MULTI-ROUND DIVISIBLE REAL-TIME SCHEDULING ALGORITHM ON MULTIPROCESSOR PLATFORMS

PEGAH RAZMARA

UNIVERSITI TEKNOLOGI MALAYSIA

MULTI-ROUND DIVISIBLE REAL-TIME SCHEDULING ALGORITHM ON MULTIPROCESSOR PLATFORMS

PEGAH RAZMARA

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

OCTOBER 2015

To my beloved mother and father

ACKNOWLEDGEMENT

In the name of Allah The Most Gracious The Most Merciful, Praise is to Allah who created us and gave us intelligence and guidance and peace is upon our prophet the teacher of all mankind and peace is upon his family.

I would like to thank my supervisor Dr.Suriayati bt Chuprat , the most patient person as far as know, for her advises , guidance , encouragement and support in my research.

My special thanks go to my family. My mother and father deserve special mention for their inseparable prayer, encouragement and endless patient.

ABSTRACT

Recent real-time systems and applications are becoming more complex and contain more functionality. Therefore, these systems are increasingly to be implemented upon multiprocessor platforms, as they require complex sharing of data, synchronization and parallelism. To overcome this limitation, recent researches have applied Divisible Load Theory (DLT) to real-time multiprocessor scheduling and the theory is known as Real-time Divisible Load Theory (RT-DLT). However, most current studies in this field are about distributing data in single-round algorithm and there are limited studies in multi-round strategy in real-time systems to reduce idle time. Moreover, current multi-round studies have some performance problems mainly due to inefficient use of available resources and long execution time for task scheduling. This research is carried out to address the problem of task execution on real-time multiprocessor platforms to reduce inserted idle time in order to meet task deadline. Therefore to achieve that, this research developed three significant multi-round algorithms which are: MultiMINPROCS, OPTROUND and MINCOMPTIME in expanding the current single-round RT-DLT to multi-round RT-DLT. Series of experimental evaluations showed that the three developed algorithms had improved the performance of previous both single-round and multi-round algorithms. The first algorithm computed the minimum number of processors needed to complete the job by its deadline, 40% improved the previous single-round algorithm and 33% improved previous multi-round algorithm. The second algorithm determined the most efficient number of round. Finally the third algorithm computed the minimum completion time in order to meet the task's deadline, 35% improved the previous single-round algorithm and 38% improved previous multi-round algorithm.

ABSTRAK

Sistem masa nyata dan aplikasi terkini menjadi semakin kompleks dan mengandungi lebih banyak fungsi. Justeru itu, sistem ini semakin kerap dilaksanakan dalam platform pelbagai pemproses, disebabkan oleh keperluan dalam perkongsian data, penyelarasan dan keselarian yang kompleks. Bagi mengatasi kekurangan ini, kajian sebelum ini telah menggunakan Teori Pembahagian Beban (DLT) bersama penjadualan multipemproses masa nyata dan teori ini dikenali sebagai Teori Pembahagian Beban Masa Nyata (RT-DLT). Walau bagaimanapun, kebanyakan kajian semasa dalam bidang ini adalah merangkumi data dalam algoritma pusingan tunggal dan terhad kepada kajian dalam strategi multi-pusingan sistem masa nyata untuk mengurangkan masa melahu. Selain itu, kajian semasa pusingan pelbagai ini mempunyai masalah prestasi disebabkan oleh penggunaan sumber sedia ada yang tidak efisien dan mempunyai masa pelaksanaan yang lama dalam penjadualan tugas. Kajian ini dijalankan untuk menangani masalah pelaksanaan tugas pada platform masa nyata multipemproses untuk mengurangkan kemasukan masa melahu bagi memenuhi tempoh had tugas. Oleh itu, bagi mencapai matlamat kajian ini tiga algoritma penting multi-pusingan iaitu MultiMINPROCS, OPTROUND dan MINCOMPTIME telah dibangunkan bagi menambah balik algorithma pusingan tunggal RT-DLT kepada multi-pusingan RT-DLT. Pengujian eksperimen secara bersiri menunjukkan pembangunan algoritma bertambah baik bagi pusingan tunggal dan multi-pusingan algoritma sebelumnya. Algoritma pertama mengambil kira bilangan minimum pemproses yang diperlukan untuk menyelesaikan tugas sebelum tempoh had adalah 40% lebih baik daripada algoritma pusingan tunggal dan 33% lebih baik daripada algoritma multi-pusingan kajian sebelumnya. Algoritma kedua menentukan bilangan pusingan paling cekap. Algoritma ketiga pula mengira masa minimum tempoh had diselesaikan, dengan 35% lebih baik daripada algoritma pusingan tunggal dan 38% lebih baik daripada algoritma multi-pusingan kajian sebelumnya.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DLT	-	Divisible Load Theory
DM	-	Deadline Monotonic
EDF	-	Earliest Deadline First
EDZL	-	Earliest Deadline Zero Laxity
FX	-	Fixed Priority
FIFO	-	First In First Out
IIT	-	Inserted Idle Time
LP	-	Linear Programming
MP	-	Multi-processor Platforms
QoS	-	Quality of System
RM	-	Rate Monotonic
RT-DLT	-	Real-time Divisible Load Theory

LIST OF SYMBOLS

A	-	Arrival Time
A_i	-	Arrival time of i^{th} Task
C_i	-	Computation Requirement of i^{th} Task
C_P	-	Computation Cost
C_m	-	Communication Cost
CH_j	-	Chunk Size of j^{th} Round
C(n)	-	Completion Time
c_i	-	i^{th} Communication Speed
D	-	Deadline
D_i	-	Deadline of i^{th} Task
L_i	-	i^{th} Link
m	-	Number of Rounds
n	-	Number of Processors
n_{min}	-	Minimum Number of Processors
P_i	-	i^{th} Processor
r_i	-	Ready Time i^{th} Task
s_i	-	Start Time i^{th} Task
T_f	-	Processing Time
T_i	-	i^{th} Task
T_i	-	Idle Time
V	-	Total Size of Each Round
W_{total}	-	Total Workload
w_i	-	i^{th} Computation Speed

Greek Symbols

σ_i	-	i^{th} Workload
σ	-	Total Size of Workload
α_i	-	i^{th} Fraction of workload
β	-	Ratio of C_P and (C_P + C_m)
Δ	-	Time between Current Instant and Deadline
ξ	-	Execution Time

CHAPTER 1

INTRODUCTION

1.1 Overview

In this chapter, we present a general overview of the applications performance problem in current real-rime computational systems. In particular, our work is focused on data intensive parallel applications. This chapter introduces the motivation inspiring this work, as well as an overview of studies related to our research. In addition, it presents the goal and contributions of this work and describes the research motivation. Finally, we present the organization of this document.

1.2 Research Background

Real-time computer systems are systems that function efficiently and their correctness depends on meeting their performance criteria. In these systems, correctness of system behavior depends not only on the *logical* results of the computations, but also on the *temporal* instant that those issued are produced. Temporal restrictions of real-time systems are commonly specified as deadlines and these kinds of systems are assumed to complete their work and deliver their results on a timely basis. In other words, time is a vital part of the explanation of a real-time system.

Within this kind of system, there are two wide groups: hard real-time and soft real-time systems. Hard real-time systems are those that have a strict attachment to deadline constraints; or else, the consequence is disastrous(Buttazzo, 2005, Buttazzo, 2011). In these kinds of systems, to guarantee deadline, we need to know the worst case execution times and for predictability, we need to know if deadlines may be missed.

One of the examples of a hard real-time system is a system of flight control, which in this system, if it does not respond to pilot's command within microseconds, the system might fail and may cause catastrophic situations.

In contrast, soft real-time systems are those that do not have a strict attachment to deadline constraints (Buttazzo et al., 2006); but it is desirable to do so and if deadline is missed, there is a penalty. In this kind of system, we should provide statistical guarantees and we need to know the statistical distributions of execution times.

In other words, missing deadline in hard real-time systems causes disaster and in soft real-time systems, it can lead to a serious loss. Streaming media player is an example of soft real-time system, which means that if the system does not consider the performance criteria in a single step, then the quality of system becomes reduced and eventually may be lost. Most real-time systems have combination of both hard real time and soft real time tasks. In this research, hard real-time system is a major concern.

In recent times, real time applications require composite and increased functionality significantly and it would not be reasonable upon uniprocessor platforms to implement them. Thus, these systems consider using implementation upon multiprocessor platforms increasingly, with complexity in sharing of data, synchronization and requirements of parallelism. However, formal models in real time workloads are *specifically* designed for the modeling of processes executed in uniprocessor platforms and in the capture characteristic of multiprocessor real-time systems that these models prone to fail. Moreover, they may enforce more restrictions upon implementation and design of system. Mok and Dertouzos (1978) showed that the algorithms that are optimal for single processor systems are not optimal for increased numbers of processors.

One of limitations from models of uniprocessor to multiprocessor is that at each time, only one task can be executed upon at most one processor. It means that, the task does not have permission to execute in parallel platforms. Therefore, to solve this problem, many real-time models and scheduling algorithms have been explored. Divisible load model which is distributed by divisible load theory is a computation model that can be divided arbitrarily to different load pieces of workload and would be able to provide a good real-world application.

In general, if a scientific application is appropriately designed to take advantage of systems parallelism, its executions would be usually carried out in a fast and efficient way. Nevertheless, to process data efficiently is not only a matter of having enough processing units, but it also depends on specific characteristics of the workload of the application. In many cases, these applications can be naturally implemented in parallel by partitioning their data sets into smaller pieces and distributing them among the processing units of the parallel system. However, each partition may have different processing times and this situation may lead to significant imbalances in the execution time of the processing units of the application.

Lin et al. (2007, 2010) and Chuprat (2007, 2008, 2010 and 2011) and Mamat (2008, 2009, 2010, 2011 and 2012) have done research in this area and have applied *Divisible Load Theory (DLT)* to real-time systems in multiprocessor platforms. However, the process of load distribution causes communication delays and idle time for almost all the processing nodes since a processor can start computing only after receiving the entire load fraction assigned to it.

1.3 Motivation

The motivation of this research is to extend Divisible Load Theory (DLT) to real-time systems on multiprocessor platforms to schedule the large real-time task and reduce processor idle time during the initialization of computation phase. Therefore to achieve that, the load fractions are sent in more than one round and totally, the whole workload is distributed between the workers in multi-round algorithm in many small installments or fractions rather than one big fraction in single-round algorithm. Moreover, another motivation of this research is to optimize the utilization of the resources by efficient use of available resources and minimize the processing time of the task is distributed between the processors in order to meet the task deadline. Therefore, this study, will introduce multi-round algorithms in real-time multi-processor applications to improve the previous findings.

1.4 Problem Statement

The subject of scheduling a divisible workload on real-time multiprocessor platforms is quite understood when all processors become available instantly at the same time. However, in real systems, usually, all processors that are required at the start time of scheduling are not available because of previous scheduling task or local task. This reason causes inserted idle time. Most current studies in this field are about distributing data in single-round algorithm and there are limited studies in multi-round algorithm. In Divisible Load Theory (DLT), it is known that, distributing with single round promotes idle-time and aggregated idle-time would increase completion time and more processing nodes. Moreover, existing multi-round studies have some performance problems mainly due to inefficient use of available resources and long execution time for task scheduling. These issues lead to general research question:

What is the effective multi-round algorithm to extend Real-Time Divisible Load Theory (RT-DLT) to scheduling of real-time workloads upon multiprocessor platforms which reduce the idle time?

Specifically, we consider four important issues:

i. *How can we calculate the minimum number of processors upon multiprocessor platforms in order to meet a job's deadline?*

ii. *How can we calculate the optimal number of rounds upon multiprocessor platforms in order to meet a job's deadline?*

iii. *How can we compute the earliest completion time in order to meet a job's deadline?*

iv. *What is the effective multi-round algorithm to extend Real-Time Divisible Load Theory (RT-DLT) upon multiprocessor platforms in order to meet a job's deadline by evaluate and comparison with the previous multi-round algorithm?*

Therefore, to solve all the afore-mentioned problems, we will use multi-round algorithms instead of single-round algorithm and will extend and develop it accordingly, so as to calculate the minimum number of processors and minimum execution time that requires meeting an application's deadline in order to reduce idle time.

1.5 Research Objectives

The main aim of this research is to extend the art of Real-Time Divisible Load Theory (RT-DLT) for multiprocessor hard real-time scheduling in order to reduce processor idle time during computation and communication. Therefore, in this research we will design and develop multi-round algorithm which is designed for multiprocessor hard real-time systems in order to improve the previous single-round and multi-round method to reduce initial idle time, optimum utilization of processors and minimize the task execution time . Some of our objectives which we have distinguished to achieve this aim are:

i. To build an effective multi-round scheduling algorithm that will calculate the minimum number of processors that must be assigned to a job in order to meets the deadline.

ii. To design an efficient multi-round scheduling algorithm to determining the most efficient number of rounds.

iii. To develop a scheduling multi-round algorithm to determine the minimum completion time upon a number of processor of multi-processor platform in order to meet deadline.

iv. To compare and evaluate our real-time multi-round algorithms to previous multi-round algorithm.

1.6 Research Scope and Limitations

In this research, we use special formal real-time model that is used in many real-time distributed system designs. Thus, there are certain units which are known as workloads that require performing. In the base of real-time scheduling, there are many constraints being observed. In this research, we limit our studies on the only one of these limits which is the deadline of workloads or jobs. With consideration of the resource system, in this research, we focus on determining the minimum number of processors used and the optimal number of rounds needed for our multi-round algorithm. Moreover, the minimum completion time will be another finding in this research. Even though other resources of system, such as energy and bandwidth of network are also important, they are not our scopes within this research.

Furthermore, in RT-DLT, several network topologies like stars, meshes and tree have been used. But we will limit our research to the single-level tree topology. This kind of topology could be the simplest model among other network topologies but contain many important issues when DLT is applied to real-time systems. So far, most researchers have developed and analyzed divisible load theory with *single-round* strategy.

However, there are many problems like blocking in scheduling and idle time for almost all processors in this method and a processor can start executing the workload fraction only after receiving the whole load fraction assigned to it. Therefore, these complexities will be solved effectively by utilizing divisible load theory with *multi-round* strategy. Thus we will restrict our research only to divisible load theory with multi-round method.

Also, there are some kinds of scheduling algorithms that can be joined potentially into RT-DLT such as Earliest Deadline Zero Laxity (EDZL)(Baker et al., 2008), Deadline Monotonic (Leung and Whitehead, 1982, Audsley et al., 1993b), Rate Monotonic(Liu and Layland, 1973, Baruah and Goossens, 2003) and etc. But, our work is related to Earliest Deadline First (EDF) (Liu and Layland, 1973) , (Baruah et al., 2003).

1.7 Summary of Contributions

The actual contributions in this research are generally relevant to our objectives that we have defined in the previous sections.

In some data, particularly intensive applications on real-time platforms, dividing the workloads into data fractions does not guarantee meeting the deadlines stipulated. It means that, total execution time of workload often exceeds the task deadline. Moreover, recent methods do not consider resource management efficiency and could not utilize IITs completely. Accordingly, we will design and implement an optimum methodology which will be applied on real-time multi-processor platforms with proper subset of task intensive application.

Therefore, to prove all issues in this thesis; we will produce 4 main contributions that can be summarized as:

i. Introducing an algorithm to calculate the minimum number of processors by creating a multi-round scheduling algorithm that must be assigned to a task in order to meet deadline. In this multi-round algorithm, we will divide the workload into small fractions and will allocate them to working processor in optimal number of rounds. Thus, the number of processors used will be minimal.

ii. Creating an effective multi-round scheduling algorithm to determine the most efficient number of rounds. In this algorithm, we will select the round with minimum number of processors and the earliest completion time as an optimal round in multi-round algorithm.

iii. Developing a scheduling multi-round algorithm to determine earliest completion time upon a number of processor on multi-processor platform in order to meet deadline. Thus, for approving this method, we will use multi-round algorithms to determine the minimum processors and the optimal number of round for calculating the earliest completion time.

iv. Evaluating our scheduling real-time multi-round algorithms to previous multi-round algorithm to ensure that the designed multi-round algorithm will assist to overcoming the idle time concern as well as decrease the processing nodes and task completion time.

1.8 Organization of Thesis

In this chapter, we have developed our objectives and have also described our contributions and motivation of our research. Moreover, we have specified our work limitations and scopes. In the next chapter, we will explain the literature review, related works and results on real-time systems and Divisible Load Theory (DLT). In Chapter 3, we will present our research methodology and research procedure that will be used in our work. So, taking such explanation as a starting point will lead to significant analysis in developing our research to calculate the minimum number of processing node in Chapter 4.

After that, Chapter 5 presents a multi-round algorithm to calculate the optimal number of rounds in our multi-processor platforms. In Chapter 6, we will calculate the minimum completion time in an optimal round by using minimum number of resources and in Chapter 7, we will evaluate and compare our foundlings with previous multi-round algorithm on multi-processor real-time systems. Finally, in Chapter 8, we will conclude our thesis and will present suggestions for the future studies in similar field. Therefore, in Figure 1.1, we present the total flow of our thesis accordingly.

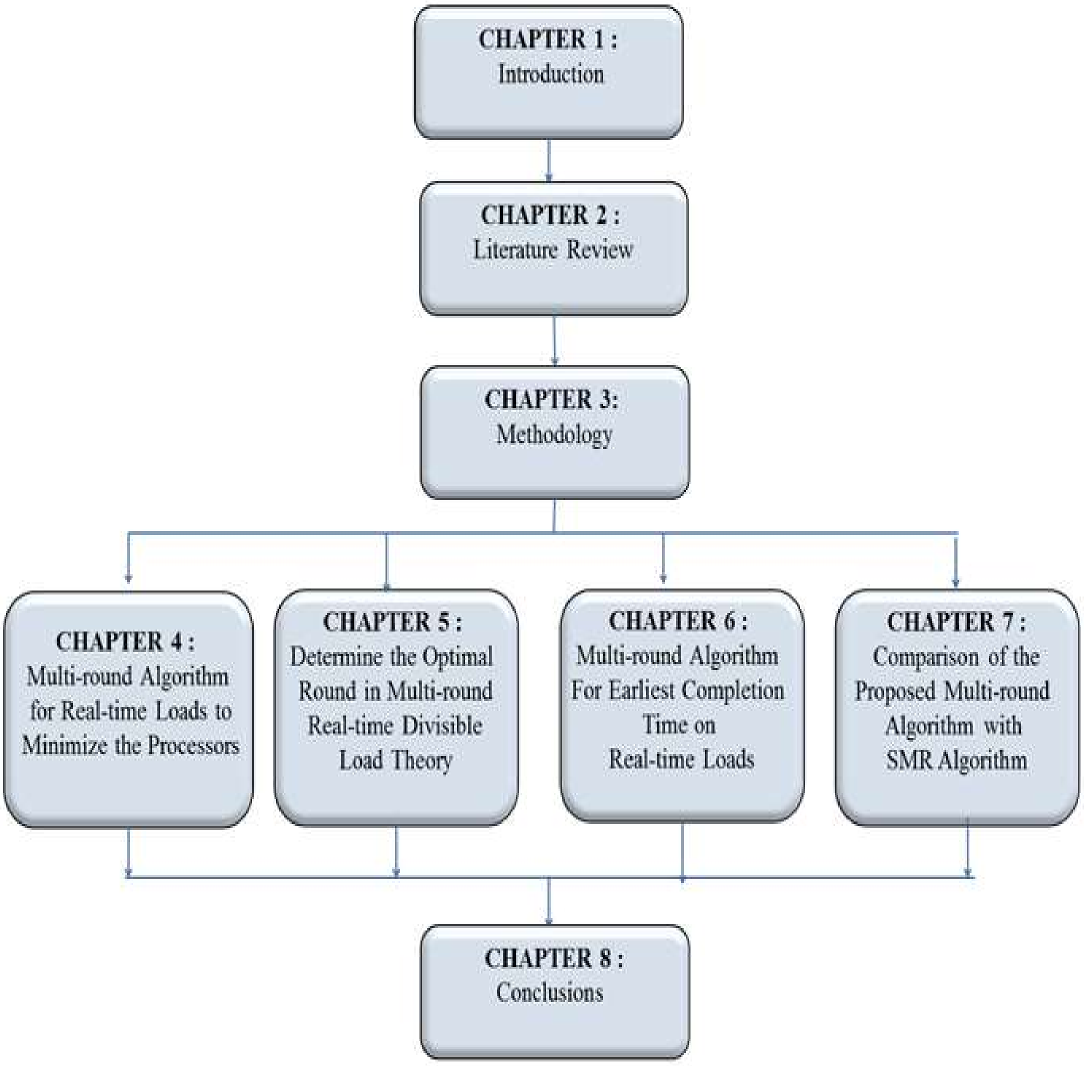

Figure 1. 1 Organization of Thesis

REFERENCES

Audsley, N., Burns, A., Richardson, M., Tindell, K. & Wellings, A. J. 1993a. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal,* 8**,** 284-292.

Audsley, N., Burns, A. & Wellings, A. 1993b. Deadline monotonic scheduling theory and application. *Control Engineering Practice,* 1**,** 71-78.

Audsley, N. C., Burns, A., Richardson, M. F. & Wellings, A. J. Real-Time scheduling: the deadline-monotonic approach. in Proc. IEEE Workshop on Real-Time Operating Systems and Software, 1991, 133-137.

Baker, T. P. & Baruah, S. K. 2007. Schedulability analysis of multiprocessor sporadic task systems. *Handbook of Real-Time and Embedded Systems***,** 3-1.

Baker, T. P., Cirinei, M. & Bertogna, M. 2008. EDZL scheduling analysis. *Real-Time Systems,* 40**,** 264-289.

Balafoutis, E., Paterakis, M., Triantafillou, P., Nerjes, G., Muth, P. & Weikum, G. 2003. Clustered scheduling algorithms for mixed-media disk workloads in a multimedia server. *Cluster Computing,* 6**,** 75-86.

Baruah, S. Feasibility analysis of preemptive real-time systems upon heterogeneous multiprocessor platforms. Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International, 2004a. IEEE, 37-46.

Baruah, S., Funk, S. & Goossens, J. 2003. Robustness results concerning EDF scheduling upon uniform multiprocessors. *IEEE Transactions on Computers,* 52**,** 1185-1195.

Baruah, S., Koren, G., Mao, D., Mishra, B., Raghunathan, A., Rosier, L., Shasha, D. & Wang, F. 1992. On the competitiveness of on-line real-time task scheduling. *Real-Time Systems,* 4**,** 125-144.

Baruah, S. K. 2004b. Optimal utilization bounds for the fixed-priority scheduling of periodic task systems on identical multiprocessors. *IEEE Transactions on Computers,* 53**,** 781-784.

Baruah, S. K. Partitioning real-time tasks among heterogeneous multiprocessors. Parallel Processing, 2004. ICPP 2004. International Conference on, 2004c. IEEE, 467-474.

Baruah, S. K. Task Partitioning Upon Heterogeneous Multiprocessor Platforms. IEEE Real-Time and Embedded Technology and Applications Symposium, 2004d. Citeseer, 536-543.

Baruah, S. K. 2006. The non-preemptive scheduling of periodic tasks upon multiprocessors. *Real-Time Systems,* 32**,** 9-20.

Baruah, S. K., Cohen, N. K., Plaxton, C. G. & Varvel, D. A. 1996. Proportionate progress: A notion of fairness in resource allocation. *Algorithmica,* 15**,** 600-625.

Baruah, S. K., Gehrke, J. E. & Plaxton, C. G. Fast scheduling of periodic tasks on multiple resources. *Parallel Processing Symposium, International*, 1995. IEEE Computer Society, 280-280.

Baruah, S. K. & Goossens, J. 2003. Rate-monotonic scheduling on uniform multiprocessors. *IEEE Transactions on Computers,* 52**,** 966-970.

Baruah, S. K., Mok, A. K. & Rosier, L. E. Preemptively scheduling hard-real-time sporadic tasks on one processor. *Real-Time Systems Symposium*, 1990. Proceedings., 11th, 1990. IEEE, 182-190.

Bataineh, S., Hsiung, T.-Y. & Robertazzi, T. G. 1994. Closed form solutions for bus and tree networks of processors load sharing a divisible job. *IEEE Transactions on Computers,* 43**,** 1184-1196.

Bharadwaj, V. 1996. *Scheduling divisible loads in parallel and distributed systems*, John Wiley & Sons.

Bharadwaj, V., Ghose, D. & Mani, V. 1994. Optimal sequencing and arrangement in distributed single-level tree networks with communication delays. *Parallel and Distributed Systems, IEEE Transactions on,* 5**,** 968-976.

Bharadwaj, V., Ghose, D. & Mani, V. 1995. Multi-installment load distribution in tree networks with delays. *Aerospace and Electronic Systems, IEEE Transactions on,* 31**,** 555-567.

Bharadwaj, V., Ghose, D., Mani, V. & Robertazzi, T. G. 1996. *Scheduling divisible loads in parallel and distributed systems*, Wiley-IEEE Computer Society Press.

Bharadwaj, V., Ghose, D. & Robertazzi, T. G. 2003. Divisible load theory: A new paradigm for load scheduling in distributed systems. *Cluster Computing,* 6**,** 7-17.

Bokhari, S. H. 1981. On the mapping problem. *Computers, IEEE Transactions on,* 100**,** 207-214.

Bubendorfer, K. P. & Hine, J. H. 1999. *A compositional classification for load-balancing algorithms*, School of Mathematical and Computing Sciences, Victoria University of Wellington.

Burns, A. & Wellings, A. J. 2001. *Real-time systems and programming languages: Ada 95, real-time Java, and real-time POSIX*, Pearson Education.

Buttazzo, G. C. 2005. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. *Real-Time Systems*.

Buttazzo, G. C. 2011. *Hard real-time computing systems: predictable scheduling algorithms and applications*, Springer.

Buttazzo, G. C., Lipari, G., Abeni, L. & Caccamo, M. 2006. *Soft Real-Time Systems: Predictability vs. Efficiency*, Springer.

Casavant, T. L. & Kuhl, J. G. 1988. A taxonomy of scheduling in general-purpose distributed computing systems. *Software Engineering, IEEE Transactions on,* 14**,** 141-154.

Chan, S., Bharadwaj, V. & Ghose, D. 2001. Large matrix–vector products on distributed bus networks with communication delays using the divisible load paradigm: performance analysis and simulation. *Mathematics and Computers in Simulation,* 58**,** 71-92.

Cheng, Y.-C. & Robertazzi, T. G. 1988. Distributed computation with communication delay (distributed intelligent sensor networks). *Aerospace and Electronic Systems, IEEE Transactions on,* 24**,** 700-712.

Chuprat, S. Divisible load scheduling of real-time task on heterogeneous clusters. Information Technology (ITSim), 2010 International Symposium in, 2010. IEEE, 721-726.

Chuprat, S. & Baruah, S. Scheduling divisible real-time loads on clusters with varying processor start times. *Embedded and Real-Time Computing Systems and Applications, 2008. RTCSA'08. 14th IEEE International Conference on,* 2008. IEEE, 15-24.

Chuprat, S. & Baruah, S. K. Deadline-based scheduling of divisible real-time loads. *ISCA PDCS*, 2007. 7-12.

Chuprat, S. & Baruah, S. K. Real-Time Divisible Load Theory: Incorporating Computation Costs. *RTCSA (1)*, 2011. 33-37.

Chuprat, S. & Salleh, S. 2007. A deadline-based algorithm for dynamic task scheduling with precedence constraints. *Proceedings of the 25th confernece on Proceedings of the 25th IASTED International Multi-Conference: parallel and distributed computing and networks ,* 551**,** 158.

Chuprat, S., Salleh, S. & Baruah, S. K. Evaluation of a linear programming approach towards scheduling divisible real-time loads. *Information Technology, 2008. ITSim 2008. International Symposium on, 2008*. IEEE, 1-8.

Cvetanovic, Z. 1987. The effects of problem partitioning, allocation, and granularity on the performance of multiple-processor systems. *IEEE Transactions on Computers,* 100**,** 421-432.

Dertouzos, M. L. & Mok, A. K. 1989. Multiprocessor online scheduling of hard-real-time tasks. *Software Engineering, IEEE Transactions on,* 15**,** 1497-1506.

Devine, K. D., Boman, E. G. & Karypis, G. 2006. Partitioning and load balancing for emerging parallel applications and architectures. *Parallel Processing for Scientific Computing,* 20,93- 99.

Drozdowski, M. & Lawenda, M. 2006. *Multi-installment divisible load processing in heterogeneous systems with limited memory*. *Parallel Processing and Applied Mathematics.* 847-854.Springer.

Katwijk, J. V. & Zalewski, J. 2001. Parallel and distributed real-time systems: An introduction. *Scalable Computing: Practice and Experience,* 2.

Kim, H. J., Jee, G.-I. & Lee, J. G. 1996. Optimal load distribution for tree network processors. *Aerospace and Electronic Systems, IEEE Transactions on,* 32**,** 607-612.

Krueger, P. & Livny, M. 1987. *Load balancing, load sharing and performance in distributed systems*, University of Wisconsin-Madison, Computer Sciences Department.

Lee, W. Y., Hong, S. J. & Kim, J. 2003. On-line scheduling of scalable real-time tasks on multiprocessor systems. *Journal of Parallel and Distributed Computing,* 63**,** 1315-1324.

Leung, J. Y.-T. & Whitehead, J. 1982. On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance evaluation,* 2**,** 237-250.

Li, X., Liu, X. & Kang, H. Sensing workload scheduling in sensor networks using divisible load theory. *Global Telecommunications Conference,* 2007. GLOBECOM'07. IEEE, 2007. IEEE, 785-789.

Li, X., Veeravalli, B. & Ko, C. 2003. Distributed image processing on a network of workstations. *International Journal of Computers and Applications,* 25**,** 136-145.

Lin, X., Deogun, J., Lu, Y. & Goddard, S. 2008. Multi-round real-time divisible load scheduling for clusters. *High Performance Computing-HiPC 2008.* Springer.

Lin, X., Lu, Y., Deogun, J. & Goddard, S. 2007a. Enhanced real-time divisible load scheduling with different processor available times. *High Performance Computing–HiPC 2007.* Springer.

Lin, X., Lu, Y., Deogun, J. & Goddard, S. Real-time divisible load scheduling for cluster computing. *Real Time and Embedded Technology and Applications Symposium,* 2007. RTAS'07. 13th IEEE, 2007b. IEEE, 303-314.

Lin, X., Lu, Y., Deogun, J. & Goddard, S. Real-time divisible load scheduling with different processor available times. *Parallel Processing, 2007. ICPP 2007. International Conference on, 2007c*. IEEE, 20-20.

Lin, X., Mamat, A., Lu, Y., Deogun, J. & Goddard, S. 2010. Real-time scheduling of divisible loads in cluster computing environments. *Journal of Parallel and Distributed Computing,* 70**,** 296-308.

Liu, C. L. & Layland, J. W. 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM),* 20**,** 46-61.

Mamat, A. Real-time divisible load scheduling for cluster computing. THE UNIVERSITY OF NEBRASKA-LINCOLN , 2011.

Mamat, A., Lu, Y., Deogun, J. & Goddard, S. Real-time divisible load scheduling with advance reservation. *Real-Time Systems, 2008. ECRTS'08. Euromicro Conference on,* 2008. IEEE, 37-46.

Mamat, A., Lu, Y., Deogun, J. & Goddard, S. An efficient algorithm for real-time divisible load scheduling. *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2010 16th IEEE, 2010. IEEE, 323-332.

Mamat, A., Lu, Y., Deogun, J. & Goddard, S. 2012a. Efficient real-time divisible load scheduling. *Journal of Parallel and Distributed Computing,* 72**,** 1603-1616.

Mamat, A., Lu, Y., Deogun, J. & Goddard, S. 2012b. Scheduling real-time divisible loads with advance reservations. *Real-Time Systems,* 48**,** 264-293.

Mamat, A., Lu, Y., Deogun, J. S. & Goddard, S. 2009. An Efficient Algorithm for Real-Time Divisible Load Scheduling. *CSE Technical reports***,** 74.

Manimaran, G. & Murthy, C. S. R. 1998. An efficient dynamic scheduling algorithm for multiprocessor real-time systems. *Parallel and Distributed Systems, IEEE Transactions on,* 9**,** 312-319.

Mok, A. K.-L. & Dertouzos, M. L. 1978. Multiprocessor scheduling in a hard real-time environment, *Domain Specific Systems Group [Massachusetts Institute of Technology].*

Moore, R., Prince, T. A. & Ellisman, M. 1998. Data-intensive computing and digital libraries. *Communications of the ACM,* 41**,** 56-62.

Ouelhadj, D. & Petrovic, S. 2009. A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling,* 12**,** 417-431.

Paprzycki, M. & Zalewski, J. 1993. Call for Papers Distributed and Parallel Real Time Systems Special Issue of INFORMATICA. *An International Journal of Computing and Informatics,* 17**,** 413-419.

Ramamritham, K., Stankovic, J. A. & Shiah, P.-F. 1990. Efficient scheduling algorithms for real-time multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems,* 1**,** 184-194.

Robertazzi, T. G. 1993. Processor equivalence for daisy chain load sharing processors. *IEEE Transactions on Aerospace and Electronic Systems,* 29**,** 1216-1221.

Robertazzi, T. G. 2003. Ten reasons to use divisible load theory. *Computer,* 36**,** 63-68.

Robertazzi, T. G. 2007. Networks and grids: technology and theory, *Springer Science & Business Media.*

Rommen, C. 1991. The probability of load balancing success in a homogeneous network. *IEEE Transactions on Software Engineering,* 17**,** 922-933.

Shokripour, A. & Othman, M. Categorizing researches about DLT in Ten groups. *Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09*. International Association of, 2009a. IEEE, 45-49.

Shokripour, A. & Othman, M. Survey on divisible load theory and its applications. Information Management and Engineering, 2009. ICIME'09. *International Conference on, 2009b*. IEEE, 300-304.

Shokripour, A., Othman, M. & Ibrahim, H. 2010. A new algorithm for divisible load scheduling with different processor available times. *Intelligent Information and Database Systems*, 221-230.

Shokripour, A., Othman, M., Ibrahim, H. & Subramaniam, S. 2011. A method for scheduling heterogeneous multi-installment systems. *Intelligent Information and Database Systems.* Springer.

Shokripour, A., Othman, M., Ibrahim, H. & Subramaniam, S. 2012. New method for scheduling heterogeneous multi-installment systems. *Future Generation Computer Systems,* 28, 1205-1216.

Shokripour, A., Othman, M., Ibrahim, H. & Subramaniam, S. 2013. A new method for job scheduling in two-levels hierarchical systems. *Intelligent Information and Database Systems.* Springer.

Sohn, J. & Robertazzi, T. 1995. An optimum load sharing strategy for divisible jobs with time-varying processor speed and channel speed. *Proceedings of the ISCA International Conference on Parallel and Distributed Computing Systems*. 27-32.

Sohn, J. & Robertazzi, T. G. 1996. Optimal divisible job load sharing for bus networks. *IEEE Transactions on Aerospace and Electronic Systems,* 32, 34-40.

Srinivasan, A. & Baruah, S. 2002. Deadline-based scheduling of periodic task systems on multiprocessors. *Information Processing Letters,* 84, 93-98.

Veeravalli, B., Li, X. & Ko, C. C. 2000. On the influence of start-up costs in scheduling divisible loads on bus networks. *Parallel and Distributed Systems, IEEE Transactions on,* 11, 1288-1305.

Veeravalli, B. & Ranganath, S. 2002. Theoretical and experimental study on large size image processing applications using divisible load paradigm on distributed bus networks. *Image and Vision Computing,* 20, 917-935.

Willebeek-Lemair, M. H. & Reeves, A. P. 1993. Strategies for dynamic load balancing on highly parallel computers. *Parallel and Distributed Systems, IEEE Transactions on,* 4**,** 979-993.

Wolniewicz, P. & Drozdowski, M. 2002. Processing time and memory requirements for multi-instalment divisible job processing. *Parallel Processing and Applied Mathematics.* Springer.

Xu, C. 1997. *Load balancing in parallel computers: theory and practice*, Springer.

Yang, Y. & Casanova, H. UMR: A multi-round algorithm for scheduling divisible workloads. Parallel and Distributed Processing Symposium, 2003. Proceedings. International, 2003. IEEE, 9 pp.

Yang, Y., Casanova, H., Drozdowski, M., Lawenda, M. & Legrand, A. 2007. On the complexity of multi-round divisible load scheduling.

Yang, Y., Van Der Raadt, K. & Casanova, H. 2005. Multiround algorithms for scheduling divisible loads. *IEEE Transactions on Parallel and Distributed Systems,* 16**,** 1092-1102.