. Ini berlaku berikutan sifat algoritma Max-Min (cara ianya bekerja) memberi keutamaan kepada pengumpukkan tugas-tugas dengan masa pelaksanaan maksimum terlebih dahulu dan ini akan menyebabkan tugas-tugas dengan masa pelaksanaan minimum terpaksa menunggu lebih lama dalam suatu giliran dan tidak melaksanakan mereka secara serentak. Bagi

menangani masalah-masalah ini, kajian ini mencadangkan tiga algoritma baharu yang berinspirasikan mod kelompok statik teragih. Algoritma pertama (yang dicadangkan) adalah berdasarkan Min-Min, dipanggil Min-Diff, algoritma kedua adalah berdasarkan kepada Max-Min, dipanggil Max-Average, dan algoritma ketiga adalah untuk mengendalikan pengimbangan beban, dipanggil Efficient Load Balancing (ELB). Dalam algoritma Min-Diff, satu Giliran Tugas Awal (dalam urutan menaik) dihasilkan, di mana perbezaan di antara masa pelaksanaan maksimum dan minimum dikira dan dibandingkan dengan masa selesai yang minimum.

Sumber yang sesuai untuk penjadualan dipilih dengan sewajarnya. Dalam algoritma Max-Average, tugas-tugas yang diperuntukkan kepada sumber-sumber atas dasar Sederhana Waktu Penyelesaian (AvCT). Dalam algoritma ELB, tugas-tugas diedarkan di kalangan sumber berdasarkan julat masa pelaksanaan mereka. Kami menyimulasikan algoritma-algoritma yang dicadangkan dengan menggunakan penyimulasi yang berasaskan Java yang dibina khusus untuk simulasi pengkomputeran Grid. Prestasi algoritma dinilai menggunakan beberapa metrik: makespan, penggunaan sumber purata, aliran masa, kecergasan dan mengimbangi beban. Keputusan-keputusan eksperimen daripada algoritma yang dicadangkan dibandingkan dengan keputusan yang diperolehi daripada algoritma penanda aras standard (MCT, MET, Min-Min dan Max-Min). Keputusan eksperimen menunjukkan algoritma-algoritma yang dicadangkan (Min-Diff, Max-Average dan ELB) mampu menghasilkan penyelesaian yang berkualiti tinggi apabila dibandingkan dengan algoritma sedia ada.

ACKNOWLEDGEMENTS

All Praise is due to Almighty ALLAH as HE is all merciful, Most Gracious and Most Compassionate who gathered all knowledge of infinity. May ALLAH praises our beloved Prophet Muhammad (peace and blessing be upon Him), and renders him and his household safe and secure from all evil. All, Grace and Thanks belong to Almighty Allah.

I owe deep gratitude to the ones who have contributed greatly in completion of this dissertation.

Firstly, I would wish to convey my earnest gratitude to my supervisor, Dr. Abdullah Muhammed (one out of millions), for providing me with a platform to play on challenging areas of grid computing. His valuable comments and suggestions every stage of my research were encouraging. He provided me unflinching encouragement and support in numerous ways. I am indeed indebted to him more than he knows.

I am also grateful to my supervisory committee members, Dr. Azizol Abdullah and Dr. Masnida Hussin for giving encouragement and sharing their knowledge during the culmination of this dissertation.

I would like to thank administrative and technical staff members of the Department who have been kind enough to assist us in their respective offices, most especial Miss Sabarina (my supervisor secretary).

I would wish to thank all my friends and lab-mates for their encouragement and sympathy. They fixed my life beautiful and served me every-time when I was in some difficulty.

I would likewise desire to appreciate Ibrahim Badamasi Babangida University Lapai for giving the chance to advance my studies not forgetting my mentor, and brother in person of Prof. Muhammad Nasir Maiturare who makes the processes achievable.

Most importantly, none of this would have been possible without the love and patience of my household. My Father, Late Alh. Yahaya Maipan-uku, Mother, wife, kids, sisters, brothers and my entire family members to whom this thesis is dedicated to, has been a constant source of love, concern, support and strength all these years. I would wish to convey my earnest gratitude to them.

I certify that a Thesis Examination Committee has met on 10 June 2016 to conduct the final examination of Jamilu Yahaya Maipan-Uku on his thesis entitled "Batch Mode Heuristic Approaches for Efficient Task Scheduling in Grid Computing System" in accordance with the Universities and University Colleges Act 1971 and the Constitution of the Universiti Putra Malaysia [P.U.(A) 106] 15 March 1998. The Committee recommends that the student be awarded the Master of Science.

Members of the Thesis Examination Committee were as follows:

Shamala a/p K Subramaniam, PhD
Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Rohaya binti Latip, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Suhaidi Hassan, PhD
Professor
School of Computing, Information Technology Building
Universiti Utara Malaysia
(External Examiner)

ZULKARNAIN ZAINAL, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 26 July 2016

This thesis was submitted to the Senate of the Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

Abdullah bin Muhammed, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chaiman)

Azizol b Hj Abdullah, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Masnida binti Hussin, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

BUJANG BIN KIM HUAT, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work
- quotations, illustrations and citations have been duly referenced
- the thesis has not been submitted previously or comcurrently for any other degree at any institutions
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be owned from supervisor and deputy vice –chancellor (Research and innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software

Signature: ______________________ Date: ________________________

Name and Matric No: Jamilu Yahaya Maipan-Uku, GS41910

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) were adhered to.

Signature: ______________________________

Name of Chairman of Supervisory Committee: Dr. Abdullah bin Muhammed

Signature: ______________________________

Name of Member of Supervisory Committee: Dr. Azizol b Hj Abdullah

Signature: ______________________________

Name of Member of Supervisory Committee: Dr. Masnida binti Hussin

TABLE OF CONTENTS

		Page
ABSTRACT		i
ABSTRAK		iii
ACKNOWLEDGEMENT		v
APPROVAL		vi
DECLARATION		vii
LIST OF TABLES		xii
LIST OF FIGURES		iv
LIST OF ALGORITHMS		xv
LIST OF ABBREVIATIONS		xvi
CHAPTER		
1	**INTRODUCTION**	1
	1.1 Background	1
	1.2 Problem Statement	3
	1.3 Objectives of the Research	4
	1.4 Significance of the Study	4
	1.5 Scope of the Research	4
	1.6 Thesis Organisation	5
2	**LITERATURE REVIEW**	6
	2.1 Introduction	6
	2.2 Heterogeneity	8
	2.3 Grid Computing Architecture	8
	2.3.1 Grid fabric	9
	2.3.2 Core Grid middleware	9
	2.3.3 User-level Grid middleware	10
	2.3.4 Grid applications and portals	10
	2.4 Task Scheduling in Grid Computing Environment	10
	2.4.1 Immediate and Batch Mode Scheduling	11
	2.4.1.1 Dynamic Scheduling	12
	2.4.1.2 Static Scheduling	12
	2.5 Related Works	12
	2.5.1 Min-Min Scheduling Algorithm	13
	2.5.2 Max-Min Algorithm	18
	2.5.3 Min-Min and Max-Min Hybridisation	18
	2.5.4 Load Balancing Algorithm	18
	2.6 Summary	20
3	**RESEARCH METHODOLOGY**	**22**
	3.1 The research steps is outlined in the following points, depict in Figure 3-1:	22
	3.1.1 Requirement Analysis	22
	3.1.2 Research Design	24
	3.1.2.1 Tasks Set	24

3.1.2.2	Tasks Scheduler	24
3.1.2.3	Resource Monitor	25
3.1.2.4	Resources	25
3.1.3	Implementation	25
3.1.3.1	Input Parameters	26
3.1.3.2	Simulation Instances	28
3.1.4	Analysis/Evaluation	28
3.1.4.1	Evaluation Parameters	28
3.2	Dataset	31
3.2.1	Number of Nodes	31
3.2.2	Fraction of Resource	32
3.2.3	Number of Tasks	32
3.2.4	Fraction of Heavy Task and Light Task	32
3.2.5	Task Arrival Time	33
3.3	Summary	33
4	**BATCH MODE SCHEDULING ALGORITHMS**	34
4.1	Introduction	34
4.2	Immediate Mode Scheduling Algorithms	34
4.2.1	Minimum Execution Time (MET)	34
4.2.2	Minimum Execution Time (MCT)	35
4.3	Batch Mode Scheduling Algorithms	35
4.3.1	Min-Min Scheduling Algorithm	35
4.3.1.1	Proposed Algorithm (Min-Diff)	36
4.3.2	Max-Min Scheduling Algorithm	39
4.3.2.1	Proposed Algorithm (Max-Average)	39
4.3.3	Efficient Load Balancing Algorithm (ELBA)	41
4.4	Time Complexity of Proposed Algorithms (Min-Diff and Max-Average)	42
4.5	Summary	42
5	**RESULTS AND DISCUSSIONS**	43
5.1	Introduction	43
5.2	Performance Evaluation of Min-Diff	43
5.2.1	Makespan	44
5.2.2	Average Resource Utilisation	45
5.2.3	Conclusion	50
5.3	Performance Evaluation of Max-Average	50
5.3.1	Makespan	50
5.3.2	Average Resource Utilisation	52
5.3.3	Conclusion	56
5.4	Performance Evaluation of Efficient Load Balancing Algorithm	56
5.4.1	Load balancing	56
5.4.2	Flow Time	59
5.4.3	Fitness	60
5.4.4	Conclusion	62
5.5	Summary	62

6	**CONCLUSION AND FUTURE WORK**	63
	6.1 Conclusion	63
	6.2 Future Work	64
REFERENCES		65
BIODATA OF STUDENT		72
LIST OF PUBLICATIONS		73

LIST OF TABLES

Table		Page
2-1	Summary of centre Min-Min review	16
3-1	Parameters needed to be fed into the simulator	27
3-2	System Configuration	31
4-1	Expected Execution Time Table	37
4-2	Complexity of Batch mode Scheduling Algorithms	44
5-1	Makespan performance of different Min-Diff with algorithms for 512t x 16r	44
5-2	Makespan performance of Min-Diff with different algorithms for 1024t x 32r	45
5-3	Resource Utilisation of Min-Diff with different algorithms for 512t x 16r	46
5-4	Resource Utilisation of Min-Diff with different algorithms for 1024t x 32r	47
5-5	Makespan comparison of Min-Min & Min-Diff using 5r x 100t combinations	47
5-6	Makespan comparison of Min-Min & Min-Diff using 10r x 1000t combinations	48
5-7	Makespan comparison of Min-Min & Min-Diff using 10r x 1500t combinations	48
5-8	Makespan comparison of Min-Min & Min-Diff using 20r x 2000t combinations	48
5-9	Makespan comparison of Min-Min & Min-Diff using 25r x 2000t combinations	48
5-10	Average resource utilisation comparison of Min-Min & Min-Diff using 5r x 100t combinations	48
5-11	Average resource utilisation comparison of Min-Min & Min-Diff using 10r x 1000t combinations	49
5-12	Average resource utilisation comparison of Min-Min & Min-Diff using 10r x 1500t combinations	49

5-13 Average resource utilisation comparison of Min-Min & Min-Diff using 20r x 2000t combinations 49

5-14 Average resource utilisation comparison of Min-Min & Min-Diff using 25r x 2000t combinations 49

5-15 Positions of algorithms based on their performances 49

5-16 Makespan performance of Max-Average with different Algorithms for 512t x 16r 51

5-17 Makespan performance of Max-Average with different algorithms for 1024t x 32r 51

5-18 Average resource utilisation comparison of Max-Average with different heuristics for 512t x 16r 53

5-19 Average resource utilisation comparison of Max-Average with different Heuristics for 1024t x 32r 53

5-20 Makespan comparison of Max-Min & Max-Averageusing 5r x 100t 54

5-21 Makespan comparison of Max-Min & Min-Average using 10r x 1000t combinations 54

5-22 Makespan comparison of Max-Min & Max-Average using 10r x 1500t combinations 54

5-23 Makespan comparison of Max-Min & Max-Average using 20r x 2000t combinations 54

5-24 Makespan comparison of Max-Min & Max-Average using 25r x 2000t combinations 54

5-25 Average resource utilisation comparison of Max-Min & Max-Average using 5r x 100t combinations 55

5-26 Average resource utilisation comparison of Max-Min & Max-Average using 10r x 1000t combinations 55

5-27 Average resource utilisation comparison of Max-Min & Max-Average using 10r x 1500t combinations 55

5-28 Average resource utilisation comparison of Max-Min & Max-Average using 20r x 2000t combinations 55

5-29 Average resource utilisation comparison of Max-Min & Max-Average using 25r x 2000t combinations 55

5-30 Comparison of load balancing by different algorithms for 512t x 16r 56

5-31 Comparison of load balance by different algorithms for 1024t x 32r 57

5-32 Comparison of load balancing by different algorithms using 5r x 100t combination 58

5-33 Comparison of load balancing by different algorithms using 10r x 1000t combination 58

5-34 Comparison of load balancing by different algorithms using 10r x 1500t combination 58

5-35 Comparison of load balancing by different algorithms using 20r x 2000t combination 58

5-36 Comparison of load balancing by different algorithms using 25r x 2000t combination 58

5-37 Comparison of flow-time by different algorithms for 512t x 16r combination 59

5-38 Comparison of flow-time by different algorithms for 1024t x 32r combination 60

5-39 Comparison of fitness by different algorithms for 512t x 16r combination 61

5-40 Comparison of fitness by different algorithms for 1024t x 32r combination 61

LIST OF FIGURES

Figure		Page
1-1	Evolution of Gird	2
2-1	Typical views of Grid Environment	7
2-2	Simple grid settings with three virtual organisations	7
2-3	A layered Grid architecture and components	9
2-4	Three Phases of Grid Scheduling	10
2-5	Batch Mode Scheduling Methods	11
3-1	Research Methodology	23
3-2	Layered Simulator Architecture	24
3-3	Decision for choosing an algorithm	25
3-4	Simulation Steps	26
4-1	Pseudo Code of Min-Min Algorithm	36
4-2	Pseudo code of the proposed algorithm (Min-Diff)	37
4-3	Pseudo code of Max-Min Scheduling Algorithm	39
4-4	Pseudo code of the proposed algorithm (Max-Average)	40
4-5	Pseudo code of Efficient Load balancing (ELBA)	41
5-1	Makespan Comparison of Min-Diff and Min-Min 512 x 16	44
5-2	Makespan Comparison of Min-Diff & Min-Min 1024 x 32	45
5-3	Avgru Comparison of Min-Diff and Min-Min 512 x 16	46
5-4	Avgru Comparison for four instances using 1024 x 32	47
5-5	Makespan Comparison of Max-Min and Max-Average for 512 x 16	51

5-6 Makespan Comparison of Max-Min and Max-Average for 1024 x 32 52

5-7 Avgru of Max-Min and Max-Average using 512 x 16 52

5-8 Avgru of Max-Min and Max-Average using 1032 by 32 53

5-9 Load Balancing Chart for 512 x 16 Combinations 57

5-10 Load Balancing Chart for 1024 x 32 Combinations 57

5-11 Flow-time Chart for 512 x 16 Combinations 59

5-12 Flow-time Chart for 1024 x 32 Combinations 60

5-13 Fitness Chart for 512 x 16 Combinations 61

5-14 Fitness Chart for 1024 x 32 Combinations 62

LIST OF TASK SCHEDULING ALGORITHMS

Algorithm		Page
1.	Minimum Execution Time (MET)	36
2.	Minimum Completion Time (MCT)	37
3.	Min-Min Scheduling Algorithm	37
4.	Max-Min Scheduling Algorithm	41
5.	Max-Average Scheduling Algorithm	41
6.	Efficient Load Balancing Algorithm	43

LIST OF ABBREVIATIONS

X_{min}	Minimum completion time
X_{max}	Maximum completion time
MinDiff	Differences between X_{max} and X_{min}
MinECT	Minimum Execution Completion Time
Ti	Meta-task Id of meta-task i
Rj	Resource Id of resource j
Ci,j	Completion time for meta-task i on resource j
Xi,j	Execution time for meta-task i on resource j
Rj	Ready time of j
RU	Resource Utilisation
MT	Meta-Tasks
Avgru	Average resource utilisation
RT	Resource Ready Time
ET	Task Execution Time
MET	Minimum Execution Time
MCT	Minimum Completion Time
α	range of completion time
MIPS	Million Instruction Per Second
VO	Virtual Organisation
ETC	Expected Time to Compute
HiHi	Heavy tasks along with high capacity resources
HiLo	Heavy tasks along with low capacity resources
LoHi	Light tasks along with high capacity resources
LoLo	Light tasks along with low capacity resources
EET	Expected Execution Timetable
ECT	Expected Completion Timetable
MinET	Minimum execution time
MaxET	Maximum execution time

CHAPTER 1

INTRODUCTION

1.1 Background

The present situation with the computational structures is, in a few viewpoints, closely resembling that of the power systems (electricity) toward the origination of the twentieth century. Just about then, the era of power has been feasible, even at the same time it was important to have accessible generators of power. The reason that allowed its foundation, was the disclosure of new innovations to be specific about the systems of transmission and broadcast of power (electricity). These surprises made it possible to have a stable and low value organisation. In this approach, the power turned out to be all around available. By similarity, the grid is embraced in designated a computational base of circulating resources, exceedingly heterogeneous, connected by heterogeneous networks and a middleware that provides solid, basic, straightforward, effective and worldwide access to their computational capabilities.

In the few decades, grid computing grew rapidly over a brief period of time, pushed by essential innovation headways and enthusiasm of huge Information Technology (IT) organizations, for example, International Business Machines (IBM), Sun Microsystems, Oracle and Hewlett-Packard (HP). The ideas that lead to the basis of today's grid computing can be traced back to late 1980s, where the first concept was developed by researchers of distributed super-computing for optimisation with prominence on scheduling algorithms to accomplish high-performance computing (e.g. Condor-G). In the late 1990s, the terms of computational grids and grid computing were advanced by (Foster et al., 2003) who developed the Globus toolbox as a general middleware for Grid Systems. From that point onwards, Grid Computing, systems, and technology are propelling in a relentless way. Figure 1-1 shows the evolution chart of grid computing.

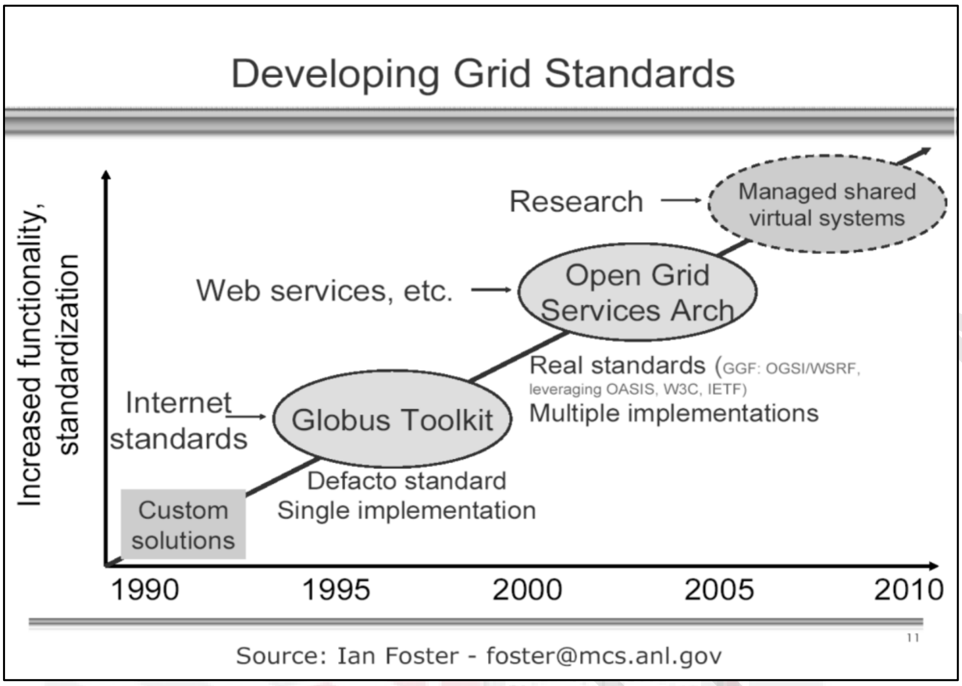

Figure 1-1 Evolution of Grid Computing (Foster, 2003)

As many applications need access to specialised machines, data or compute power (Plaszczak, 2006), a technology that enables resource virtualisation, on-demand provisioning, and resource sharing between organisations like Grid is becoming important. Grid computing is the ability of utilising a lot of open standards and protocols, to acquire access to applications and data, processing power, storage capacity and a huge array of other computing resources over the Internet. A Grid is a type of parallel and distributed organisation that enables the sharing, selection, and accumulation of resources spread across multiple administrative areas based on their resource availability, capability, performance, cost, and a user's quality of service required (Plaszczak, 2006). All the same, the major problem any Grid-like computational system is scheduling (Xhafa & Abraham, 2008). Grid task scheduling according to (Alherbi & Sharma., 2012) is an integrated part of computing, which effectively utilises the unused time of resources. It is responsible for mapping jobs, tasks to grid resources under various measures and grid environment configuration.

In a Grid computing system, mapping is necessary schemes adopted in assigning tasks to machines and the order of execution of tasks assigned to each machine for effective utilisation of resources and reduction of the overall completion time (Maheswaran et al., 1999). Mapping heuristics can be grouped into two categories, immediate/on-line mode and batch mode heuristics. In the immediate mode / on-line mode, a task is mapped onto a machine as soon as it arrives at the system. Examples of immediate/online mode heuristics include; Minimum Completion Time (MCT), Minimum Execution Time (MET), Switching algorithm (SA), K-percent best (Kpb) and Opportunistic Load Balancing (OLB). While in the batch mode, tasks are not mapped onto the machines as they arrive; instead, they are collected into a set that is examined for mapping at prescheduled times called mapping events. Example of batch mode heuristics includes; Min-Min heuristic, Max-Min heuristic, and Suffrage heuristic (Soheil et al., 2013).

1.2 Problem Statement

Task scheduling algorithm is considered as a significant subject in the present grid concept. The need for active scheduling surges to achieve better performance in computing. Usually, it is hard to discover an ideal resource distributor that minimises the schedule times of jobs and efficiently consumes the resources. The three key stages of grid scheduling remain resource discovery, gathering resource information, and job execution. The selection of the best couples of jobs and resources in the next phase has remained an NP-complete problem (Vijayalakshmi & Vasudevan, 2015).

The grid task scheduling algorithm is responsible for allocating jobs/tasks to Grid resources in accordance with various standards and Grid environment configurations. In a computational grid, task scheduling problem is enhanced by minimising makespan and maximising system utilisation; distribute the loads among the resources as evenly as possible and fulfils economic system demand/user constraints.

Many researchers still remain interested in carrying out research in scheduling algorithms for heterogeneous grid computing environment despite being studied for years, this is due to their ability to produce good quality solutions. These algorithms include: Min-Min and Max-Min tasks scheduling algorithm.

Min-Min algorithm finds the task with minimum execution time and assigns to resource that produces minimum completion time for the task. Allocating tasks in this manner gives a better performance when number of larger tasks exceeded the lighter tasks, but it major problem is that, it causes high completion time, poor resource utilisation, and load imbalance when the lighter number of tasks exceeds the larger tasks.

Furthermore, Max-Min algorithm works the same way with Min-Min, but the tasks with maximum execution time will be executed first. The problem here is that, allocating tasks in this way causes poor resource utilisation and high completion time when the number of larger tasks exceeds the lighter tasks.

It is obvious that, tasks selection and distributing them in an appropriate manner is a key challenge in grid scheduling system and it can be implemented using batch mode heuristics. Therefore, a substantial enhancement in the computational efficiency of the algorithms is something interesting to be investigated.

1.3 Objectives of the Research

The aim of this study is to propose new task scheduling algorithms that are able to produce good quality solutions when compared with the standard Min-Min and Max-Min scheduling algorithms in solving grid scheduling problem. The objectives are as follows:

a. to propose a new task scheduling algorithm (Min-Diff) that is able to find an optimal solution for different grid task scheduling problems.
b. to propose a new task scheduling algorithm (Max-Average) that is able to minimise completion time and maximises resource utilisation rate.
c. to propose an efficient load balancing algorithm that can distribute tasks to the grid computing resources as evenly as possible.

1.4 Significance of the Study

This study aims to minimise the overall completion time (makespan), enhance the resource utilisation and load balancing that are needed for efficient scheduling in grid computing environment.

1.5 Scope of the Research

This study only concentrates on static scheduling algorithms of batch mode for mapping independent tasks in a grid computing environment.

1.6 Thesis Organisation

The rest of the chapters are organised as follows;

In **Chapter 2**, we present a scientific literature of grid scheduling system. The general concept of the grid, and batch mode heuristics approaches.

In **Chapter 3,** we discuss the research steps, implementation, and evaluation of input and output parameters for the research.

In **Chapter 4,** we discuss the existing batch mode algorithms (Min-Min and Max-Min) and our new proposed tasks scheduling algorithms (Min-Diff, Max-Average and Efficient Load Balancing).

In **Chapter 5,** we present the simulation results and discussion. The results for the comparison of our proposed methods with the standard benchmark algorithms are also displayed.

In **Chapter 6,** we discuss the Research contributions, findings, future works and a summary of the research.

REFERENCES

Alharbi, F, (2012). Simple Scheduling Algorithm with Load Balancing for Grid Computing. *Asian Transactions on Computers* (ATC ISSN: 2221-4275). Vol. 2 (8 - 15), pp.8-9.

Amalarethinam, G.D.I. & Kfatheen V.S., (2014). Max-min Average Algorithm for Scheduling Tasks in Grid Computing Systems. International Journal of Computer Science and Information Technologies. Vol. 3, pp. 3659-62.

Amalarethinam, D. G., & Mary, G. J. (2011). A new Directed acyclic graph (DAG) based Dynamic Task Scheduling Algorithm (DYTSA) for Multiprocessor Systems. International Journal of Computer Applications. Vol. 19(8), pp. 24-28.

Anand, K.C. & Rajendra, S., (2011). New Heuristic for Scheduling of Independent Tasks in Computational Grid. *International Journal of Grid and Distributed Computing*. Vol. 4(3), pp. 25-27.

Anousha, S., Anousha, S., & Ahmadi, M. (2014). A New Heuristic Algorithm for Improving Total Completion Time in Grid Computing. In Multimedia and Ubiquitous Engineering, Springer Berlin Heidelberg, pp. 17-26.

Bansal, S., & Hota, C. (2011). Efficient Algorithm on Heterogeneous Computing System. International Conference on Recent Trends in Information Systems, (78-1-4577-0792-6/11), pp. 57-61.

Bardsiri, A. K., & Hashemi, S. M. (2012). A Comparative Study on Seven Static Mapping Heuristics for Grid Scheduling Problem. International Journal of Software Engineering and Its Applications. Vol. 6(4), pp. 247-256.

Bote-Lorenzo, M. L., Dimitriadis, Y. A., & Gómez-Sánchez, E. (2004). Grid characteristics and uses: A grid definition. In Grid Computing, Springer Berlin Heidelberg, pp. 291-298.

Braun, T. D., Siegel, H. J. & Beck, N., (2001). A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing. Vol. 61, pp. 823 – 831.

Buyya, R., & Venugopal, S. (2005). A gentle introduction to grid computing and technologies. Computer Society of India (CSI). Vol. 29(1), pp. 9-19.

Cao, L., Liu, X., Wang, H., & Zhang, Z. (2014). OPT-Min-Min Scheduling Algorithm of Grid Resources. Journal of Software. Vol. 9(7), pp. 1868-1875.

Cao, J., Spooner, D., Jarvis, S., & Nudd, G. (2005). Grid load balancing using intelligent agents. Future Generation Computer Systems, Vol. 21(1), 135-149.

Chaturvedi, A.K, & Sahu, R., (2011). New Heuristic for Scheduling of Independent Tasks in Computational Grid. International Journal of Grid and Distributed Computing. Vol. 4(3), pp. 25-27.

Chen, H., Wang, F., Helian, N. & Akanmu, G., (2013). User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing. *University of Kent Canterbury United Kingdom*, pp. 1-6.

Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., & Tuecke, S. (2000). The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. Journal of network and computer applications. Vol. 23(3), pp. 187-200.

Devipriya, S., & Ramesh, C. (2013). Improved Max-Min Heuristic Model for Task Scheduling in Cloud. *Institue of Electrical and Electronics Engineers (IEEE)*, pp. 2.

Dong, F. & Selim, G. A. (2006). Scheduling Algorithms for Grid Computing: State of the Art and Open Problems. School of Computing, Queen's University Kingston, Ontario. Technical Report No. 2006-504.

Doreen, D., Miriam, H., & Easwarakumar, K. (2010). A Double Min Min Algorithm for Task Metascheduler on Hypercubic Peer-to-Peer (P2P) Grid Systems. International Journal of Computer Science Issues. Vol. 7(4).

Elzeki, O.M., Reshad, M.Z. & Elsoud, M.A., (2012). Improved Max-Min Algorithm in Cloud Computing. International Journal of Computer Applications. Vol. 50, pp. 22 -25.

Etminani, K., Naghibzadeh, M., & Yanehsari, N.R., (2008). A Hybrid Min-Min Max-Min Algorithm with Improved Performance. *Department of Computer Engineering, Ferdowsi University of Mashad, Iran.* Vol. 32, pp.1 – 3.

Farhad, S.G., Ahadi, M., Maleki, I., Habibpour, R and Kamalinia, A. (2013). Analysis of Scheduling Algorithms in Grid Computing Environment. International Journal of Innovation and Applied Studies. Vol. 4(3), pp. 560 – 64.

Foster, I., & Kesselman, C. (1998). Computational grids. Cern European Organization for Nuclear Research-Reports-Cern, pp. 87-114.

Foster, I., & Kesselman, C. (Eds.). (2003). The Grid 2: Blueprint for a new computing infrastructure. Elsevier, pp. 30-48.

Freund R.F, Gherrity M, Ambrosius S, Campbell M, Halderman M, & Hensgen D., (1998). Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. Heterogneous Computing Workshop (HCW) '98, Orlando, pp. 184–99.

Fujimoto, N., & Hagihara, K. (2004). A Comparison among Grid Scheduling Algorithms for Independent Coarse-Grained Tasks. Vol. 2(4), pp. 7.

Ghosh, T., Goswami, R., Bera, S., & Barman, S. (2012). Load Balanced Static Grid Scheduling Using Max-Min Heuristic. *The 2nd Institute of Electronical and Electronic Engineers (IEEE) International Conference on Parallel, Distributed and Grid Computing*, pp. 419-423.

Gupta, K., & Singh, M., (2012). Heuristic Based Task Scheduling In Grid. *International Journal of Engineering and Technology (IJET)*. Vol. 4, pp.254 – 258.

Hak Du, K. & Jin Suk, K., (2001). An On-line Scheduling Algorithm for Grid Computing Systems. Electronics and Telecommunications Research Institute, Taejon, Korea & School of Computer Science, University of Seoul, Seoul, Korea. Vol. 32, pp.2 – 6.

Hao, Y., & Liu, G. (2015). Evaluation of nine heuristic algorithms with data-intensive jobs and computing-intensive jobs in a dynamic environment. Software, Institute of Engineering and Technology (IET), Vol. 9(1), pp. 7-16.

He. X, X-He Sun, & Laszewski. G.V., (2003). QoS Guided Min- Min Heuristic for Grid Task Scheduling. Journal of Computer Science and Technology. Vol. 18, pp. 442-451.

Hemamalini, M., & Srinath, M. V. (2015). Memory Constrained Load Shared Minimum Execution Time Grid Task Scheduling Algorithm in a Heterogeneous Environment. Indian Journal of Science and Technology, Vol. 8(15).

Hemamalini, M., (2012). Review of Grid Task Scheduling in Distributed Heterogeneous Environment. International Journal of Computer Applications. Vol. 40, pp. 24 – 26.

Hesam, I., Ajith, A., & VS., (2009). Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environment. *International Joint Conference on Computational Sciences and Optimization*, pp. 8-10.

Ibarra, O. H., & Kim, C. E. (1977). Heuristic algorithms for scheduling independent tasks on nonidentical processors. Journal of the ACM (JACM). Vol. 24(2), pp. 280-289.

Joshy, J., & Fellenstein, C. (2004). Grid Computing. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference. Chapter 1, pp. 5.

Kamalam, G. K., & Bhaskaran, V. M. (2010). An improved Min-Mean heuristic scheduling algorithm for mapping independent tasks on heterogeneous computing environment. Journal of Computational cognition. Vol. 8(4), pp 1-5.

Kaminsky, A. Building Parallel Programs: SMPs, Clusters, and Java. Cengage Course Technology (2010). ISBN 1-4239-0198-3.

Kaminsky, A. (2013). BIG CPU, BIG DATA: Solving the World's Toughest Computational Problems with Parallel Computing. Creative Commons.

Kartal, E., Tabak, B. B., C., & Cevdet, A., (2014). Improving the Performance of Independent Task Assignment Heuristics MinMin, MaxMin and Sufferage. Institute of Electronical and Electronics Engineers (IEEE) Transactions on Parallel and Distributed Systems. Vol. 25, pp.1248 – 1249.

Kaur, R., & Patra, P. (2013). Resource Allocation with improved MinMin Algorithm. International Journal of Computer Applications, Vol. 76(15), pp. 61-67.

Kaur, D., & Singh, S. (2014). An Improved min-min Algorithm for Job Scheduling using Ant Colony Optimization. International Journal of Computer Science and Mobile Computing. Vol. 3(5), pp. 552-556.

Kfatheen, S.V., Banu, M.N., & Selvi, S.K., (2014). TAAG: An Efficient Task Allocation Algorithm for Grid Environment. International Journal of Engineering and Technology (IJET). Vol. 4, pp. 1961-1968.

Kfatheen, S.V., Banu, M.N., & Selvi, S.K., (2014). TLLB: Two-Level Load Balanced Algorithm for Static Meta-Task Scheduling in Grid Computing. International Journal of Computer Applications. Vol. 105, pp.38 – 41.

Kim, H., & Kim, J. 2004. An online scheduling algorithm for grid computing systems, Grid and Cooperative Computing, pp. 34-39.

Kokilavani, T. & Amalarethinam, D.I.G., (2011). Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing. International Journal of Computer Applications. Vol. 20(2), pp. 43-47.

Kumar, A. (2003). Understanding the Grid Computing. The IIIT A e-magazine. Vol. 1(6).

Li, W., & Zhang, W. (2009). An improved Scheduling Algorithm for Grid Tasks. *International Symposium on Intelligent Ubiquitous Computing and Education.* Vol. 35, pp. 9-12.

Li, Y., & Lan, Z. (2004). A survey of load balancing in grid computing. In Computational and Information science. Springer Berlin Heidelberg, pp. 280-285.

Liu, K., Chen, J., Jin, H., & Yang, Y. (2009). A Min-Min Average Algorithm for Scheduling Transaction-Intensive Grid Workflows. New Zealand. Conferences in Research and Practice in Information Technology. Vol. 99, pp. 1-8.

Liu, L., & Li, G. (2010). An Improved MIN-MIN Grid Tasks Scheduling Algorithm Based on QoS Constraints. *International Conference on Optics, Photonics and Energy Engineering*. Vol. 10, pp. 281-283.

Livny, M., & Melman, M. (1982, April). Load balancing in homogeneous broadcast distributed systems. In ACM Sigmetrics Performance Evaluation Review. Vol. 11(1), pp. 47-55).

Luo, P., & Shi, Z. (2007). A revisit of fast greedy heuristics for mapping a class of independent tasks onto heterogeneous computing systems (K. Lü, Ed.). Journal of Parallel and Distributed Computing. Vol. 67, pp. 695-714.

Ma, T., Yan, Q., Liu, W., Guan, D., & Lee, S. (2011). Grid task scheduling: algorithm review. Institution of Electronics and Telecommunication Engineers (IETE) Technical Review. Vol. 28(2), pp. 158-167.

Maheswaran, M., Ali, S., Siegel, H. J., Hensgen, D., & Freund, F. R., (1999). Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems1. *Journal of Parallel and Distributed Computing*. Vol. 59, pp. 107 – 118.

Mao, Y., Chen, X., & Li, X. (2014). Max–Min Task Scheduling Algorithm for Load Balance in Cloud Computing. Proceedings of International Conference on Computer Science and Information Technology, *Advances in Intelligent Systems and Computing*. Vol. 255, pp. 457-465.

Meraji, A, & Salehnamadi, M.R., (2013). A Batch Mode Scheduling Algorithm for Grid Computing. Journal of Basic and Applied Scientific Research. Vol. 3 (4), pp. 173-181.

Min-You Wu, M.U., Shu, W. & Zhang, H., (2000). A Static Mapping Algorithm for Meta-tasks on Heterogeneous Computing Systems. Institute of Electrical and Electronic Engineers (IEEE). Vol. 1, pp. 1 – 6.

Minal, S., Rajesh, B., & Rupesh, M., (2013). Task Assignment in Heterogeneous Environment with Advanced Scheduling Algorithms. International Journal of Advanced Research in Computer Science and Software Engineering. Vol. 3, pp. 1079 – 1082.

Ming, G., & Li, H. (2011). An Improved Algorithm Based on Max-Min for Cloud Task Scheduling. Recent Advances in Computer Science and Information Engineering Lecture Notes in Electrical Engineering. Vol. 125, pp. 217-223.

Moallem, A. (2009). Using swarm intelligence for distributed job scheduling on the grid. Doctoral dissertation, University of Saskatchewan Saskatoon, pp. 11-14.

Mukherjee, A. (2011). An Efficient Job-Grouping Based Scheduling Algorithm for Fine-Grained Jobs in Computational Grids (Doctoral dissertation).

Nagda H., & Visariya, D. (2012). Task Scheduling in Grid Computing. Team Coda, Department of Computer Science, Rochester Institute of Technology (RIT).

Panda, S.K., Agrawal, P., & Mohapatra, D.P., (2013). X-DualMake: Novel Immediate Mode Scheduling Heuristics in Computational Grids. Department of Computer Science and engineering Indian School of Mines Dhanbad, India, International Business Machine (IBM) Bangalore, India, National Institute of Technology Rourkela, India, pp. 1-2 .

Panda, S. K., Agrawal, P., Khilar, P. M., & Mohapatra, D. P. (2014). Skewness-Based Min-Min Max-Min Heuristic for Grid Task Scheduling. Fourth International Conference on Advanced Computing & Communication Technologies, IEEE, pp. 1-8.

Parsa, S & Reza, E. M., (2009). RASA: A New Task Scheduling Algorithm in Grid Environment. World Applied Sciences Journal 7 (Special Issue of Computer & Information Technology), pp.152-155.

Pinel, F., Dorronsoro, B., Pecero, J.E., Bouvry, P., & Khan, S.U. (2012). A two-phase heuristic for the energy-efficient scheduling of independent tasks on computational grids. Springer Cluster Computer. Vol. 16, pp. 421-433.

Plaszczak, P., & Wellner, R. (2006). In Grid computing the savvy manager's guide. Amsterdam: Elsevier/Morgan Kaufmann. Vol. 1, pp. 5.

Pramit Karmakar: http://www.slideshare.net/pramit9836/grid-computing-32947751.

Rajkumar Buyya & Srikumar Venugopal (2005). A gentle introduction to grid computing and technologies. Computer Society of India (CSI) Communications. Vol. 29(1), pp. 9.

Reda, N. M., Tawfik, A., Marzok, M. A., & Khamis, S. M. (2014). Sort-Mid tasks scheduling algorithm in grid computing. Journal of Advanced Research.

Sharma, M.G., & Bansal, P., (2012). Min-Min Approach for Scheduling in Grid Environment. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*. Vol. 1, pp. 26.

Shivaratri, N. G., Krueger, P., & Singhal, M. (1992). Load distributing for locally distributed systems. Computer. Vol. 25(12), pp. 33-44.

Singh. M & Suri. P.K, QPS., (2008). A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid. Information Technology Journal. Vol. 7, pp. 1176-80.

Soheil, A., & Mahmoud, A., (2013). An Improved Min-Min Task Scheduling Algorithm in Grid Computing. *Springer-Verlag Berlin Heidelberg*. Vol. 32, pp.103 – 106.

Tsai, M. Y., Chiang, P. F., Chang, Y. J., & Wang, W. J. (2011). Heuristic scheduling strategies for linear-dependent and independent jobs on heterogeneous grids. In Grid and Distributed Computing, Springer Berlin Heidelberg. CCIS 261, pp. 496-505.

Tseng, L-Y., Chin, Y-H., & Wang, S-C., (2009). A minimized makespan scheduler with multiple factors for Grid computing systems. Expert Systems with Applications journal. Vol. 36(8), pp. 11118-11130.

Tuecke, S, Kesselman, C., & Foster, I. (2003). The anatomy of the grid. pp. 171-197.

Vijayalakshmi, R., & Vasudevan, V. (2015). Static Batch Mode Heuristic Algorithm for Mapping Independent Tasks in Computational Grid. Journal of Computer Science, Vol. 11(1), pp. 224.

Wenzheng, L., & Wenyue, Z. (2009, May). An improved scheduling algorithm for grid tasks. In 2009 International Symposium on Intelligent Ubiquitous Computing and Education, IEEE, pp. 9-12.

Wu, M. Y., Shu, W., & Zhang, H. (2000). Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. Proceeding(9^{th}), Institute of Electrical and Electronic Engineers (IEEE), pp. 375 – 385.

www.slideshare.net/pramit9836/grid-computing-32947751.

Xhafa, F., Barolli, L., & Durresi, A. (2007). Batch mode scheduling in grid systems. *International Journal of Web and Grid Services*. Vol. 3(1), pp. 19-19.

Xhafa, F., & Abraham, A. (2008). Meta-heuristics for grid scheduling problems. *Springer Berlin Heidelberg*, pp. 1-37.

Yagoubi, B., & Slimani, Y. (2006). Dynamic load balancing strategy for grid computing. Transactions on Engineering, Computing and Technology. Vol. 13, pp. 260-265.

Yu, X., & Yu, X. (2009). A new grid computation-based Min-Min algorithm. In Fuzzy Systems and Knowledge Discovery, 2009. Fuzzy System and Knowledge Discovery (FSKD)'09. IEEE Sixth International Conference. Vol. 1, pp. 43-45.