DYNAMIC REPLICATION AWARE LOAD BLANCED SCHEDULING IN DISTRIBUTED ENVIRONMENT

SAID BAKHSHAD

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

DYNAMIC REPLICATION AWARE LOAD BLANCED SCHEDULING IN DISTRIBUTED ENVIRONMENT

SAID BAKHSHAD

THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR
2018

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Said Bakhshad

Registration/Matric No: WGA130049

Name of Degree: Master

Title: Dynamic Replication Aware Load Balanced Scheduling in Distributed Enviroment

Field of Study: Wireless Network

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

DYNAMIC REPLICATION AWARE LOAD BLANCED SCHEDULING IN DISTRIBUTED ENVIRONMENT (GRID)

ABSTRACT

Grid computing is an effective distributed and adaptable processing network that manages a huge number of data applications. Proficient usage of existing resources in distributed grid network is still of great demand today. Additionally, it is of more crucial demand in a very dynamic dispersed surrounding such as grid. The grid processing is a viable computing surrounding. Data replication is viewed as a vital boost mechanism in data grids. The storage space limitations of traditional distributed systems can be overcome, to completely point out the resources of computing sites of under-utilized computing resources in the distributed environments. However, the scope state of-the-art replication procedures ignore the replica locations during jobs scheduling. It assigns the request of the node and then the node's Replication Manager searches the presence of the replica. Several algorithms have been proposed and studied for scheduling and data replication, however a little research has been done so far on capturing and minimizing the migration rate of data from an existing available replica site to a next site on the basis of data scheduling in order to minimize the transfers and deletion rate. In this regard, Modified Dynamic Hierarchical Replication (MDHR) is one of the recent and important effort toward this issue. MDHR is dependent on the last request of the data replica, size of the replica, no. of accesses, and it chooses the outstanding replica from the replica list based on turnaround time or response time, the access latency, demand waiting in queue for execution, the grid sites distance and CPU capability of processing. But it did not consider replica location at the time of scheduling which led to increase in the execution time and data migration rate as well. In this manner, scheduling is critical, which causes the assigining of job to site with replica. In this case, if jobs are not scheduled properly at particular points, the processing resources will be squandered.

We propose a novel dynamic Replication Aware Load Balanced Scheduling (DRALBS) algorithm, that considers the replica location dynamically at the time of the scheduling of the job. The simulation of the proposed algorithm shows promising results and better performance compared to the current state-of-the-art (MDHR) algorithm. The response and average access time has been significantly decreased, thus reducing the overall mean job execution time data migration and deletion rate as well as bandwidth consumption.

Keywords: Scheduling, Data Grid, Replication, Migrations, Gridsim

ABSTRAK

Pengkomputeran grid adalah rangkaian pemprosesan yang diedarkan dan disesuaikan dengan berkesan yang menguruskan sejumlah besar aplikasi data. Penggunaan mahir sumber sedia ada dalam rangkaian grid diedarkan masih banyak permintaan hari ini. Di samping itu, ia adalah permintaan yang lebih penting dalam rangkaian yang tersebar dengan sangat dinamik seperti grid. Pemprosesan grid adalah pengkomputeran yang bersesuaian. Replikasi data dilihat sebagai mekanisme peningkatan penting dalam grid data. Keterbatasan ruang penyimpanan sistem diedarkan tradisional dapat diatasi, untuk sepenuhnya menunjukkan sumber-sumber laman pengkomputeran sumber pengkomputeran yang tidak digunakan dalam persekitaran yang diedarkan. Bagaimanapun, prosedur replikasi keadaan skop keadaan mengabaikan lokasi replika semasa penjadualan kerja. Ia memberikan permintaan nod dan kemudian Pengurus Replikasi nod mencari kehadiran replika. Beberapa algoritma telah dicadangkan dan dipelajari untuk penjadualan dan replikasi data, tetapi sedikit kajian telah dilakukan setakat ini untuk menangkap dan meminimumkan kadar penghijrahan data dari tapak replika sedia ada ke tapak berikutnya berdasarkan penjadualan data untuk meminimumkan pemindahan dan kadar penghapusan. Dalam hal ini, Modified Dinamik Hierarki Replikasi (MDHR) adalah salah satu usaha baru-baru ini dan penting terhadap isu ini. MDHR bergantung kepada permintaan terakhir replika data, saiz replika, tidak. akses, dan ia memilih replika yang luar biasa dari senarai replika berdasarkan masa pemulihan atau masa tindak balas, latensi akses, permintaan menunggu dalam barisan untuk pelaksanaan, jarak tapak grid dan kemampuan pemprosesan CPU. Tetapi ia tidak menganggap lokasi replika pada masa penjadualan yang menyebabkan peningkatan dalam masa pelaksanaan dan kadar penghijrahan data juga. Dengan cara ini, penjadualan adalah kritikal, yang menyebabkan penyerahan kerja ke tapak dengan

replika. Dalam kes ini, jika pekerjaan tidak dijadualkan dengan betul pada titik tertentu, sumber pemprosesan akan habis.

Kami mencadangkan algoritma Penjadualan Kelebihan Bantu Penjejakan Beban (DRALBS) dinamik baru, yang menganggap lokasi replika secara dinamik pada masa penjadualan kerja. Simulasi algoritma yang dicadangkan menunjukkan hasil yang menjanjikan dan prestasi yang lebih baik berbanding dengan algoritma state-of-the-art (MDHR) semasa. Sambutan dan masa akses purata telah berkurangan dengan ketara, sekali gus mengurangkan kadar penghijrahan data masa kerja keseluruhan dan kadar penghapusan serta penggunaan bandwidth.

ACKNOWLEDGEMENTS

I begin by thanking my Benefactor, Allah Almighty of being indebted for His countless bounties and innumerable divine favors. I am beyond honored and deeply privileged to be grateful to Him for his richest blessings, grace, mercy, constant guidance and for always replenishing me when I need it the most throughout my careerThanks are not enough to be given to my parents, for their ongoing supports. There are no words that are valuable to equalize their love for me. This thesis would not have been possible without the support of my brother, Jamal, who always support me and encouraged me to pursue my study.

I would like to express my biggest gratitude to my supervisor, **Dr. Rafida Binti Md Noor,** for his invaluable support and guidance throughout my Master. My heartfelt appreciation goes to his commitment, encouragement, expertise, understanding, and patience that added largely to my graduate experience. I really felt thankful to have him as my supervisor and without his continued motivation; I would certainly not have considered my graduate career. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

I would like to extend my special thanks to my co-supervisor Dr. Wasif for his valuable assistance provided at all levels of the research project. He backed me with academic writing expertise, technical support and became more of a mentor and friend, than a supervisor. It was though his, persistence, understanding and kindness that I completed my thesis.

I would also like to express my regards towards my friends and family for the support they provided me through my entire life and in particular; I must acknowledge my elder brother and best friend, Saif ul-Islam, without whose love, constant motivation, and encouragement, I would not have a chance to pursue my dreams and make them come true.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

BR	Best Region
CDN	Content Distribution Network
CE	Computing Elemrnt
CPU	Central Processing Unit
DHR	Dynamic Hierarchical Replication
EDHR	Enhanced Dynamic Hierarchical Replication
FID	File ID
LAN	Local Area Network
MDHR	Modified Dynamic Hierarchical Replication
MIPS	Million of Instruction Per Second
NOA	Number of Access
PRS	Primary Replica Selver
QoS	Quality of Service
RB	Resource Broker
RC	Replica Manger
RPPT	Replica Placement Problem
RTSP	Replica Transfer Scheduling Problem
SE	Storage Element
SRS	Secondary Storage Server
WAN	Wide Area Network
WSS	Weight Scheduling Strategies

CHAPTER 1: INTRODUCTION AND OVERVIEW

1.1 Introduction

This chapter begins with a background study that gives an outline about the explored work and also presents the key motivation that led to the interest of carrying out this research in the chosen research area along with the significance of the proposed solution. Furthermore, the chapter also describes the problem statement and objectives to be achieved by following the proposed methodology of the research. Finally, the outline procedure on how to organize the thesis is presented as well.

1.2 Background

Grid computing is an effectively distributed and adaptable processing network that manages a huge number of data applications. Grid computing is an effective computing environment for geographical distributed resources. Replication of Data is a very important step for optimizing a large amount of data files by replicating the creation of data in many sites of the grid. Furthermore, the disadvantages in the storage capicity of traditionally dispersed systems can be defeated by completely pointing out the resources of computing sites and under-utilized computing resources in every region of the grid around the entire world for distributed jobs in the grid. Load and resource management are also very important services at the service level of grid computing infrastructure, where issues of the load balancing show a common thing for most grid computing infrastructure developers (Casas, Taheri, Ranjan, Wang, & Zomaya, 2017; Morris et al.,1986; Rajaretnam, Rajkumar, & Venkatesan, 2016).

The Grid is mainly partitioned into two categories, Data Grid and Computational Grid. Computational Grid is mainly for computational hungry applications which need a little scale of data. However, Data Grids require a platform that needs analysing and studying of massive big data intensive applications in a grid (Vashisht, Kumar, & Sharma, 2014;

Abrams, Standridge, Abdulla, Fox, & Williams, 1996; Souli-Jbali, Hidri, & Ayed, 2015). Replication of data is one of the important steps for optimizing the large amount of data files by replicating the creation of data in many sites of the grid. However, storage and memory size of resources are demanding more data day by day, but still they are very far from being aware of the requests of saving big amount of files. The main issue is to make a decision of the needed amount of replication copies to be created and the site in which they will be created. Replication is very important to ensure availability, reliability, scalability, adaptability etc. Scheduling is the method in which the replication of data are utilized to decrease make span time, consumption of storage, delay in data access and system file transfer capacity. Scheduling the job at its best site in Grid environment can minimize the data transferring or migrations on nodes, where many of the required and requested data files are available (Casas et al., 2017; Mansouri, 2014; Hunt, oldszmidt, King, & Mukherjee, 1998; Beck & Moore, 1998; Bhattacharjee, Ammar, Zegura, Shah, & Fei, 1997). In a Data Grid environment, minimizing job's response time and turn-around time (line size in which jobs are waiting, execution time of the job and data transfer time) mostly rely on which location should be placed for the execution of the job and where to fetch the required files of data for the job execution. Therefore, the scheduler/broker should assign the jobs to proper grid sites while ensuring minimum data transfers (Mansouri, 2016; Vashisht et al., 2014; Banga, Douglis, Rabinovich, et al., 1997). Secondly, the centralized data servers have been transformed into a performance bottleneck by the pervasive development of the Internet. In a very short span, thousands of requests may be received by popular sites. Such frequent queries may lead to overhead on servers and the network, resulting in increased delay of services being provided to the end users. Such bottleneck may prove to be more effective when we consider media servers or grids that may process very large data sets in order to process remote applications (Andresen, Yang, Holmedahl, & Ibarra, 1996; Abdi & Hashemi, 2015). Two different approaches can be used to control this issue which are, either the demands of client are distributed on several servers or the contents of a server are moved very close to the client. The first solution may result in decreased server load and faster processing of client requests whereas the second solution lowers the time it takes for the demand and network response for travel. Based on these approaches, caching, mirroring and replication are the three developed techniques. However, these techniques induce some complexity and increased operational cost (Mansouri, 2014; Bestavros, Crovella, Liu, & Martin, 1998; Casas et al., 2017). It is therefore not appropriate to avoid a task scheduling with such increased complexity, scalability and functionality in distributed systems. Unavailability of a certain resource under complex and critical situations may lead to data corruption and loss. In the worst case, it may result in premature termination of executing and thus frequent performance degradation. Even a millisecond delay in data access can cause a great loss for commercial business oriented applications in terms of customer's satisfaction. It may not only violate the service level agreement (SLA) but also cause effects to the customer's base and revenue respectively. Efficient scheduling is thus of critical importance in the paradigm of distributed systems (Mansouri, 2014; Abdi & Hashemi, 2015; Vashisht et al., 2014). Overview of research is given in the following sections.

1.3 Data Avialibility Techniques

There are several techniques used to store copies of original data for users access in such a way as to reduce execution time, data latency and bandwidth consumption as well as improve network efficiency. They may be either the client side or server side techniques. Some ordinary used techniques are described in the following sub sections.

1.3.1 Caching

Keeping the most commonly requested data closer to the requesting nodes temporarily is a very common technique known as Caching. A cache is commonly placed at the same machine as the client. By doing so, network access is completely eliminated when cache hits. Secondly, placing the data on a closer machine than the server which holds the original data causes lower overhead. However, caches are generally smaller than the size of the server-stored data.

Distributed client-server architectures use caching in order to enhance the system performance in environments such as LAN and WAN, e.g. AFS file system (Morris et al., 1986). Similar approaches are studied and applied in web conjunction, where the web pages or parts are considered as cache items. Dynamic page caching (Banga et al., 1997; Banga et al., 1997; Iyengar & Challenger, 1997), cache replacement policies (Abrams et al., 1996), pre-fetching (Fan, Cao, Lin, & Jacobson, 1999; Padmanabhan & Mogul, 1996), cache consistency (Cate, 1992; Cao & Liu,1998), and cache architectures (Dykes & Robbins, 2001; Gadde, Rabinovich, & Chase, 1997) are the several aspects of caching which have undergone a lot of research.

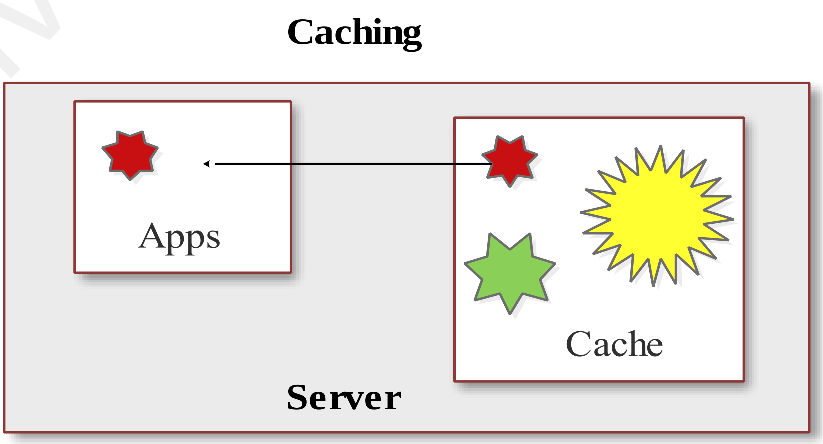


Figure 1.1 Caching

1.3.2 Data Storage via Mirroring

Mirroring is defined as the movement of data files from one place of storage to other spaces in real time manner. The data copying is in real time, the data file saved from the initial place is always an accurate copy of the data file of the original device. Data mirroring is good for quick recovery of important data after a failure. Mirroring of data can be implemented offsite or locally at other totally different site positions. See Figure 1.2 Mirroring

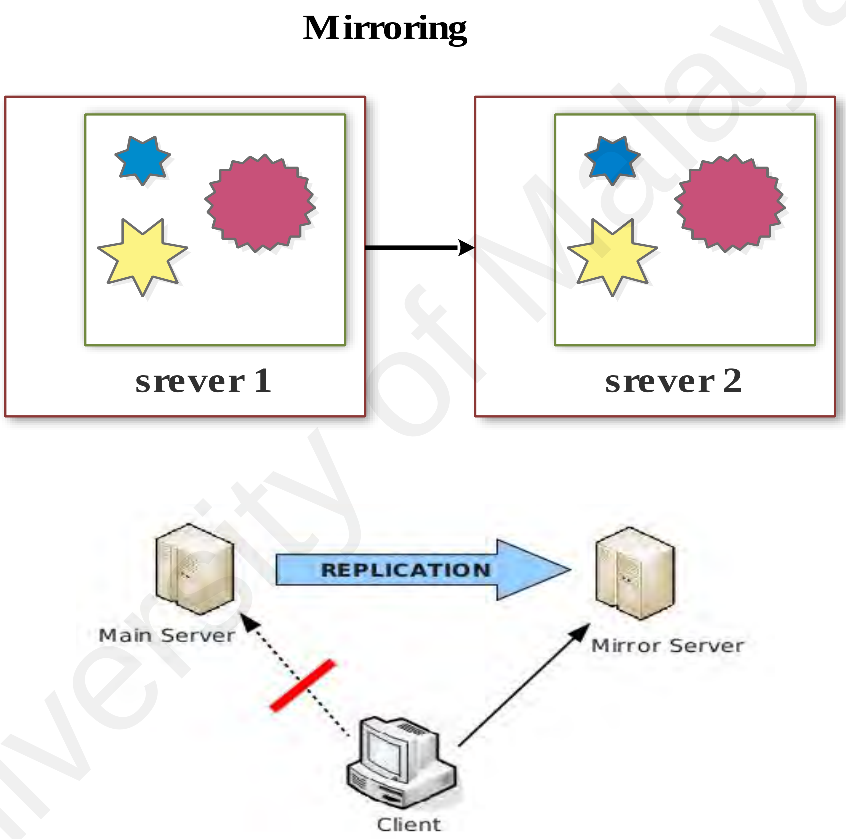


Figure 1.2 Mirroring

1.3.3 Replication

Replication is a technique that copies and distributes data while improving the system performance and accessibility where the data is distributed to different areas and remote users. Replication is used for processing client requests by deployment of several machines. Boosting system performance along with increased fault tolerance and availability are the key features provided by replication. Balancing load and the faster processing of requests can be handled by combining replication with distribution of client requests over low load servers. Moreover, by mirroring distributing placement of servers on different parts of the network, the latency induced by the communication between client and server can be reduced. Different approaches have been used for the replication of internet services, such that the whole system behaving as one powerful service with client requests are being transparently forwarded to the available servers. These approaches include server-side redirection (Rajaretnam et al., 2016; Andresen et al., 1996; Shahriari, Biglarbegian, Melek, & Kurian, 2016; Aversa & Bestavros, 2000), client-side redirection (Baentsch, Baum, Molter, Rothkugel, & Sturm, 1997; Yoshikawa et al., 1997) DNS redirection (Beck & Moore, 1998; Mansouri, 2016; Peng, 2004) and router redirection (Hunt et al., 1998). Selection of server also undergoes a set of criteria's which are dependent upon different performance parameters and dissemination of information regarding any server's status (Amir, Peterson, & Shaw, 1998; Loukopoulos, Ahmad, & Papadias, 2002; Crovella & Carter, 1995; Cardellini, Colajanni, & Yu, 1999; Shahriari et al., 2016). Cache is also considered as a special case of replication where only some part of the data object is stored to the servers in the system. Under specific conditions, the cache replacement algorithms can be classified as an online cache replacement, greedy cache replacement and distributed cache replacement, (Shahriari et al., 2016; Dowdy & Foster, 1982). A client-side redirection policy is thus defined as a forwarded client request which results in a cache miss to the server.

1.4 Replication and scheduling

For a system to keep working properly, replication and scheduling are very important. Approximately, terabytes to petabytes of data may be requested in a day. The failure to schedule the jobs properly leads to decreased availability of resources, hence, causing inefficient management of the data set of such large scale. Therefore, a small mistake can cause major breakdown when compared to the design of the data set. Life-critical systems and high availability requirements thus highly requires replication and scheduling. We~~,~~ hereby consider replication and scheduling strategies in ensuring lower and minimized network cost. Replication actually works by reducing the average time of job execution and provides long time optimization solution. A scheduling policy needs to work efficiently with minimum network bandwidth consumption, handling availability of resources. Multiple requests may occur at the same time during system operation which is seen as geographically dispersed under different distributed systems. Resource discovery, matchmaking and job execution are three main steps of scheduling when studied under complex systems. A global search of available resources is done by schedulers considering the restrictions and the history profile of the resources after which the best available resource is selected based on different parameters. The job is then executed and data sets are replicated by the commands generated by the scheduler (Mansouri & Dastghaibyfard, 2013). Running long source codes over thousands of machines that are combined together to form a single distributed system offers lesser reliability. We therefore require, reduced amount of data transfer among the nodes, which in turn requires an effective scheduling mechanism (Rajaretnam et al., 2016; Souli-Jbali et al., 2015; Mansouri & Dastghaibyfard, 2013). Job scheduling and data replication, thus, holds a critically important place in such paradigm. Frequently requested, different types of data are still not handled by the growing storage size of computer systems. An optimized replication strategy is the only solution so far to handle such variety and size of data along with reduced time of transfer and deletion. Although, techniques such as replication of data and scheduling are applicable to distributed systems, these systems are not fully optimized.

1.5 Problem Statement

Replication accomplishes the performance and accessibility of distributed systems like video and web server networks. Replication schemes defines how to replicate data objects and on which node it is to be replicated, which stands as a major problem in this domain. Although, researchers such as [28, 29, 30, 31, 32, 33, 34, 48] have investigated on the challenges and proffer some solutions to replica placement problem (RPP).

The problem in their work is that their solutions do not schedule the data in such manner as to be able to minimize the access time of data, the execution costs as well as reduce transfers and deletions cost in a system. However, based on the challenges and drawbacks of the existing work, there is a need to develop an efficient scheduling technique which will be used for load balancing, while ensuring the existence of data replica for the job before scheduling. The proposed technique will schedule the job to those sites where replica is already present. There are two extra modules which ensures that the replica is already present where the job is going to be executed.

Therefore, the purpose of this research is to;

- Schedule the jobs such that they minimize the transfer and deletion of data and at last enhances the performance in terms of response time, mean job execution time while ensuring the existence of the replica.
- Simulate the proposed mechanism and compare our results with the existing approaches.

The inspiration of this work is to build up the replication and scheduling methodology that reduces the execution cost as well as effectively schedule the task with the amount of information required for each job. An optimum scheduling strategy can limit the exchange and erasure activities of record documents among computing nodes. Client request pattern changes in a data grid environment every time. Consider, for example a Video Server Framework where video demand changes daily. Therefore, it is not always appropriate to replicate and replace the latest files to meet the requirements of the nearest server. The jobs Scheduling to a suitable processing site is crucial because data exchange among processing sites extends the usage amount and access time of the data (Rajaretnam et al., 2016). Therefore, we have contemplated the replication and job scheduling alongside. Main influences of the work are:

1. We presented the consideration of scheduling strategies for jobs and data intensive applications in the data grid environment. Our work enhances the scheduling strategy of MDHR (Mansouri, 2014) and schedules the job considering the location of replica of the relevant job.
2. In this work, data replication and scheduling strategy are targeted. Our approach currently adopts two steps to minimize the access time of data and execution expense, that is (a) Scheduling the jobs to the most accessible processing sites where replica is already present, and (b) Minimize transmission time with respect to their network bandwidth consumption and load on the sites.
3. Simulation of a proposed work is done in GridSim Toolkit 5.2 and defines the architecture of a data grid that supports replications and scheduling of jobs. Extensive Broad simulations were carried out in order to examine in contrast to the execution cost and turnaround time with the data replication strategies.

1.6 Objectives

The objectives are as follows.

- Analyze and identify the limitations of existing scheduling mechanisms.
- Design a scheduling strategy that minimizes turnaround time, migration rate and response time along with load balancing.
- Propose a "dynamic replication aware load balanced scheduling in distributed environment DRALBS" Considering the location of replica at the time of job scheduling dynamically.
- Evaluate the proposed technique using the state-of-the-art replica aware scheduling approaches.

1.7 Soppe of Research

The scope of this research is restricted to considering the issue of fault tolerance in grid distribute environment. Although our research has improved results significantly, there is still need to mature this strategy with a specific end goal to propel our insight around there.

The main limitation of this research is that it does not analyze the problem of the identification of a critical issue of failure in the existing scheduling model. This work only covers the replication job scheduling in a free failure scenario. Therefore, the lack of fault tolerance is one of the main issues related to replication and scheduling technique in distribute environment. This research can also be explained with the failure recovery strategy. For instance, fault may happen any time in processing nodes, and can result in exponential growth in implementation cost and response time. We assumed the scenario in which failure did not seem to occur which can also be beneficial in realizing the variant of the present issue. Inspecting the defects and implementing the failure resistance mechanism can slightly expand the execution time and the usage cost but in general the scheduling procedure was significantly enhanced.

1.8 Thesis Layout

The entire thesis comprises of six chapters and each chapter is divided into three main parts. These are: Introduction: to describe the objective of the chapter, Body containing the relevant data, Conclusion on the evaluation of the objective to be achieved in the whole chapter with a linkage to the next chapter. The thesis is arranged as follows;

The flow diagram and overview of the thesis is shown in Figure 1.3

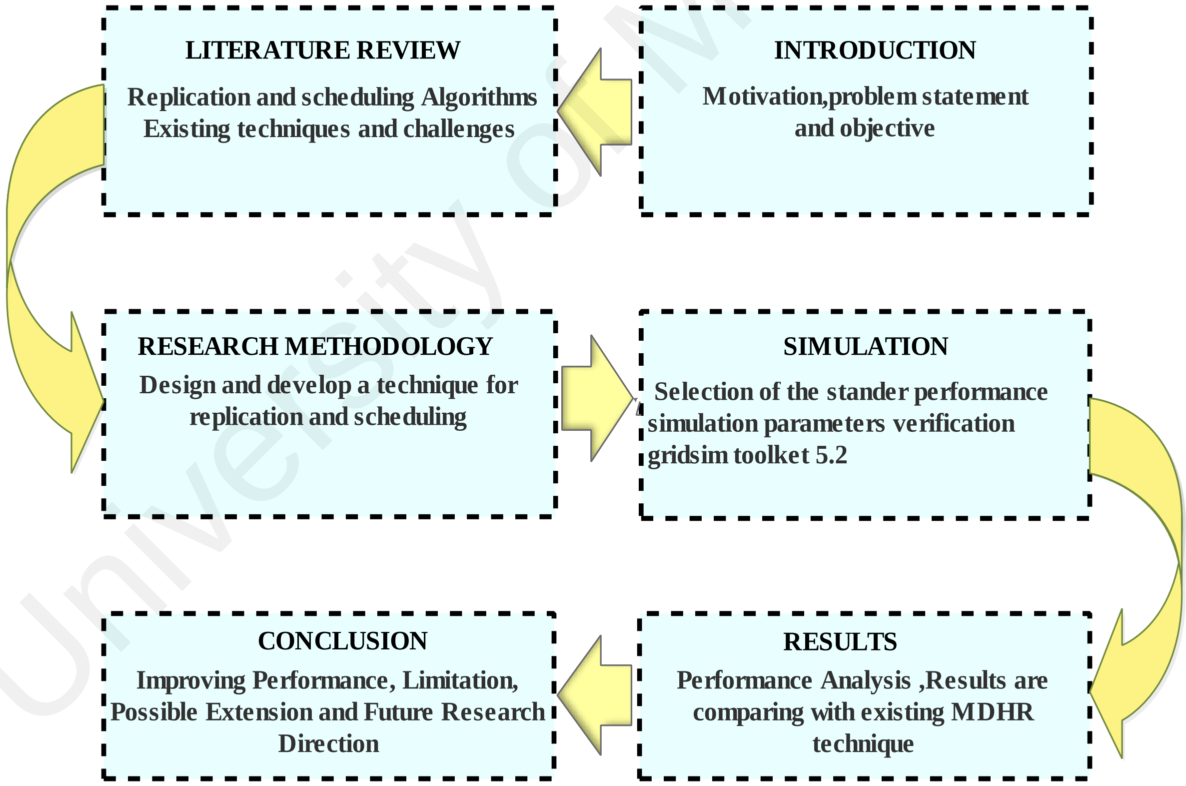


Figure 1.3 thesis layout

Chapter 2 Literature Review: This presents an extensive review and analyze scheduling algorithms for replication, categorizes those algorithms and compares them on the basis of performance, advantages and limitations. The challenges and open issues are also identified in the existing techniques for improving the scheduling replication. Lastly, we advocate the development of effective replica aware scheduling algorithm. Chapter 3 Research methodology: This chapter presents the research framework and a proposed model on the basis of problem analysis in the existing scheduling techniques. It also expresses the bandwidth between regions and Local Area Networks (LAN) connected to each other in the regions and the sites which are connected together in LAN's having high bandwidth as compared to regions and LAN. It further explains in detail the architecture of the proposed model, and explain the working procedure of each module for getting information from sites, and based on these information, the scheduling techniques schedule job according to its requirements. Two extra modules were added which are, replica manager and load manager, for replica location and load balancing respectively and the scheduling technique builds upon replica location. Chapter 4 Proposed a scheduling strategy; This chapter elaborates our proposed Replica Aware Scheduling Strategies to assign a job to a suitable node that already has the required data for jobs in a load balancing manner and improves the overall network performance. The chapter provides a clear understanding of job scheduling in a distributed environment to overcome the limitation of existing techniques, especially in the recent MDHR replication technique by using our proposed technique. Moreover, concluding remarks are also provided at the end.

Chapter 5: Result and Discussions; this chapter presents the experimental research, i.e. simulation results of the proposed scheduling techniques. The result clearly shows that the performance of the proposed algorithms is more encouraging as compared to MDHR. The conducted experimental research demonstrates the encouraging results in terms of response time, migration rate, latency, bandwidth and network consumption along with load balancing. Finally, the main objectives of the chapter is the summary of the results in tabular forms for easy and smooth understanding. Chapter 6 concludes the thesis by reporting on the reassessment of the research objectives. It elucidates the main finding of the study, highlights the significance of the proposed technique and also states limitations and the possible future extensions.

CHAPTER 2: LITREATURE REVIEW

2.1 Introduction

The chapter begins with a simplified overview of the replication techniques and the issues regarding the replication location during the scheduling procedure in a distributed environment. Computing grid is a sort of dispersed computing system. Grid network is mainly partitioned into two sections; grid computing and data grid. Computational grids utilizes computationally escalated applications that needs a little measures of information, while data grids manage those applications which needs study and analysis of a big record of data . Replication of Data is also a very important step for optimizing the large amount of data files by replica creation of data in many sites of the grid. Data replication is viewed as a vital boost mechanism in data grids, however there are some issues in decision making. The two important problems are replica placement and replica selection. The problems are explained in further details as the classification of replication along with the basic concepts being provided for clear understanding of replications and scheduling techniques. This chapter also reviews the types of replication techniques and challenges of the data replication scheduling techniques. It also presents the state-of-the-art scheduling solution in grid with consideration from the beginning to the last. We classify these techniques into static and dynamic. We deeply analyzed each techniques to identify the advantages, limitations and performance of every one and the exact problem addressed by a particular technique with simulation of the corresponding techniques. The dynamic feature of a grid distributed environment provides a potential of scheduling in grid. Furthermore grid as a distributed network environment provides sharing of resources and execution of big data. The critical discussion extends the knowledge of replication and scheduling for rapid access and smooth execution along with network load balancing.

The remainder of this chapter is structured as follows; Section 2.2 represents a comprehensive overview of Distributed environment. Section 2.3 explains the procedure of Replication and scheduling technique. Section 2.4 describes the classification of replica techniques into two main types: static or dynamic. Section 2.5 describes the Challenges of the Replication process. We are faced with two problems in replication, replica placement and replica selection. In subsection 2.5.1 we present the replica placement problem as to where to deploy the selected server. Section 2.5.2 presents the selection of the best replica site to easily access users. Section 2.6 presents the challenge in replication selection and placement techniques. In Section 2.7, the existing replication and scheduling techniques are described in more details. Section 2.8 discusses the comparisons of these techniques and in the last section 2.9, we provide the concluding remarks.

2.2 Comprehensive Overview Of Distributed Environment

Grid computing is a kind of distributed computing system that gives access to various computational resources which are shared by different organizations, in order to create an integrated powerful virtual computer. The grid consists mainly of two parts, namely: data grid and computational grid. Computational Grid is utilized for computationally escalated applications that solves complex, time consuming computational problems and relatively requires less measures of information. Furthermore, it is a wide region distributed computing environment that empowers sharing, choice, and combination of geographically distributed resources in various positions. The data grid comprises of a series of nodes, every one of which contains multiple computing, storage, and networking resources. The techniques of replication scheduling is one of the key factors affecting the data grid performance by replicating data from geographically dispersed records. A successful scheduling technique of job assignments is required to proficiently utilize accessible resources.

2.3 Replication and scheduling

Replication is the way toward making numerous copies of records on the sites of the grid network. Replicas of data are created to upgrade the accessibility of information, stack adjusting among storage elements(SE) components, increases data availability, decreases system execution time and to give easy approach to data files in a wide distributed environment, where, disappointment will probably happen. In an event whereby a copy of record crashes, other duplicates are made accessible in grid sites (Grace & animegalai, 2014). However, irrational replication can result in excessive use of system resources and ultimately the degradation of data access latency (Wang, Yao, Xu, & Pan, 2017). An unreasonable replication strategy significantly increases data access latency by interrupting the operation while waiting for the data to be scheduled to another region. Meanwhile, the node on which the job is ran is deactivated, degrading resource utilization. Therefore, the replication strategy must carefully consider data scheduling issues (Wang et al., 2017). For the purpose of more effective utilization of resources and network bandwidth, there is need to pay special attention to replication scheduling. Scheduling is the policy of assigning resources to users in time. Generally, in scheduling issues, three components must be explicitly specified at each time which are: resources, scheduling requirement and load.

Therefore, to reduce the migrating rate of data, we need to design a viable scheduling technique to assign jobs to those sites which have required data. The scheduler should be able to minimize the exchange of data file among grid sites by assigning job to a more suitable site which, contains most of the requested data. Therefore, replication and effective scheduling is one of the best solutions for large scale applications that are geographically distributed over a wide area such as data grid. The initial thought of duplication is to put the file near to a user in order to make an effective and quick access (Mansouri et al., 2013). Two methods are used for the optimization of data replication namely: short-term optimization and long-term optimization. Short-term optimization can be accomplished by static replication while, long-term optimization can be attained by dynamic replication. A viable job scheduling is a challenging research issue (Mansouri, 2014).

2.4 Classification of Data Replication Techniques

The replication are categorized into two sorts of mechanisms of which one is static and the other is dynamic replication (Mansouri, 2016). In a static replication, the quantity of copies and the site where replicas ought to be set are choosed statically at the time of setting up the grid which means, it creates and manages replicas manually. The static replication systems are easy to actualize, however they are not being utilized on an extensive scale since they don't bolster information replication during the job execution. In other words, it does not adopt or change the dynamic behaviour of grid users. Static replication techniques have the benefits of quick job scheduling and reduced overhead (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013; Andresen et al., 1996).

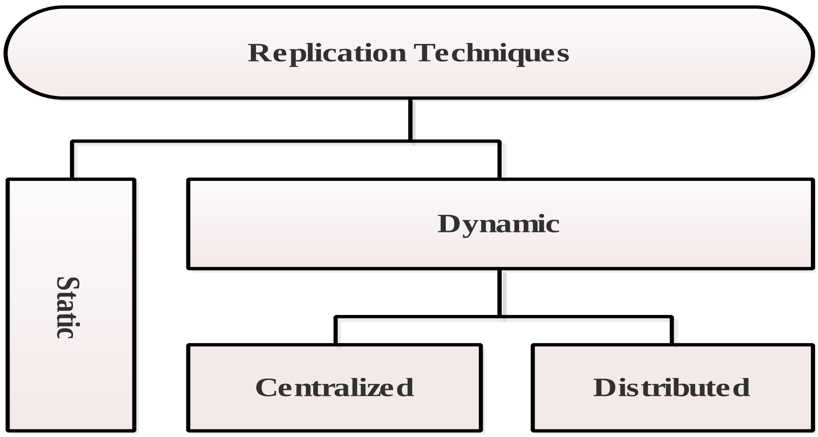


Figure 2.1 Classifications of Data Replication Technique

The Dynamic replication technique can adjust changes in view client demand profile, bandwidth, and storing volume capability. Dynamic replication techniques can make decisions in a wise manner for placing a file on site depending on the storage space and site availability. New copies of files are automatically generated on different locations depending on the dynamic behaviour of the data grid. In a centralized technique, copies are generated just on the head or top of a node, on the other hand, in the case of distributed techniques, copies are generated in some selective sites, to the top or head node. Due to the dynamic behaviour of grid users, wherein a user can enter and leave the grid environment at any time, the achievement of maximum data availability becomes hard. But there is no doubt that dynamic scheduling is better than static replication with regards to the reliability and scalability in grid environment.

2.5 Challenges of Replication scheduling procedure

The replica selecting and placing methods of replication procedure are linked together to select the best site and place copy on site in data grid. As the quantity of computing big volume applications in a distributed environment increases, replication strategies are broadly used to build the accessibility of information, enhancing execution of query latency and load adjustments in the data grid. In data replication technique, duplicate copies of data files can be placed in sites and can be used during execution of the job. Grid users can request and leave the system in a dynamic grid environment at any time. Replica selection and placement techniques depends on the data grid design. Controlling the dynamic nature of grid architecture, making a decision of copy space for saving, placement, and replication cost problem selection are big chalenges which has direct effect on the grid performance.

The architectures that are employed in grid are as follows: multi-tier (Little & Venkatesh, 1995), graph topology (Pitoura, Ntarmos, & Triantafillou, 2012; Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013), Hierarchical (Hunt et al., 1998; Beck & Moore, 1998; Huang & Peng, 2013; Fei, Bhattacharjee, Zegura, & Ammar, 1998) and peer-to-peer (Yoshikawa et al., 1997; Narendran, Rangarajan, & Yajnik, 2000; Loukopoulos, Lampsas, & Ahmad, 2005). The decision of Replica assignment includes many important demands like, the time and place of data copy, and the found required relicated file, as well as checking the storage availability of the site before placing copy of a file. In the data copy placement method of utilization, it first checks the available storing capacity for replica. But having checked that, the advantages of replication should be greater than replication cost. The advantages of replication consist of reliability, availability, scalability, felexibility, which improves the system execution. Subsequently, data replicating leads to many sites having the same copy. The best replica is chosen by replica slection method . The role of response time is important in the replica selection and make span time of the job.

Data replication is a productive technique of the copy and distribution of data to different locations in a geographically distributed network (Souli-Jbali et al., 2015; K. A. Kumar, Deshpande, & Khuller, 2013). It is a typical strategy for facilitating frequent access to large data in a distributed environment and providing high-level data availability, reliability and improved fault tolerance (Grace & Manimegalai, 2014). For this reason, this method is used for the data management of databases in a distributed system. Recently, data grid researchers have been discussing on a number of techniques to create multiple file copies to speed up file access times and put them in more than one place in a distributed network. Therefore, the replication procedure is one of the main considerations influencing the execution of data networks by replicating data in dispersed data stores. This problem was investigated under the scope of the replica placement problem (RPP) (Litke, Skoutas, Tserpes, & Varvarigou, 2007; Ali-Eldin & El-Ansary, 2011; Al-Shayeji, Rajesh, Alsarraf, & Alsuwaid, 2010; Du, Hu, Chen, Cheng, &Wang, 2011; Eslami & Haghighat,2012; Huang & Peng, 2013; SouliJbali et al., 2015; K. A. Kumar et al., 2013).

There are three most imperative issues in data replication (i) Replica Placement, puts replica of the desired data on the appropriate node (ii) Replica selection, chooses the best replica site for easy access of the required data. (iii) Replica Management: create, delete, modify replica overwrite.

2.5.1 Replica Placement

To organize a replicated service, it must first be decided on where to deploy the selected server. There are many algorithms and systems models present, which are helping to increase and capture the performance related to this problem. In the next section, we grouped research studies in this area based on their association to illustrious classical optimization problems, their implementation cost and the cost of replica placement and earlier work prepared for replica placement problems.

Placing the replica to an appropriate place is itself a problem that has been researched somewhat broadly, but the number of complications has been projected and seen in this area. In the work of (Souli-Jbali et al., 2015), they have measured minimizing the replica–client distance as the optimization function that was focusing on replica placement to the node that is nearest to the replica server, but this replica placement can be weak to weight inequities. While in the work by (N. Kumar & Kim, 2013), their main goal was load balancing, also others like (Ali-Eldin & El-Ansary, 2011: Al-Shayeji et al., 2010; Du et al., 2011), have focused on reading access cost, and (Jamin et al., 2000) focused on client traffic which concentrates on both update and reading of the requests. With the issue of replica placement, other issues in combination with this issue discussed are (Jamin, Jin, Kurc, Raz, & Shavitt, 2001) processing capacity, (Narendran et al., 2000) bandwidth and also server storage capacity (Apers, 1988; Little & Venkatesh, 1995). In this research, we have used the same model as discussed in (N. Kumar & Kim, 2013; Loukopoulos, Tziritas, Lampsas, & Lalis, 2007; Tziritas, Loukopoulos, Lampsas, & Lalis, 2008; Jamin et al., 2000). Our proposed model, DRALBS (Replication Aware Load Balanced Scheduling) defines additional parameters in order to improve the problem that we have focused on. A vast number of research literature is available on the problem related to scheduling, it comprises of many fields, including parallel computing in task scheduling of multiprocessor systems (Pitoura et al., 2012; Zaman & Grosu, 2011) and also in automobile routing/sequencing prospective (Sun, Gao, Yang, & Jiang, 2011; Sun et al., 2011). Solutions for these problems consist of a number of algorithms which includes branch and bound algorithms, genetic algorithms and randomized algorithms (Sun et al., 2011). Replication does not not only reduces the file transferring and deletions, but also improves the bandwidth consumption in data grids (Sun et al., 2011). Data Grids provide communications for the processing of access, managing of large data sets, and transferring of data sets which are stored in distributed repositories (Khan & Ahmad, 2010; Rajaretnam et al., 2016). It is used, to emphasize on fulfilling the necessities of scientific collaborations, which shares large amount of data and is of necessity to analyze and share the data and results among collaborators. Examples of such applications are commonly: climate simulation (Khan & Ahmad, 2010; Rajaretnam et al., 2016) astronomy (Grace & Manimegalai, 2014) and high energy physics. Dealing with such large amount of data can cause geographically spread challenges which includes how scheduling strategy efficiently processes each and every job and the means of the replication process performance. Data replication is used mostly in a distributed system to highlight the issues of efficiency and reliability of the data retrieving and accessing process in wide areas (Jamin et al., 2000). This is an effective method that is used to improve the efficiency of the network performance by reducing the client server communications (Du et al., 2011). Replication of data objects on different servers and then transferring of a client request to that server is the main issue and most of the work has been carried out already on the discovery of replicated server, replica placement strategy, maintaining consistency on replica placement and scheduling the replica placement (Souli-Jbali et al., 2015; Abdi & Hashemi, 2015). Appreciable amount of work have been done to increase the performance in a distributed network environment and some works also focus on the scheduling of jobs and tasks before the placement of replicas. This approach is used to reduce the implementation cost of migration from the old state to the current state of the system and it is also used to process the end user's request frequently. In the work of (Mansouri et al., 2013), they aggravated on minimizing the implementation cost while updating the replica server from the old to a new state at the centre of the user requirements, without having to consider any scheduling of task and data. Regarding the example of a video server system in which the demand of video changes per day, satisfying the essential demand from the nearest server all the time, is not a suitable method. Due to the fact that movement of the data between several sites consumes much time, job scheduling for the suitable server site is essential (Litke et al., 2007). The scheduler assigns jobs to those sites where the required data are present. The job execution process gets the required information without any communication postponement for receiving data from other sites. DRALBS is the scheduling and replication strategy that improves job access time and also minimizes implementation costs. Previous research papers conclude that not much work has been done on applying both of the strategies concurrently.

2.5.2 Replica Selection

Replica selection is also very important in reducing response time and migrations. Replica selection method chooses the best site of replica for the client to frequently gain access to demanded information at the time of execution. At the point when a site requires to get a copy of data, it questions the store of meta-data with the required data features as input. The archive of meta-data has legitimate document name and cooperative-wanted-attributes. The site first checks the name of the logical data which comprises of the essential information of a client. After this, it then utilizes the area of the replica facility to find all nodes of the replica that covers the data cases against requiring coherent record. Lastly, the replica selection method picks an ideal copy site for the file fetch bring-in light of job limits.

2.5.3 Replica Management

Replica management is one of the important part of a distributed data grid. The basic functionalities of replica management are the creation of replica, deletion of replica and change overwrite of a replica. Inside the Data Grid and also in distributed computing communities, multiple projects resolve issues related to replica management. One way to use replica management machanisim, is to replicate a pre-defined collection or set of data to another Grid node. A resource broker can communicate with the replica management system that optimises the jobs scheduling and access. According to the management point of view, the issues that are identified with replica management in a data grid distributed environment are: consistency of replication, reliability and lifetime management of replica.

2.6 Existing literature on Replication and Scheduling strategies

Various research works are present on the data placement problem, (Grace & Manimegalai, 2014; Rajaretnam et al., 2016) presents the work of the placement of data files on the number of servers. Considering an example of movies demand in a distributed environment by end users, in this example, if there exists a single copy of a movie, the respective server might be overloaded from user requests; hence the replication of movie files must be present on many servers. Changing and replacing the replicated data with the end user preferences is not an easy task, this may cause increase in the transition cost of moving the data from one replica server to another but may also delay the fulfilment of end user requests. Many recent research work have presented job scheduling problem with the replication of data concurrently. The reduction of time for job execution is the purpose of dynamic replication and scheduling mechanisim in distributed surroundings. However, scientific applications comprise of enormous information data in a system of grid. There will be some restrictions of the number of files to be saved on each computing node. If the computing site is occupied with replicas and no space is available for new replicas, some of them are deleted from the site in order to store new replicas. Intial work on dynamic data replication in the data grid has been presented by (Pitoura et al., 2012) of which they proposed 6 replication strategies, that includes: (1) No Replication, (2) Plain, (3) Cascading, (4) Best Client, (5) Cascading, (6) Cascading plus Cashing and Fast Spread. There are three types of data localities, namely: (1) temporal locality, (2) geographical locality, and (3) spatial locality. Temporal locality is defined as: the situation wherby a file that is accessed recently is most probable to be requested again soon. Whereas geographical data locality is the number of files to be recently accessed by a user which are most suitable to be accessed by nearby users as well. Spatial data locality is the file accesses which are recently accessed and are more probable to be required in the near future. These techniques were examined with multiple data designs: (a) no locality, (b) access of data with a little influence of temporal locality, (c) data connection with a slight influence of geographical and temporal locality. Simulation result demonstrates that a different pattern of accesses needs different replication techniques. The study concluded that fast spread indicates predictable execution through different access designs. In the works of (Grace & Manimegalai, 2014; Rajaretnam et al., 2016), they presented a data placement problem with uniform ratio servers. Significant efforts have been carried out in replica placement under different circumstances, such as video, web servers and content distribution networks (CDNs) (Peng, 2004; Apers, 1988). Dealing with data management issues, which includes issues such as replicas creating, distributing, accesseing and updating replica of object, is crucial to the achievement of the above systems. Replica placement problem (RPP) also referred to as data assignment problem (Vashisht et al., 2014; Little & Venkatesh, 1995) thinks about the data management. On the other hand, RPP presumes just a settled framework state and cannot deal with frequent change management of the replication strategy. In the study of (Little & Venkatesh, 1995), Continuous Replica Placement Problem (CRPP) allows more successive updates, but without taking into consideration of the cost factor. Also works of researchers such as (Grace & Manimegalai, 2014; Rajaretnam et al., 2016) on the Replica Transfer Scheduling Problem (RTSP) considered the data replication frequently, as well as handled the cost minimization issue while exchanging information from one node to another. In addition, RTSP intends not just in minimizing the cost of replica transfers as a substitute of meeting the time criteria, but also on the scheduling of jobs and tasks needed in RTSP. In another group of studies by (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013) the focus of research was on a technique called "Bandwidth Hierarchy Replication (BHR)" which is utilized to diminish tentrance time by increasing network-based localities and escaping the network blocking. This strategy divides the sites into a number of regions in which the network bandwidth within the area is lesser as compared to the bandwidth inside the area. This is because the fetching time of the needed file that lies within the similar area is less. This

BHS technique still has two lacking needs, the first is, if the replica is present inside the region, it concludes, and, second, instead of appropriate sites, files that are replicated are located in any of the desired nodes. This BHR technique works well in cases whereby the storage limit of the node is low.

2.6.1 Dynamic Hierarchical Replication Algorithm (DHR)

"Dynamic Hierarchical Replication" (DHR) technique first checks the replica of data feasibility. If the size of the requested data file is more than node SE size, then the data will be used remotely. If it is in the replica candidate list, then the data will choose the site having the greatest bandwidth by the requesting node of the grid that has a minimum ratio of demands. DHR ensures data copy in BS (maximum accesses number of a specific data copy). If the required volume of the best Storage Element is equal to or larger than the required data size, then it will replicate the files, otherwise certain files will be removed. It removes the data files with small size of the files that are present in the local grid region and LAN(Mansouri & Dastghaibyfard, 2014). DHR promote some changes in the performance of mean job time execution, but still it demonstrates two inadequacies in the scheduling policy (Mansouri et al., 2013). It uses bandwidth parameter and LRU policy for replica placement that erases some important record that could not be accessible in the nearest zone and might be required soon which causes an increase in the data rate of transferring and consequently reduces the time of execution. Another drawback of this technique is in the selection procedure, that is, the prediction of response time on the basis of data transfer time and number of requests which has an insufficient parameter efficient scheduling strategy. Response time performs an important role in the replica selection and turnaround time of a job.

2.6.2 Enhanced Dynamic Hierarchical Replication (EDHR)

The "EDHR " technique is designed to improve the performance of HDR. This technique has basically three categories: The selection of replica, placement of replica and Replica data management procedures. In the selection procedure, the task is assigned to a certain node and the mandatory record files are sent to some other nodes to become new replicas. There is a significant advantage in choosing the best replica, when various resources of the nodes are having replicas of the same file. The response time is the main parameter that gives impact to the selection of the replica and the processing time of the job as well. Each storage space element (SE) has multiple requests simultaneously and the storage has to respond to a single request concurrently, so the requests have to wait in line. The (EDHR) technique chooses the best-replica position in a somewhat lesser execution time that could be computed by following exchange information time and quantity demands that will be waiting in the request of the storage of the site in waiting state. In case of choosing replica, check the existence of data file, if it is present in the same LAN, then a list of the replica candidates will be generated and the resources of the node will be chosen with minimum demands. If it cannot find record in "LAN", then EDHR will look further in the grid region of the distributed environment. However, if the record is already present in an existing grid area, then a replica candidates record shall be generated and a copy of data with least demands shall be chosen. Alternatively, it will generate a record/array of those data copies which are present in some different areas, and by using this record, it will select the replica with least demands. Placement of replica: The EDHR technique creates the data replica in the Best Storage Element (BSE). To choose the BSE, EDHR searches for SE with least Value in SE (VSE) in the demanded area. In the estimation of VSE, the number of replica is requested and lastly the accessing of the replica are made, nevertheless, these values are useful as they give the probability of the need to request the replica again (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013). Replica data management technique: If the storage volume for the required data is insuficient in the BSE, then the chosen data record is copied. Moreover, if the data record is present in the LAN of a local grid, then the data record should be accessed from their own place. Now, if the required file does not exsist in the same LAN and the storage volume for replication is not sufficient to save the file in the same node of LAN, then delete files as per user requirements.

2.6.3 Weighted Scheduling Strategy (WSS)

The technique of “Weighted Scheduling Strategy (WSS)” finds the “Best Region (BR)”. This indicates the area with maximum number of the demanded data for a task (files extent). WSS calculates RC for every grid region and selects the BR, (The grid area has a least “RC value”). The “Combined Cost- CC” for every node inside BR is calculated and job is assigned to those resources of the node having low “Minimum Combined Cost (MCC)”. For this reason, the WSS will not loop each of the resources of the site to locate the best site with minimum cost. It only considers the jobs strength of list/queue in wait and the computing capacity of the grid site (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013).

2.6.4 Modified Dynamic Hierarchical Replication (MDHR)

Data replication is a very important technique for improving response time in grid applications. In replication, multiple file copies are been generated and placed at many locations to reduce the file fetch times. MDHR technique is a new improved technique of Dynamic Hierarchical Replication (DHR). However, replication of the data file will be utilized intelligently as the space for storage is limited for each site of the grid. So, it is good to set up a technique for the data replication placement/replacement task. The drawback of DHR strategy in replica placement procedure is covered by MDHR because it keeps the most important replica instead of less important replica. The MDHR overwrites the replicas of data which relies on the last access time it was approached, how many times it was accessed, and the size of the storage replica file. It also chooses the best replica site from the entire replica files depending on the quick response time which can be found by considering the data exchange time of the data, the delay in the accesses, the replicas demands currently in the queue of the storage, the average distance among sites and CPU process capability (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013). Furthermore, MDHR checks the available storage capacity for new replica, if it is not sufficient, then it will remove those record that have minimum transfer time i.e. the record that occurs in the LAN and the local region. However, if the storage space for new replica is still not enough, then it makes decision on the basis of last time request of replica, number of access and replica data size. The other problem of DHR is the selection of best replica among many replicas of the system. The MDHR methodology picks the best area of replica for the client's execution, in light of new parameters besides the information exchange time, namely, latency for storage queues, storage access time and distance between sites. Although, MDHR enhances the performance of DHR strategy, it still needs to improve because during the scheduling method, it does not consider the location of replica on a specific site. The access time relies upon the replica location and job scheduling where it was submitted for execution. Moreover, scheduling is important for bandwidth consumption and load balance, a proper scheduling procedure can assign a job to a suitable node where replica is already present. Otherwise the computational resources will not be utilized properly if the jobs are not scheduled accordingly, thereby making the computational resources of the distributed system to be useless (Mansouri et al., 2013; Mansouri & Dastghaibyfard, 2014).

2.7 Gap in Research

In this chapter, we have examined the problems and have closely investigated the state-of -the-art "Modified Dynamic Hierarchical Replication (MDHR)" scheme. The chapter also presented existing solutions that tackles different issues in our research domain. In the context of the gap, we specifically discussed the present issues related to the existing MDHR. Tables 2.1 presents the existing work. The drawbacks and challenges of each work are also stated in the table. The motivation of the research is from these drawbacks and challenges of the existing approaches in the grid computing environment.

2.8 Comparison of replication and Scheduling Techniques:

Table 2-1: Comparison of several replication and Scheduling Techniques

Title/ author	Technique	Advantage	Limitation	Summary
Combination of data replication and scheduling algorithm for improving data availability in Data Grids. (**Najme Mansouri)**	Job scheduling algorithm called Combined Scheduling Strategy (CSS) and Modified Dynamic Hierarchical Replication Algorithm (MDHRA)	CSS uses a hierarchical grid structure and minimize the overhead of searching for the suitable site. MDHRA strategy keeps only the valuable replicas while the other less important replicas are replaced. Increase locality, that improves file	It ignores dynamic replica maintenance issues such as replica consistency. The Qos factors do not consider during dynamic replication along with service enforcement and client related needs.	In this research, two strategies are proposed, first CSS that uses hierarchical scheduling to reduce the search time for an appropriate computing node. It considers the number of jobs waiting in queue, the location of required data for the job and the computing

		access time. MDHRA strategy selects the best replica location for the users' running jobs by considering new parameters besides the data transfer time, namely, the storage access latency, waiting time in the storage queue and distance between nodes		capacity of sites. Second a dynamic data replication strategy, MDHRA that improves file access time. This strategy is an enhanced version of DHR strategy.
Comparative Models of the File Assignment Problem (LAWRENCE W.)	Replica Placement Problem (RPP). File assignment problem" (FAP)	Considers the data management. RPP presumes only a fixed system state, and cannot manage the frequent changes of a replication scheme.	RPP presumes only a fixed system state, and cannot manage the frequent changes of a replication scheme.	The author point out the critical design problem of computer systems is that of assigning files to possibly different nodes in a computer network for query/update /execution purposes; this is FAP. This paper attempts to

				classify, notationally unify, and summarize the current FAP results. That Indicates new fields of FAP research.
Continuous Replica Placement Schemes in Distributed Systems	Continuous Replica Placement Problem (CRPP)	This technique considered frequent updating requirements. CRPP embeds a scheduling problem to facilitate the proposed mechanism.	It does not consider the cost factor. It needs extended version, to evaluate the technique in other distributed scenarios such as distributed video servers.	This research investigated the case of CRPP. It includes an underlying scheduling task that complicates algorithmic design. Nevertheless, they propose suitable heuristics to solve it.
Job Super scheduler Architecture and Performance in Computational Grid Environments (Hongzhang Shan, Leonid Oliker)	Grid super scheduler architecture and three distributed job migration algorithms	The superscheduling improve the performance in terms of average waiting time and average response time, that can be achieved via	These studies do not consider the control and the proper management of data replicas and scheduling jobs properly. This work	The work described that the three local scheduling approaches together with a superscheduler improved overall performance compared with

		smart	does not address some important issues such as job migration overhead, grid network costs, super scheduler scalability, fault tolerance, multiple-resource requirements, and architectural heterogeneity	locally isolated systems. The results demonstrated that superscheduling can deliver substantial performance gains
Dynamic Data Grid Replication Strategy Based on Internet Hierarchy (Sang-Min Park)	Bandwidth hierarchy-based replication (BHR)	Reduce the time of accessing data and avoid any blockage of a network, by enhancing network level locality. The BHR technique works well in cases where the storage capacity is considered to be smaller.	BHR has two main shortcomings: (a) if a region contains the replica, it terminates; and (b) instead of selecting a few appropriate sites, replicas are placed on all the demanded	This replication technique improves access time of data and enhance network locality to avoid the network blockage. BHR gives good results in smaller storage capacity. It placed replica on all demanded site instead of

			sites.	some site and also terminates if the replica is already exists on a region these are the limitation of this procedure.
Complete and fragmented replica selection and retrieval in Data Grids. (Ruay-Shiung Chang)	Emergent methodology called fragmented replicas	Save the storage space wasted in storing unused data by store only some partial contents needed locally.	There is no work on the problem of fragmented replica updates that is important for the grid environment. If the fragmented replica ware not contiguous then, The algorithm for retrieval would be more complicated.	The work is done on the concept of fragmented replicas and propose algorithms for their retrieval. That is, when doing replication, a site can store only some partial contents needed locally. It can greatly save the storage space wasted in storing unused data. It also proposed a block mapping procedure to determine the distribution of blocks in every

				available server for later replica retrieval. According to this procedure, a server can provide its available partial replica contents for other members in the Grid system to access. On the other hand, a client can retrieve a fragmented replica directly by using the block mapping procedure
Job scheduling and data replication on data grids (Ruay-Shiung Chang)	Hierarchical cluster scheduling strategy (HCS) and the hierarchical replication strategy (HRS)	This Algorithm used to improve the efficiency rate of accessing the data in the grid and reduce the searching time for the appropriate computing node, mean job execution time	In HRS a replica is located in a local cluster and does not consider network heterogeneity and location of replica on a proper site at the time of job	The author addressed the problem of data movement operations in a cluster grid environment. To achieve good network bandwidth utilization and reduce data access time, by

		can be minimized, the network is used more effectively and storage space is saved.	scheduling	consider the inter-cluster communications cost. They propose a job scheduling policy (HCS) that considers not only computational capability and data location, but also cluster information, and a dynamic replica optimization strategy (HRS) where the nearby data has a higher priority to access than generating new replicas
Dynamic replication in a data grid using a Modified BHR Region Based Algorithm (K. Sashi)	Modified BHR	Replicates the most accessed data files that are supposed to be requested again in a near future. Improved data accessibility and minimize	The modified BHR strategy has a major drawback in that it replicates the file to the best site by searching all the sites	The Modified BHR Region Based Replication Algorithm increases the data availability by replicating files within the region to the

		job execution time .	individually.	region header and also storing them in the site where the file has been accessed frequently. The unnecessary replication is reduced by using replica on a particular site not on all sites Therefore, this algorithm, mean job execution time can be minimized, the network is used more effectively and storage space is saved.
Enhanced Dynamic Hierarchical Replication and Weighted Scheduling Strategy in Data Grid (Najme Mansouri)	Two strategies are proposed, weighted scheduling strategy (WSS) and a dynamic data replication strategy called "enhanced	WSS reduces the search time for a suitable computing site. EDHR improve data access time.	No touch to dynamic replica maintenance issues such as replica consistency.	Two strategies are proposed in this work. (WSS), which is a scheduling strategy that aims to reduce the search time for a suitable computing site. WSS examines the sets of

	dynamic hierarchical replication" (EDHR),			requested jobs waiting for their turn in a queue, the place to where the requested data should be replicated, and the storage capacity of a computing site. The second EDHR, which advances file access time. EDHR strategy is an enhanced version of the dynamic hierarchical replication strategy. EDHR introduces a method to delete the files from the computing nodes whenever there is not enough space for the new.
Ensuring Cloud Data Reliability with Minimum Replication by	Cost-effective data-reliable PRCR	This approach manages the minimum required	This strategy ignores the location of the replicas in	Presented a cost-effective reliability management

Proactive Replica Checking (Wenhao Li, Yun Yang)	mechanism	number of replicas, in order to reduce the replication cost, and manages large amount of data in the cloud at the minimum cost of cloud storage space.	the cloud	mechanism (PRCR) based on a generalized data reliability model. It applies an innovative proactive replica checking approach to ensure the data reliability while the data can be maintained with the minimum number of replicas (serving as a cost effectiveness benchmark for evaluation), which is no more than two. Evaluation of PRCR has demonstrated that this mechanism is able to manage large amounts of data in the Cloud, significantly reduce the Cloud storage

				space consumption at a negligible overhead.
The Impact of the Implementation Cost of Replication in Data Grid Job Scheduling (Babar Nazir)	Centralized dynamic scheduling (CDS) and replica placement strategy (RPS).	It minimizes the job execution time and implementation cost.	The technique does not consider the fault tolerance in their procedure.	The centralized dynamic scheduling strategy-replica placement strategy (CDSS-RPS) is proposed.the proposed CDSS-RPS algorithm provides an improved scheduling and replication mechanism compared to the RTSP heuristics. Still need to improve in term of fault tolerance and compared with new new advanced techniques.
Hierarchical Data Replication and Service	Hierarchical data replication	This methodology Improved data	One of the main drawback in	A novel data replication method is

Monitoring Methods in a Scientific Data Grid (Weizhong Lu)	and service monitoring methodology The multi-agent and SNMP technique are used Hierarchical data replication and service monitoring methodology.	accessing efficiency and verified the effectiveness of the methods at the same time.	this system model is that a single point failure at the meta server can degred the system performance. And another darawback is the security lackage between agent and server.	presented, which takes the characteristics of nodes and network topology into consideration. The hierarchical data replication method makes full use of the bandwidth and memory space of storage nodes in local domains. It also has the merits of self-organization and load balancing. At the same time, also present a novel service monitoring system based on a multi-agent technology in a data grid environment to realize real-time service status collection and monitoring, by

				adopting popular techniques of multi-agent and SNMP.
Efficient Dynamic Replication Algorithm Using Agent for Data Grid. (**Priyanka Vashisht**)	Efficient Dynamic Replication Algorithm (EDRA) using agents for data grid, has been proposed and implemented.	The scheduling in data grid helps in reducing the data access time. The distribution of the load on the nodes of data grid is done evenly by considering scheduling parameters.	replicas considered in the experiment are read only which cause the replica onsistency ploblem. And degrad the over all performance of a system	EDRA addressed the issue one replica for each region in BHR. It place one replica on each subregion instead of region. The number of replica increases the avialibality of data. This algorithm checks the availability status of a particular node within region and takes the decision accordingly. Scheduling policy at master and head node increases the performance of EDRA and also

				has incorporated load balancing factor which improve jobs execution.

Techniques	Year	Description	execution Parameters	Advantages	Limitations	Simulation	performance
BHR	Park et a 2003	Replicates files which are likely to be used frequently within the region in near future.BHR keeps at most one copy of file in the region.	Total job execution time	1. BHR has good performance when capacity of storing element SE is small 2. decreases the data access time by maximizing network-level locality and avoiding network congestions	1. if replica exists within the region it terminates. 2. replicated files are placed in all the requested sites not the appropriate sites (replica location not consider) BHR uses the popularity of replicas at the cluster level not at the site level.	Optor Sim	BHR takes 30% lesser execution time than LRU and show good performance in small storage Element.
Modified BHR	Sashi and Thanamani 2011	Strategy which replicates a file that has been accessed most and it may also be used in the near future.	Total job execution time Increase locality.	it will search all sites to find the best one for storing the replica.	does not consider replica location at the site level	Optor Sim	Better than No Replication, LRU, LFU and BHR replication algorithms.
WSS	2011	First determines a Best Region (BR) the region that has most of the Requested files of the job (from a size point of view). Then find out combine cost then the job will be assigned to the site with Minimum Combined Cost (MCC).	WSS considers the number of jobs waiting in a queue, the location of required data for the job (from a size point of view) and computational capability.	WSS does not search all resources to find the best one that has the lowest cost. decrease total transfer time, and consequently the network traffic	does not consider site and replica location but focus on region and cost.	Optor Sim	better fthan LRU, LFU, BHR in term for cost and network traffic

Techniques	Year	Description	Metrics /Parameters	Advantages	Limitations	Simulation	performance
DHR	2012	Job is allocated to scheduler then Replica manager transfer all unviable required files. Only replicate those files that are not available in the near sites and delete files that have less transfer time. Selects best replica, Places replicas in appropriate sites that has the highest number of access for that particular replica	Mean job execution time	DHR is Better than BHR when available space for replication is not enough. Place file in an appropriate site Save storage space. prevents from creating unnecessary replicas	Replica Selection and replica replacement strategies in DHR Strategy are not very efficient, and have same performance when available space for replication is enough	Optor Sim	DHR outperforms the methods such as LRU, LFU, BHR and 3LHA..
EDHR	2013	It considers the frequency of requests of the replica and the last time the replica was requested For replica placement. EDHR places the replica in the best site of the requested region.	Mean job execution time varying by the storage size., bandwidth. varying size of files.	It uses the Economic Model to decide and delete those files that are not beneficial in the future and replaces them with files that are more beneficial in the future. It will also not search all sites to find the best one for storing the replica	EDHR selects the best site for placing the replica by considering the frequency of requests and the last time the replica was requested but not consider the location of replica to schedule jobs to a proper site.	Optor Sim,	EDHR Better than DHR and others such as LRU, LFU, BHR and MBHR.
MDHR	2014	MDHRA considers three factors for replacing replicas: (i) the last time the replica was requested (ii) Number of accesses (iii) size of replica. it will not delete those file that have a high transferring time.	Mean job execution time varying by the storage size., bandwidth.	improves Mean Job Time and Network Usage	MDHR Using LRU for replica management which may delete some valuable file in local region and may be needed and does not care about the location of replica at time of job scheduling	Optor Sim	13% better than LRU and BHR and other such as LFU BHR Modified BHR

Table 2.1 compares the existing approaches based on functional optimization and performance in grid computing.

Although much work has been conducted in this domain, only a few are on dynamic hierarchical replication. Existing solutions address different issues that ranges from improving mean job execution, and efficient network usage. It can be observed that all the existing approaches and implementation techniques discussed, focused primarily on how to speed the mean job execution (basic security properties for embedded devices). Also, they do not provide a robust solution against dynamic hierarchical replication challenges. Along this line, there is a need to develop a "Dynamic Replication Aware Load Balanced Scheduling in Distributed Environment" to enhance Modified Dynamic Hierarchical Replication (MDHR) that is based on replica location at the time of the job scheduling dynamically.

2.9 Conclusion

Replication of data is a very important step to optimize massive data files by creating replica in many nodes of the grid distributed environment. The storage capacity limitation of a traditional distributed system and utilization of resources in every region of the grid can be overcome by proper replication and scheduling policy. There are many techniques used to face replication scheduling issues. To meet the newly imposed replication scheduling requirements, this chapter presents a broad overview of replication scheduling in the grid distributed environment. We highlighted and analysed replica placement and replica selection issues with challenges that may affect the replica scheduling policies and also illustrated its main categories with a proposed corresponding compact solution. A discussion on the replication and scheduling techniques was also presented. A comprehensive survey and classification of existing replication scheduling techniques promote ways to schedule jobs properly by knowing the replica location. In the next chapter, we shall present our system model for replication and scheduling technique in a grid distributed environment.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter covers the explanation of the system model along with their services, targeting replica location to schedule jobs for execution on suitable sites. The main contribution as opposed to the schedule of the submitted jobs to appropriate sites is that it fulfils job requirement while decreasing the migration rate and execution time by formulating it in the algorithm. The scheduling procedure is discussed in more detail by providing a system architecture of a grid computing environment. The chapter also briefly discusses the problem of replica placement and schedule implementation to solve it by a proper scheduler method. Moreover, it highlights the fundamental concepts of mostly used terminologies to help readers gain clear understanding of the scheduling technique. Using this system model, we have formulated (algorithm format) the problem statements that are of importance to our work; the important parts of the replication scheduling such as replica placement and selection techniques. For comparative study, we choose one of the recent scheduling technique namely, the MDHR technique which was proposed by Najme Mansuri (Mansouri & Dastghaibyfard, 2014).

3.2 Research Framework

Section 3.2, describes the framework of our research which is based on the scheduling procedure of a job in a distributed environment. Scheduling plays a vital role while allocating job to the nodal devices containing replica to increase response time and reduce the execution time of a job. If the jobs are not scheduled properly on a suitable site, the computational resources will be wasted (Mansouri et al., 2013). Replication is the component that consequently duplicates index information starting with one Directory Server and then to the next. While utilizing replication, you can duplicate any index tree or subtree (put away in its own database) between servers, aside from the setup or checking data subtrees.

Replication empowers you to give a very accessible registry benefit, and to geologically appropriate your information.

In useful terms, replication brings the accompanying advantages:

1. **Higher performance and reduced response times:** Repeating index tracks to an area near your clients, you can boundlessly enhance registry response times.
2. **Local data management:** Replication enables you to claim and oversee information locally while imparting it to other Directory Servers over your venture.
3. **Load balancing:** Duplicating your index tree crosswise over servers, you can decrease the entrance stack on any given machine, along these lines enhancing the response time of the server.
4. **Fault tolerance/Failover:** By reproducing index trees to numerous servers, you can guarantee that your registry is accessible regardless of whether some equipment, programming, or system issue keeps your catalog customer applications from getting to a specific Directory Server. Your customers can be alluded to another index for read and compose activities. Note that to help tripover, you should have in excess of one duplicate of your information in your replication condition.

As a rule, you duplicate every single required characteristic for every passage as characterized in the construction, to keep away from mapping infringement, however should you need to sift through a portion of the required properties utilizing the partial replication usefulness, at that point you should make sure to incapacitate outline checking. Having mapping checking empowered with partial replication can keep you from having the capacity to introduce disconnection, that is from a ldif record, since it would not enable you to stack the ldif document if the required characteristics were sifted through. It is important that the turning composition confirming may have the additional advantage of enhancing execution.

The research framework is provided in Figure 3.1.

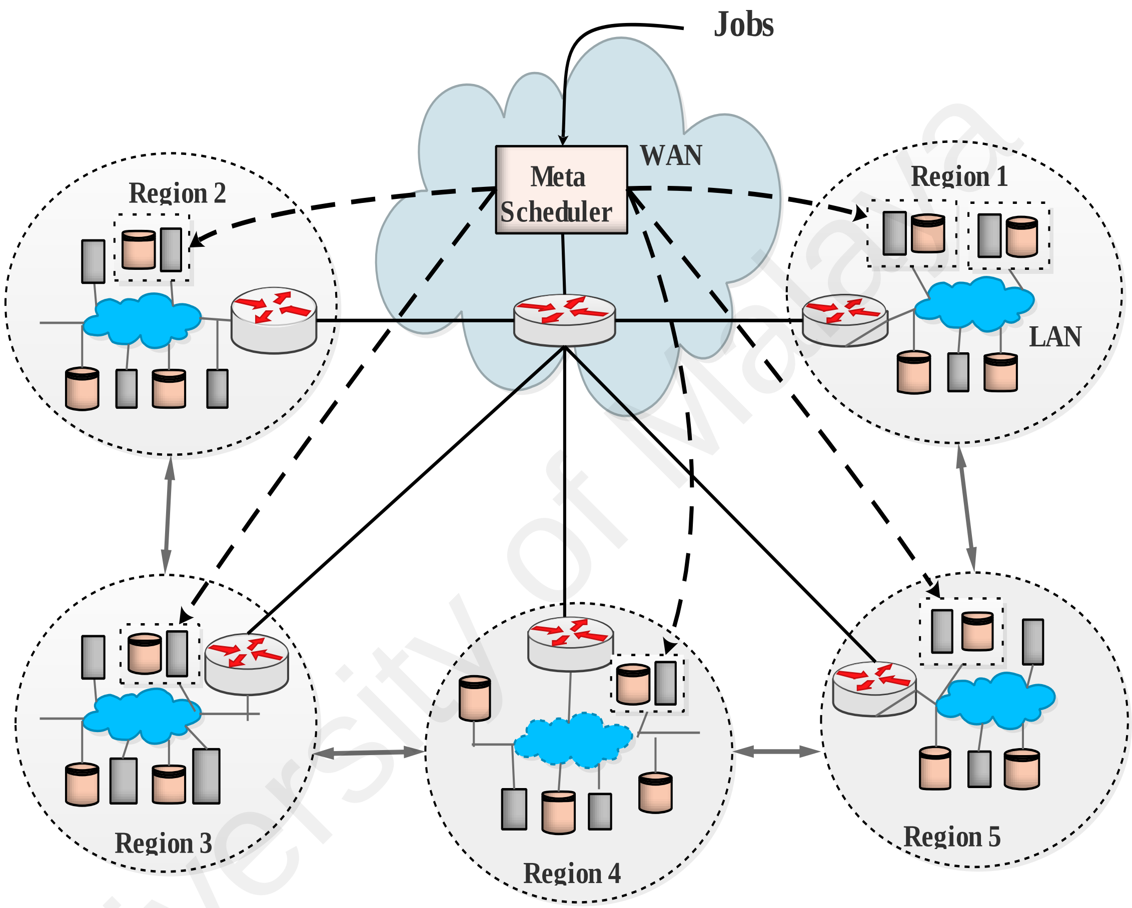
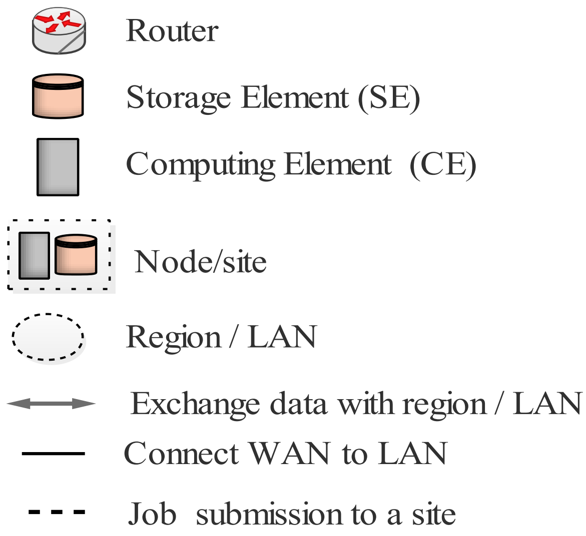


Figure 3.1: Research Framework.



3.3 Research Process

The Research method and process completed in this study is partitioned into four stages as appeared in Figure 3.2

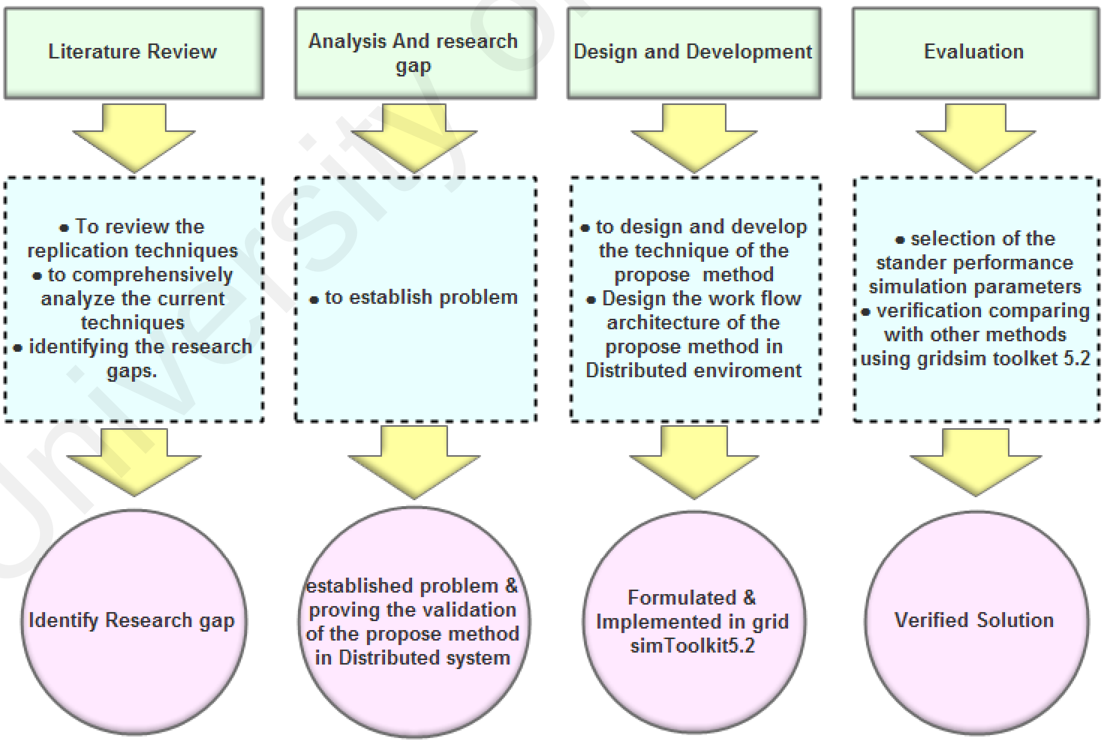


Figure 3.2: Research Process

The entire study is carried out in four main phases as described in figure 1.3. We study the job scheduling and replication implications of the distributed environment with the existing sate-of-the-art in distributed environment considering the earliest to latest techniques. The replication and jobs scheduling challenges of promising distributed architecture is also reviewed and analyzed. A detailed table of the reported techniques of replication and scheduling challenge in distributed system is also elaborated. The possible limitations of each technique that may negatively affect and target the scheduling technique alongside a suggested compact solution to help design a better technique is also highlighted. The current state-of-the-art scheduling and replication with load balances are also critically analysed to come up with a comprehensive overview. Additionally, we examine each state-of-the-art scheduling solution to recognize the differentiating distributed features utilized for each scheduling procedure. The critical discussion on the existing state-of-the-art scheduling improves the ability of the existing scheduling technique in the distributed environment and the research gap that needs exhaustive review. We clearly noticed that the scheduling issue is still a challenge and a key concern that improves the job execution time, response time, reduce data migration rate and network cost. Additionally, load balancing is also affecting the performance of the network. This critical issue should be addressed during the job scheduling of a distributed environment.

A proper job scheduling technique and wise replication procedure in a distributed environment can improve the entire network performance and is investigated and analyzed. To establish the research problem, a detailed analysis is carried out through a simulation environment and implementation of the existing scheduling technique in a distributed environment is carried out to prove that the identified problem really exists. Simulation is performed by using gridsim toolket 5.2 to calculate important metrics such as job execution time, response time, data migration rate, bandwidth consumption, as well as load balance etc.

Dynamic Replica Aware Load Balanced Scheduling (DRALBS) technique/ model were proposed, which have the ability to inspect the replica location and schedule job to a suitable site along with load balance in a distributed environment. The proposed DRALBS gets information of the replica's location and loads the information from the replica manager and load manager respectively. On the basis of these information, the proposed DRALBS points out the best site for the job execution.

The proposed DRALBS is evaluated to get a series of data by using standard parameters for the simulation environment. We acquired more promising results by modelling the proposed replication scheduling technique which was done while performing empirical analysis using simulations. The evaluation is performed by using simulation. The comparison of DRALBS with the other most recent technique of the replication scheduling technique MDHR, is carried out in the simulation environment using the aforementioned important parameters. The qualitative results are then compared with MDHR technique.

3.4 System Model

The Grid Scheduler (GS) is the vital module of a resource management because it is responsible for assigning resources to jobs, taking into account job requirements and resource status. Figure 3.2 shows how an information framework underpins these methods. An arrangement of spaces comprising of an imitation server and other registering locales are the components of what an information lattice framework contains. Capacity for imitations and computational assets for the occupations are given by the copy server and the figuring site separately. Both these working gadgets are usually known as nodal hubs.

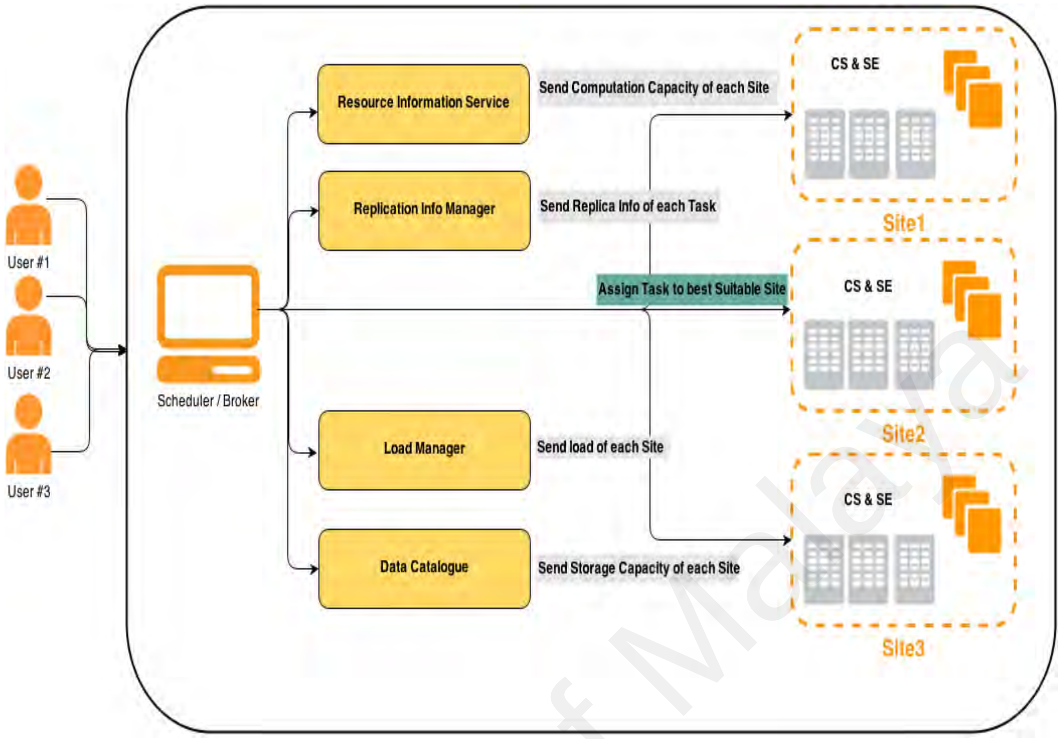


Figure 3.3 System Model.

A neighborhood (LAN) serves every one of the hubs inside an area with a comparative system data transmission of 10-100 Mbps between any two hubs of a space. However, a wide zone arrange (WAN) associates diverse spaces with under 10 Mbps of system transmission capacity between hubs of any two areas. Keeping in mind that the end goal is to enhance and get to the execution of the remote information, there exist reproduction servers that are segregated from different areas going about as one. Table 3.1 demonstrates the data transmission setup. For an imitation server existing in the same space like the registering position, "it is acknowledged as the primary replica server (PRS) for that figuring site, while all other reproduction servers are considered as secondary replica servers (SRS)". In view of particular systems, it is the activity of the Meta scheduler to give out occupations to the computing sites as shown in Fig. 3.2. The replicator receives a set of job from the Meta scheduler, which then based on its access capability, schedules the jobs to different servers. The processing data is accessed directly from the data cache of the computing site, which, if unavailable, is then requested from the PRS. The PRS then searches all the replica servers for the required data, and selects one based on the highest bandwidth being provided to the computing site. The computing site, then receives the required data from the selected replica. The access time for the data is thus improved by these replication strategies as well as the scheduling technique for assigning a job to the site. The Scheduler/Broker ensures the placement of the job on the site where its replica is available while ensuring the load balancing on the entire sites.

Table 3-1: The Bandwidth Configuration in the grid

Network type	Values	Description
Inter LAN node bandwidth	100-10 (Mbps)	All nodes that are serves in the same domain, and between any two nodes the bandwidth is 10-100 Mbps.
Inter WAN node bandwidth	Less than 10 (Mbps)	WAN connects different domains with less than 10 Mbps of network bandwidth between nodes of any two domains.

3.5 Data Replication Methods

Replicating the file to an appropriate place is itself a problem to solve preferably. Appropriate replica placement determines how and where to replicate the file and which file to replicate. Dynamic and static are the two types of Data/Information Replication strategies. Replica remain same in static technique until its duration expires or it is deleted by users. Static replication is easy to manage short term scheduling, but it is not suitable for a long term scheduling mechanism. Users request rate changes very frequently, it may decrease the benefit of static replication. Alternatively a dynamic replication leads to concern of changes in a distributed grid environment and mechanically generates new replicas for common data files and ultimately improves the data access time. Replica placement is done on the basis of previous records. As it is assumed that the widely held data in the nearest future will also be popular in a next session as well, thus, dynamic replication strategies are the flexible ones, which meets the end user requirements.

We have taken into consideration, the dynamic data replication method and replicating the widely held data, which is examined by the access number (NOA) of that data as done in the work of (Wang, Yao, Xu, & Pan, 2017). Let file ID be FID, the history table (FID, NOA) shows the access number of each file. Larger NOA of a file shows its fame proportion. Replication aims not only to minimize access time, but also increases the robustness of an application.

Our proposed work is based on a centralized dynamic replication strategy, where the centralized replication algorithms determine the scheduling and the replication methods are discussed in the next section chapter 4.

3.6 Metrics For Evaluation

In a Data Grid, diminishing the activity's turnaround time (holding up time in a line, execution time and information exchange time) relies upon where work is appointed for execution and where the required information records are acquired. This also could occur in an event when information exchange time and final information would get to all the more often as possible. Therefore, from a scheduling point of view, assigning jobs to proper sites and getting replicas from proper sites are some of the important factors to consider. Since each job before execution requires many data files, data locality has an important role in any scheduling decision. If a job is dispatched to a site where replica is available, there would be no data transfer time and the job execution time will be minimized remarkably.

Execution time: The execution of a job's time is considered to be the full implementation of the activity, which is the time for exchanging documents and processing activities. The time saved by transferring records is the most critical factor affecting the time of a job execution in the matrix of information. Enforcement time limits specifically reduces the time of information entry.

Mean job execution time (MJET): Job execution time is defined as the total time to perform each job divided by the number of completed jobs. The aggregation time includes the time from the job that enters the place to predict when the execution takes place and the time of leaving the place.

It is treated accordingly as shown in the equation.

(3.1) as:

$$MJET = Job\ Arrival\ Time - \frac{Job\ Departure\ Time}{No.of\ Jobs\ Completed} \quad 3.1$$

Jobs Scheduling for a fitting figuring nodal hub is fundamental in a disseminated condition, since information exchange among registering nodal hubs diminishes the getting of information with time. Accordingly, We have considered the duplication position at the stage of the programing strategy that minimizes execution time and the data transferring rate consequently also improves the data access time.

A perfect replica capture enhances work execution time and cost of information maintenance. The proposed data/information replication and occupation booking methodology endeavors to lessen work execution time, and additionally diminish usage cost, by limiting the information exchange and erasure rate in an information matrix condition. Besides, the proposed system at last limits usage cost and enhances the data/information gotten to rate fundamentally. The scheduler sends the job directly to the relevant site, minimizes the data access time and likelihood of transmission and deletion, as most of the required data already exists on the site, first finds the replication of the desired file in the zone point.

This will outstandingly diminish add up to the exchange time, and subsequently information get to time. The essential objective of information replication is to abbreviate the information get to time experienced by the activity and therefore decrease the activity turnaround time.

In this research, replication aims to shorten data access time perceived by the job. As a job could be submitted to any computing site for execution, which is determined by the Grid scheduler on the basis of information taken from replica manager, an individual computing site does not have an inherent data access pattern.

3.7 Conclusion

We proposed a system model for jobs scheduling in a distributed environment where the received jobs are sent to the nearest server by considering the replica location. We also provided the key concept of replication and scheduling design strategy to clearly show that our proposed method is different from other existing techniques. Moreover, we extend the detailed description of replica placement and scheduling implementation. In this chapter we also suggested a solution to improve our work in terms of response time and load balancing of all sites, of which for this purpose we used two new modules replica manager and load manager. In the next chapter, we shall explain the system model and our contributions as described previously.

CHAPTER 4: PROPOSED SCHEDULING STRATEGY

4.1 Introduction

This chapter offers comprehensive details about the proposed scheduling technique by considering the replica location to schedule jobs to the appropriate site in a grid distributed environment. The purpose of the proposed planning technique is to improve system performance by replication and scheduling techniques in terms of mean job execution time, response time, less migration rate of required data, less bandwidth consumption, and balanced load of the system. To decrease the searching time of required data and increase data availability in a distributed environment is a challenging task. The dynamic hierarchical replication is one of the best solution for this type of problem. DHR and MDHR are the most famous and well known techniques for replication and job scheduling in a scattered location. HDR is one of the recent technique. Therefore, we compare our proposed technique MDHR. The following discussion offers an in-depth description of the algorithms. .

4.1.1 Dynamic Hierarchical Replication (DHR)

The DHR technique is designed to improve the performance of HDR. This technique has basically three categories: The Replica selection, Replica Data placement and Replica data management procedures.

Replica Selection: The job is scheduled to be on a specific site and the required data files will be sent to another site to become a new copy. When a node's different resources have copies of the same file, choosing the best copy has a significant advantage. Reaction time is the main parameter that affects copy selection and job conversion time.

Each storage space element (SE) has multiple requests simultaneously and the storage has to respond to a single request concurrently, so the requests have to wait in the queue. The (DHR) technique selects the optimal copy location for a small execution time, which can be calculated by following the data transfer time and the number of requests that will be consistent with the site store in wait mode. In the case of copy data selection, if the data file already exists in the same LAN, it will create a copy candidate list and select the node's resources with minimal data requests.

If the file is not found on the same network grid LAN, DHR will search the grid LAN. If the file already exists in the same network zone, a copy candidate list is generated and a copy with the least number of requests is selected. Otherwise, it will generate a copy list/row that exists in some other areas and use this list to select the copy with the least number of requests.

Placement Technique for Replica Data: DHR technology creates a copy of the data in Best Storage Element (BSE). In order to select BSE, DHR searches for SE with the smallest SE value (VSE) in the desired area. When calculating a VSE, you must access the number of requests for a copy of the data and finally access the copy or access. These values are useful as they give the probability of the need to request the replica again (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013).

Replica Data Management Technique: If there is no required data storage space in the BSE, copy the selected required data file. Otherwise, if the data file exists in the LAN, you must be able to access the data file from its own location. Now, if there is no required space for data replication and the required data file is not present on the same grid LAN then delete files as per your requirements.

4.1.2 Modified Dynamic Hierarchical Replication (MDHR)

MDHR is an extension of DHR. MDHR minimizes the implementation cost significantly while considering the location, previous accesses, and replica size. The jobs scheduled on each computing sites are expected. Therefore, each transfer action is required to be done on the basis of link cost so it could be better if we consider replica before scheduling the job. In MDHR, as the number of the replicas increases, the rate of transfers and deletions, congestion rate, and the implementation cost also increases. Valid scheduling mechanism can minimize the data access rate meaningfully.

Data replication is a very important technique for improving response time in grid applications. In replication, it generates multiple file copies and places them at many locations to reduce the file fetch times. MDHR technique is a new improved technique and enhanced Dynamic Hierarchical Replication (DHR). However, replication of the data file will be utilized intelligently as the space for storage is limited for each site of the grid. Therefore, it is good to set up a technique for the data replication placement/replacement task. The drawback of DHR strategy in replica placement procedure is covered by MDHR because it keeps the most important replica instead of the less important replica. MDHR supersedes a copy of the data that depends on the latest access time, the number of access rights and the entire size of the directory. It chooses the best replication site of the entire repository file based on the fast response time that can be found in the data transfer data, access time, transfer request in current storage current, medium distance, etc. Some research was conducted on Website and CPU process capabilities (Mansouri & Dastghaibyfard, 2014; Mansouri et al., 2013). MDHR checks the available storage capacity for a new replica, if it is not sufficient, then it will remove the records that have the minimum process time, ie. the recording takes place in LAN and the local area.

If the storage space for a new replica is still not enough, then it makes decision on the basis of last time request of replica, number of access and replica data size. The other problem of DHR is the selection of the best replica among many replicas of the system. The MDHR methodology picks the best area of replica for the client's execution, considering new parameters besides the information exchange time, namely, latency for storage queues, storage access time, latency for storage queues, and distance between sites. Although, MDHR enhances the performance of DHR strategy, it still needs to improve because during the scheduling method, it does not consider the location of the replica on a specific site. Even though MDHR is better than other stated techniques, however it still needs to be improved because it is not considered to replicate a place at the time of workplacing. Replica manager decides on which site to assign a job where the required data presented can affect the execution time of a job and also cause increase in the data migration rate. Therefore, we choose MDHR, a most recent technique and a better techinique than other existing dynamic hierarchical replication techniques in this research.

4.2 Terminology and Concepts.

The terms which are mostly used in this work are explained for reader better understanding and concepts in the following table.

Table 4.1: Scheduling Terminology

Terminology	Concepts
Replica	A replica is a complete duplicate of a file connected to the sources record through some well-defined mechanisms.
Task	A task is an atomic unit that is scheduled by the scheduler and assigned to a resource. The attributes of the task are parameters such as priority, deadline CPU memory requirements
Job	A job is a set of atomic tasks that are performed on a set of resources. A jab can have a recursive structure. In other words, a task consists of sub tasks and the sub jobs themselves can be decomposed into basic atomic tasks. The term job application and Meta task are

	interchangeable in this work.
Resource	Resources are what you need to do your operation. For instance processor and network link for data processing and data transporting respectively or storage device.
Resource utilization	The usage amount of all the resources in a system.
Procedure	A procedure is the task to be scheduled in a system. It contains information as the number of resources and the amount of time that is required for each resource.
Site	A site (or node) is an autonomous entity that consists of one or more sources.
Makespan	Makespan is the total time length of a schedule.
job scheduling	Job scheduling is the mapping of jobs to selected resource groups that can be distributed across multiple administrative domains. A task scheduling is the mapping of tasks to a selected group of resources which may be distributed in multiple administrative domains.
Meta-scheduling	Meta-scheduling is a PC programming procedure of streamlining computational workloads by joining an association's different distributed Resource Managers into a solitary collected view, permitting bunch jobs to be coordinated to the best location for execution.
CDN	Content Delivery Network, is a gathering of servers deliberately set over the globe with the reason for conveying static substance to clients considerably quicker.
Load	The number of processes in waiting queue represents a load of resource. A low loaded resource of site response is high as compared to high loaded resource.
Fault Tolerance	To recover data in case of failures, errors and any disappointment occurrences in the grid. Another perspective is to cover disappointments and consequently discover elective approaches to take care of a given issue.

4.3 Data Replication Methods

Replicating the file to an appropriate place is itself a problem to solve preferably. Appropriate replica placement determines how and where to replicate the file and which file to replicate. There are two types of data replication, static and dynamic. In the static method, replica remains same until its duration expires or it is deleted by users. Static replication is easy to manage short term scheduling, but it is not suitable for a long term scheduling mechanism. User's request rate changes very frequently, and it may decrease the benefit of static replication. On the other hand, a dynamic replication method is concerned with changes in a distributed grid environment and mechanically generates new replicas for common data files and thereby ultimately improves the data access time. Replica placement is done on the basis of previous records. As it is assumed that the widely held data in the nearest future will also be of much popularity in the next session Thus, dynamic replication strategies are the flexible ones, which meets the end user requirements.

In this research, the dynamic data replication method and replication of the widely held data, is examined by the access number (NOA) of the data as done in the work of (Wang et al., 2017). Let file ID be FID. The history table (FID, NOA) shows the access number of each file. Larger NOA of a file shows its fame proportion. Replication aims not only at minimizing the access time, but also to increase the robustness of an application.

Our proposed work is based on a centralized dynamic replication strategy, where the centralized replication algorithms determine the scheduling and replication methods, discussed in the next section given below.

4.4 Proposed Algorithms

DRALBS (Dynamic Replica Aware Load Balanced Scheduling) is the proposed algorithm of our work. We present the scheduling and replication strategies in two steps.

Replication algorithm is applied on the data composed from the scheduler. The algorithm extends the work done in MDHRA, which is the extended form of DHR (Mansouri & Dastghaibyfard, 2014).

Initial Placement of job in DRALBS Algorithm

Algorithm 1: DRALBS Algorithm for initial job placement

Input: Number of files (jobs) to be processed

Output: Replicate the files on the most accessed computing sites

Step 1: Loop All Tasks

Step 2: Loop All Tasks Files

Step 3: if (files for tasks is present)

Step 4: List files=Add File ()

Step 5: Loop All Resources

Step 6: if (files is present on resource)

Step 7: List resources=Add Resource (files)

Step 8: Loop All Resources with Required Files

Step 9: if (resource bandwidth is max && load is balanced)

Step 10: Assign best resource

Step 11: Submit Task on best resource

Algorithm 1 is self-explanatory. However, a brief explanation of DRLABS Algorithm is as follows. The algorithm shows how to find the optimum resource for a given task to be processed. First, a task comes to assign some resource to be executed. We then look for required files for this task on all available resources. If not found, then we must assign file on a least loaded resource.

The Dynamic Load Balancing in DRALBS Algorithm

Algorithm 2: DRALBS Algorithm

Input: Task placed on the sites

Output: Tasks placed on the sites with equal load

Step 1: Loop All Resources

Step 2: If Resource load>Upper threshold

Step 3: Find least loaded resource

Step 4: Loop All least loaded resource Tasks

Step 5: If least loaded load + Task load < upper threshold and replica available

Step 6: Migrate Task to least loaded (task)

Step 7: Update load of least and heavily loaded (task)

Step 8: Take Next Resource

Step 9: Continue

Algorithm 2 is designed to place tasks on site with equal load. This part explicitly shows the task migration to achieve load balancing. The migration is dependent on the resource load plus the task load. Moreover, a threshold is used to decide when to migrate a task.

4.5 DRALBS ARCHICTURE DIAGRAM

Figure 4.1 describes the working process of the proposed strategy.

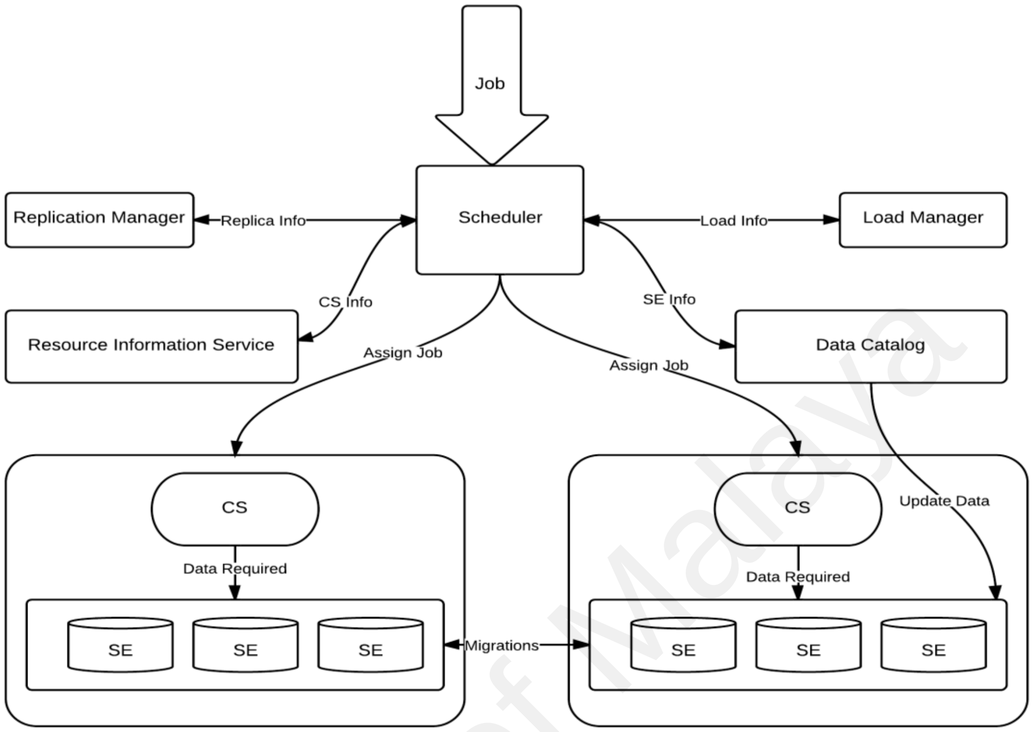


Figure 4.1 DRALBS ARCHITECTURE DIAGRAM

The mechanisms of the architecture diagram are described below.

4.5.1 Resource Information Service

The user submits a task to Meta scheduler by identifying the features of job (i.e job length communicated in Million of Instruction per Second (MIPS)) and excellence of service requirement (i.e processing time

4.5.2 Resource Information Service

Resource Information Service sends the information to the scheduler about the computing capacity of each resource of the site so that it can easily compare the job and site computing capacity for scheduling purpose.

4.5.3 Replication manager

Replication manager sends the information to the scheduler about the replica of the job data that where it is placed on the site.

4.5.4 Load Manager

Load Manager sends the information about the load on the each site of the grid. After collecting location of the replica from the replication manager and load of all the sites then the scheduler will decide that where to execute the job so that there will be no need of replica migration or accessing replica remotely.

4.5.5 Data Catalogue

Data Catalogue sends the information to the scheduler about the storage capacity of each resource of the site so that it can easily compare the job and site capacity for scheduling purpose.

4.5.6 Site

The site is a very important part of the grid. It contains both computing and storage units of the grid. The Scheduler decides the future of the job on the site and sites have to only process the request and send back the results to the scheduler.

4.6 Illustrative example

The concept of the scheduling of a job considering replica is illustrated via a simple example, shown in fig.6 which shows the scheduling technique used in our proposed algorithm.

User Submits the job to the scheduler with some requirements of the job like MIPS and storage. The Scheduler then gets the information from the Replication Manager, Load Manager, Resource information service, and data catalogue. This information contains the replica location on sites of relevant job, load of all the resources, resource capacity on all the sites, and storage capacity of the entire sites respectively. Now Scheduler has all relevant information about job replica location and all sites computing and storage capacities. The Scheduler will schedule the job where the replica is present on the site, where it matches requirement of the jobs and where the load is balanced

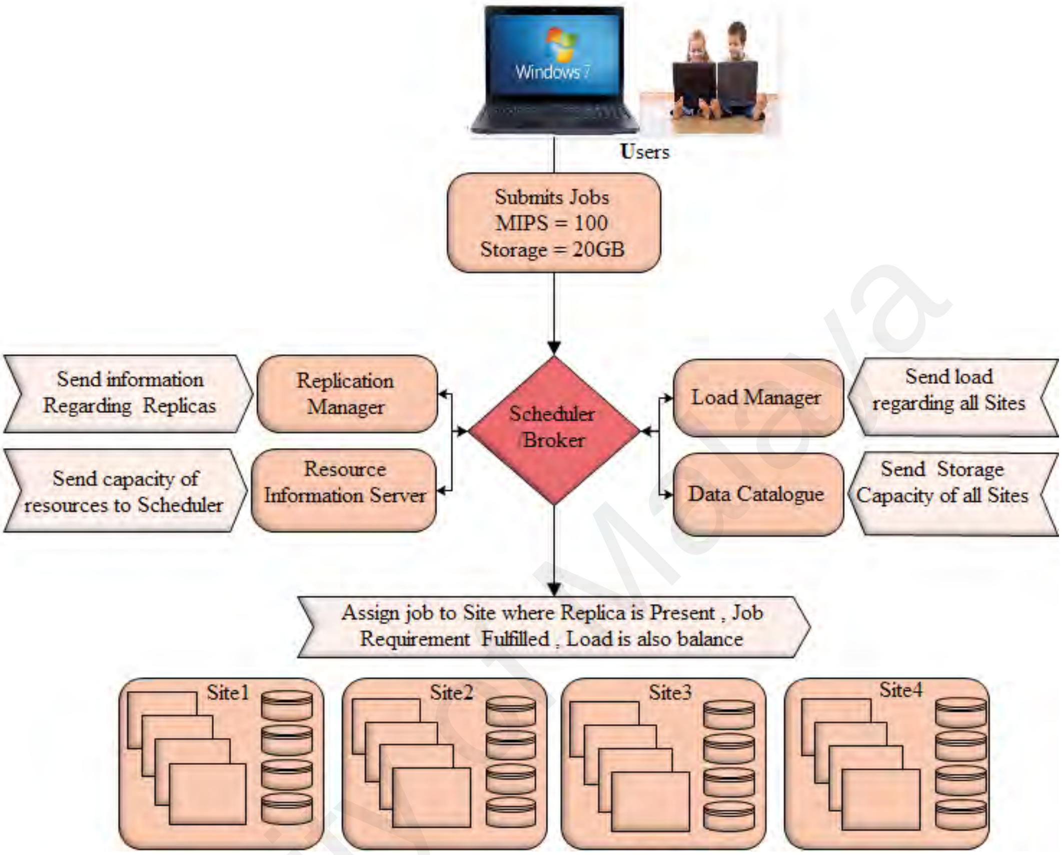


Fig.4.2 Illustrative example

4.7 Conclusion

We select the most recent technique of replication known as Modified Dynamic hierarchy replication (MDHR) which enhances the performance of previous techniques, but still has some limitation and there is need to improve its technique, therefore "we propose a Dynamic Replication Aware load balancing scheduling" technique to overcome the existing drawback and improve the performance of a system. This technique schedule jobs by consideration of the replica location while keeping load balance. Moreover, we presented the architecture diagram and describe its parts based on how we completed the proposed Dynamic Replication Aware Load Balance Scheduling. Finally, we explained it with examples for better understanding.

CHAPTER 5: RESULT AND ANALYSIS5

5.1 Evaluation and Setup

Experimental results that were formed through simulation are explained in this chapter. At first the implementation cost was compared between DRALBS and MDHR algorithms. Then a series of experiments investigates the geometric mean of turnaround time difference between DRALBS and MDHR (Mansouri & Dastghaibyfard, 2014). Lower geometric means of turnaround time mean enhances the performances from the end user point of view and the last set of experiments investigates the turnaround time, average turnaround time, the smallest and largest turnaround time.

5.2 Setup

Replication and scheduling is done in a Data Grid environment, which consist of 25 domains in our consideration and every domain has a server known as replica server. There are 80 processing nodes unsystematically distributed to 25 domains. The links between the computing and server nodes are given the bandwidth and are randomly distributed. Bandwidth with the same domain were set equivalent, but outside the domain, the bandwidth is set differently. Number of files varying from 1000 to 2000 was used and the file size was homogeneously distributed between 100 and 400. The intial copies are allocated unsystematically to every site.Table 5.1: Simulation Parameters

Table 5.1: Simulation Parameters

Parameter(s)	Value(s)
No. of Jobs	2000
No. of Job Types	6
No. of file accesses / job	16
Single file size (GB)	2
Delay of job	2500
Queue Size Maximum	200
Intra LAN Bandwidth (Mbps)	1000

5.3 Evaluation of MDHR Algorithms

Execution cost in terms of migration, executions, storage and bandwidth consumption is initially matched with the MDHR scheduling mechanism. We have created replicas on each site of the grid by uniform random distribution and assume equal storage capacity for all the replicas. We have searched the required files for each job on the resources of the sites and gotten the resources where files are available. We had to come up with the best from the list of the resources in comparison with the bandwidth and load. Finally, we submitted the job to the best resource found which have more bandwidth, balance load and replica availability.

5.4 Simulation Results and Discussion

In this section, the performance of the proposed algorithm is compared with that of the existing i.e. we compare DRALBS and MDHR. The performance metrics are response time, data migration, mean job execution time, mean job execution time with bandwidth and data migration and data on the first hit.

5.4.1 Response Time

Our simulation results with respect to response time for both DRALBS and MDHR is presented in Figure 5.1.

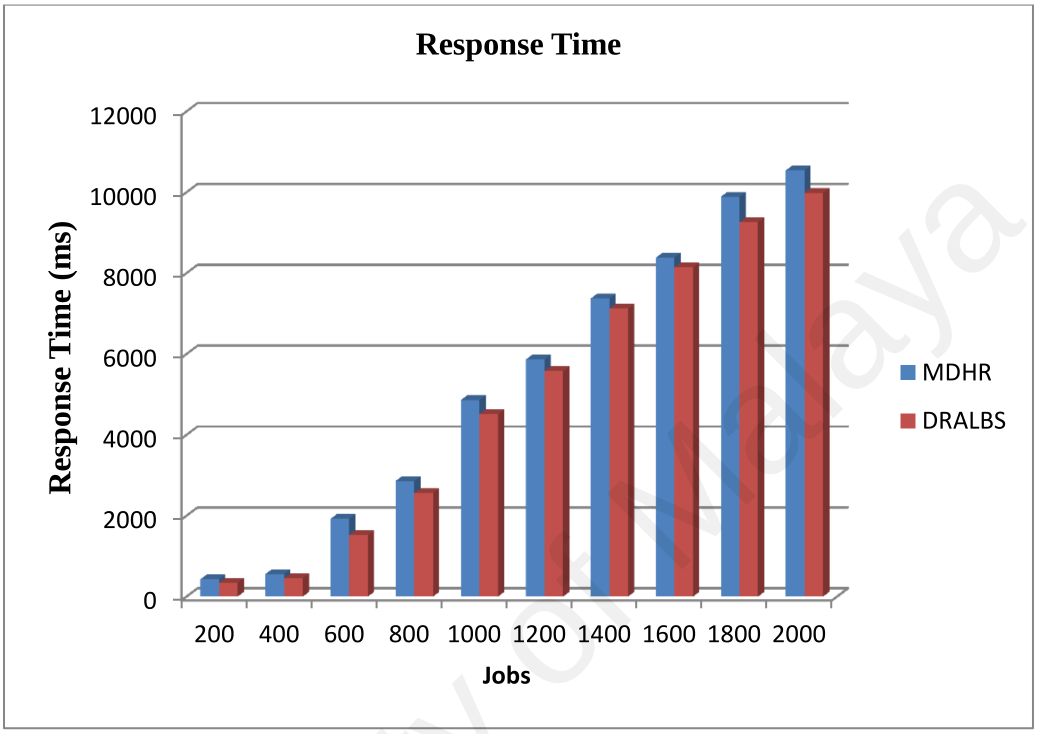


Figure 5.1 Response Time of Grid lets by DRALBS and MDHR

Fig. 5.1 demonstrates the results of the average response time with varying number of jobs. The results indicate the superiority of DRALBS over MDHR scheme in terms of response time. This is because DRALBS schedule job to a site where replica already exist with least load. The demanded job's file is found on the assign site with no need of remote access of data from another site, which makes it possible to reduce the migration rate and improves tasks response time. We provide the response time results of both algorithms in Table 5.2.

Table 5.2 Average response Time

No. of Jobs	Average Response Time		Increase response Time in %
	MDHR	DRALBS	
200	430	342	44
600	1926	1620	51
800	2863	2562	37.6
1000	2862	4510	35.2
1400	`7375	7120	18.3
2000	10550	9980	27

Table 5.2 clearly shows that the DRALBS performs better than MDHR in terms of response time at any given job.

5.4.2 Data Found Without Migration

The simulation results for both DRALBS and MDHR in terms of data found without migration is presented in Figure 5.2.

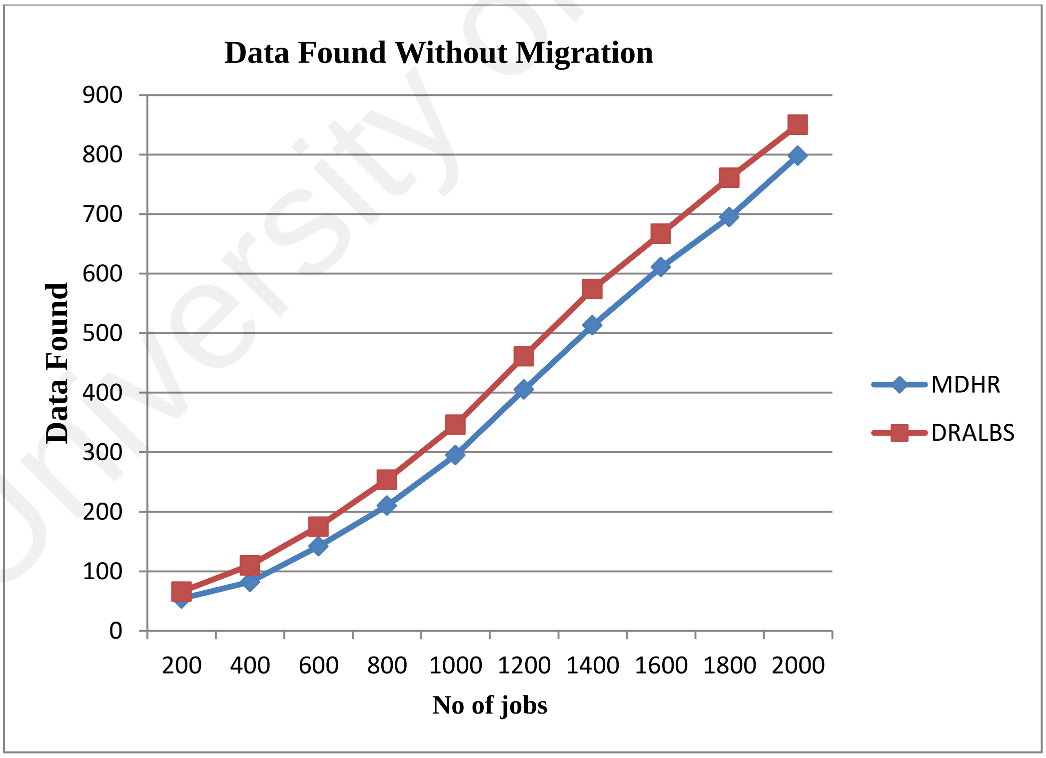


Figure 5.2 Data Found without migration by DRALBS and MDHR

Fig. 5.2 demonstrates the results of data found without migration with varying number of jobs. The results compare DRALBS and MDHR algorithm with the copy for each file size varying from 5 to 100. It indicates the superior performance of the DRALBS approach compared with its contemporaries in terms of data found without migration. The reason is that the proposed algorithm can guarantee replica position while scheduling the job which provides a good ratio of data found on the first hit in comparison with MDHR.

5.4.3 Data Migrations

Our simulation results with respect to data migrations for both DRALBS and MDHR is presented in Figure 5.3.

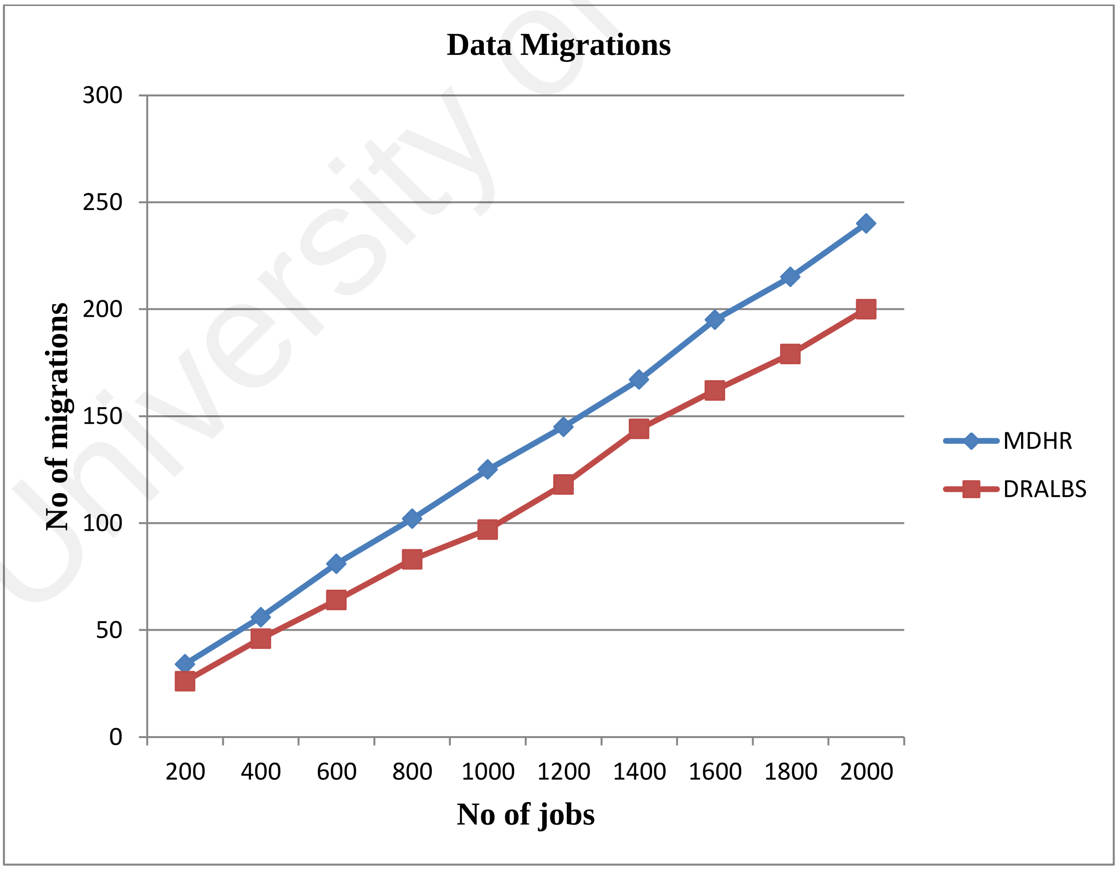


Figure 5.3 Data Migrations in DRALBS and MDHR

Fig. 5.3 demonstrates the results of data migrations with varying number of jobs. It clearly shows that DRALBS performance supercede MDHR in terms of data migration from site. This is because DRALBS has efficient scheduling technique than it's counterpart which makes data migrations to occur better in DRALBS than MDHR. The DRALBS schedule job to a suitable site that contains replica which reduces the remote access of data from/to other sites. We provide the summary of our results based on data migration in Table 5.3.

Table 5.3: Data Migration

No. of Jobs	**Data Migration**		**Reduced data migration in %**
	MDHR	**DRALBS**	
200	34	25	4.5
600	80	63	2.83
1000	125	97	2.8
1400	`167	144	2.06
2000	240	200	2

Table 5.3 shows that when using DRALBS, reduced data migration occurred on site because it considered replica location at time of scheduling. This is the main achievement of our algorithm. We have reduced the number of migrations needed for the job to execute which results in low response time, low turnaround time, better bandwidth consumption, and good execution time of the job in comparison with MDHR.

5.4.4 Mean job Time Based on Bandwidth

Our simulation results on mean job time in light of bandwidth for both DRALBS and MDHR is presented in Figure 5.4.

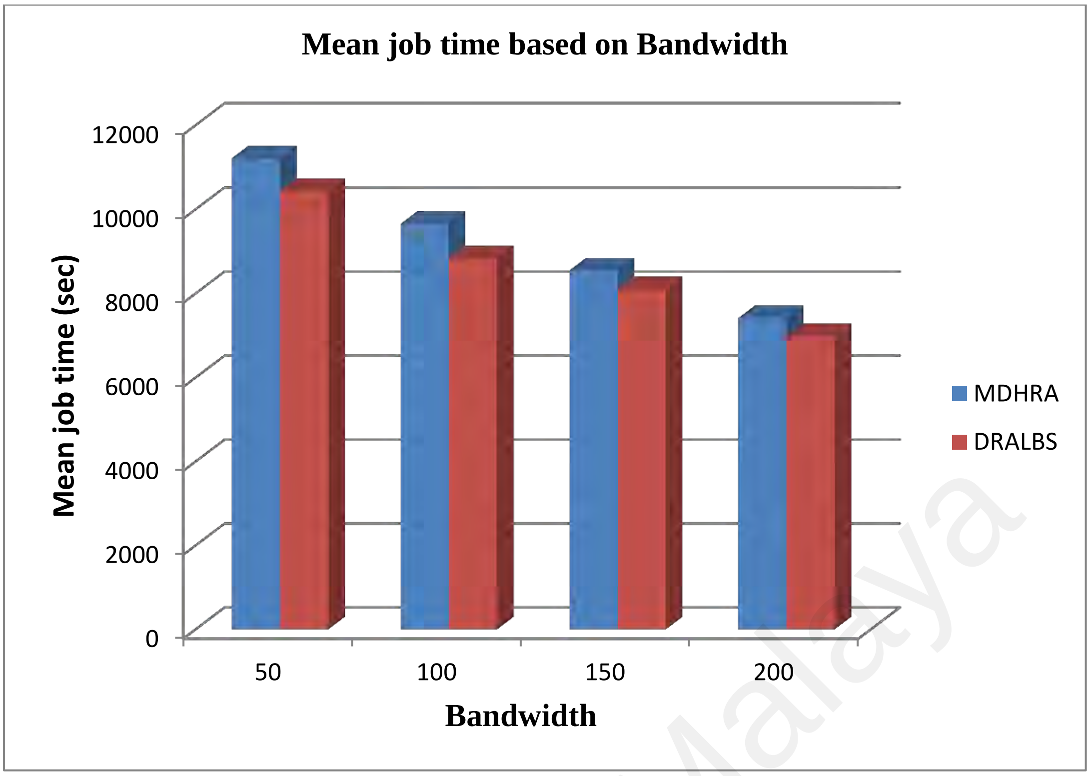


Figure 5.4 Mean job execution time with Bandwidth of DRALBS and MDHR

Fig. 5.4 represents the results for the mean job execution time with bandwidth for MDHR and DRALBS with the replica per file size varying between 5 to 100. The result proves that DRALBS performs better than MDHR in terms of the mean job time for executing bandwidth of 50, 100, 150, and 200Mbps. The reason is that DRALBS reduces the number of migrations of data files which results in low bandwidth consumption and job executes at the site of LAN in every case as shown in the result.

5.4.5 Mean Job Execution Time

The simulation results for both DRALBS and MDHR in terms of mean job execution time is presented in Figure 5.5.

Figure 5.5 Mean job execution time with jobs of DRALBS and MDHR

Fig. 5.5 demonstrates the results of mean job execution time with varying number of jobs with the replica per file size varying between 5 to 100. The results indicates that DRALBS performs better than MDHR under this performance metrics. This is because DRALBS ensures that the site has replica and then schedules job to appropriate site with least site load that minimize migrations of data files, which tremendously improved the mean job time for execution with the number of jobs in the sites of the grid 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000. Therefore, DRALBS shows better results than MDHR algorithm.

5.4.6 Mean Job Time with File Size

The simulation results for both DRALBS and MDHR in terms of mean job execution time with file size is presented in Figure 5.6.

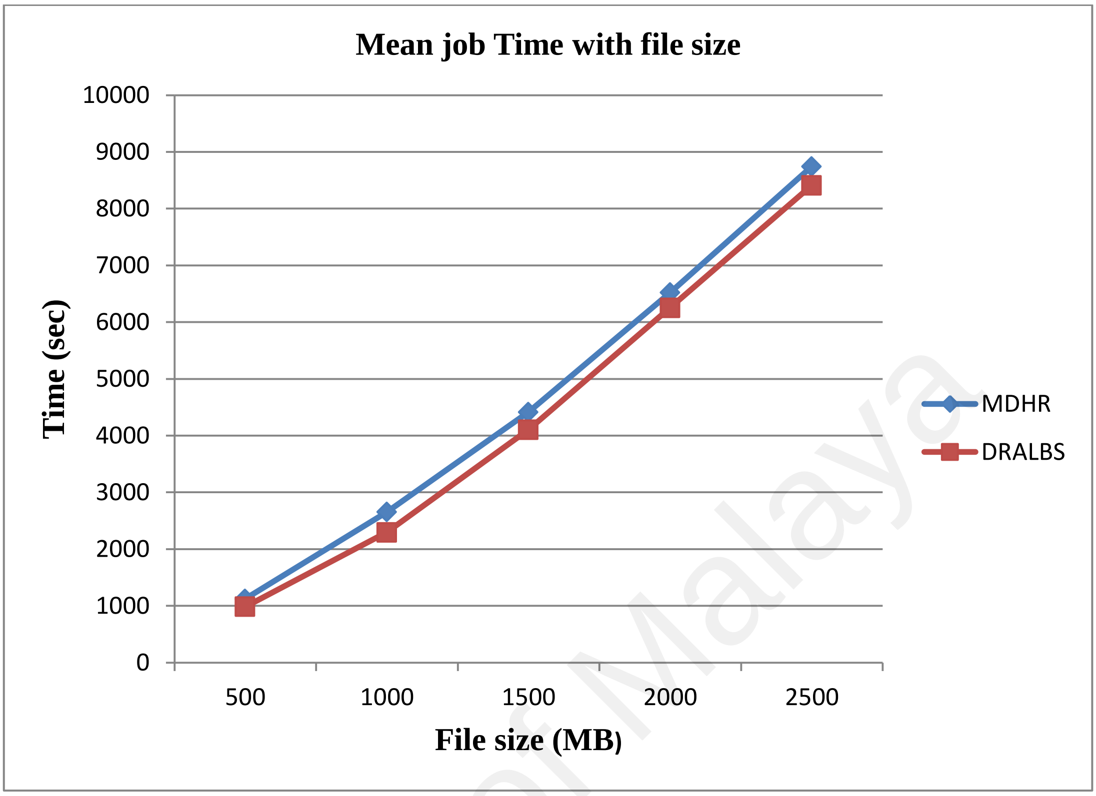


Figure 5.6 Mean Job Time with File Sizes of Jobs on MDHR and DRALBS

Fig. 5.6 demonstrates the results of mean job execution time with varying number of jobs with file size varying between 5 to 100. Thus, the results indicates that DRALBS performs better than MDHR in terms of file size. This is because of the minimum amount of migration that occurs with the DRALBS site ratio compared to MDHR. Moreover, as the file size increases, the execution time also increases. X-axis shows the file size starting from 500 to 2500 and the Y-axis shows the execution time of the job with file sizes.

5.4.7 Mean Job Time with Size of SE

The simulation results for both DRALBS and MDHR in terms of "Mean Job Time" with volume of SE is presented in Figure 5.7.

Figure 5.7 Mean job time with sizes of SE on DRALBS and MDHR

Fig. 5.7 shows the results and performance of MDHR and DRALBS in terms of mean job time with SE sizes of the sites with the copy per file length varying between 5 to 100. It shows that the mean job time with SE of DRALBS is better than MDHR. This is as a result of the low migration rate of DRALBS, which leads to a better execution time of the job on sites. From the result, the X-axis denotes the size of storage elements on the sites and the Y-axis denotes the execution time of the jobs with the mentioned size of storage elements. When there is greater storage space then we can store as much replicas and vice versa. However, our results clearly show that when the size of a storage element is large, then the response time is minimized.

In summary, for clarity purpose, we provide the summary and comparison of our performance metrics in Table 5.4.

Table 5.4 Comparison of Results between DRALB and MDHR

Performance Criteria	Comparison		Measurement Range		Result: **DRALBS** is better than **MDHR**
	DRALBS	**MDHR**	**Tim(Ms)**	**No. jobs**	
Response time	Low	High	0-1200	200-2000	Improve average Response time up to 32% as compared to MDHR
Data Migration	Low	High	0-300	200-2000	Reduce data migration rate up to 4.5% as compared to MDHR
Mean job execution time (MJET)	Low	High	0-1200	200-2000	Due aware replica location and load balanced policy, The mean job execution time has improved 77.8% that of MDHR
MJET with bandwidth	Low	High	0-1200	1-200	Assign jobs to a site having replica and less load (LAN), save average bandwidth consumption up to 24.8%
Data find on first hit on assigned site	High	Low	0-1200	20-120	The DRALBS schedule job to a suitable site that increased first hit data rate Up to 23 % that of MDHR

5.5 Conclusion

Traditional Grid Scheduling did not consider replica location at the scheduling time. It assigns the job to a site which can only fulfil the computing and storage capacity of the job. Then it will look into the replica manager of the site as to see whether its replica is available.

Our proposed algorithm does not only maintain the replica information, but also effectively utilizes it while doing the scheduling decisions. DRALBS produces better scheduling results for implementing job scheduling on the sites, but implementation cost is minimized by minimizing the migration ratio or transferring and deletion. DRALBS does not only improves the response time, but also try to decrease the possibility of transfers or deletions in the case of data requests.

As discussed above we have surely improved the response time, throughput, Mean job execution time, with jobs and with bandwidth, job execution with varying file size and storage element sizes. They all are better than the technique of MDHR. Our technique tremendously reduced the migration ratio, which results in the betterment of all the above parameters. The Storage size of the sites is also very important. As increases occur in the response time, so also does the execution time decreases. We have much more space for the placement of the replicas if the storage element size of the site is bigger.

Load balancing is also a very important factor to consider. If we have computing and storage capacity available on the site and we are only targeting some of the sites, it’s also not a good practice. We have to consider load of all the sites as well as the location of the replica.

CHAPTER 6: DISCUSSION AND CONCLUSION

6.1 Introduction

Chapter 6 puts an end to all the work by means of clarifying the summary and achievements of the study. Moreover, the main findings of the research and accomplished objectives are summarized along with highlights on the significance of Dynamic Replica Aware Load Balanced Scheduling (DRALBS). Finally, the chapter also supports the possible extensions in the future, provides limitation and the importance of this work. The chapter furthermore presents the ongoing scheduling issues for more efficient scheduling future research directions. The entire study articulates the Dynamic Replica Aware Load Balanced Scheduling (DRALBS). Section 6.2 explains the main findings and objectives of the research. Section 6.3 discusses research contribution. Section 6.4 examines the scope and limitation of research work. Section 6.5 provides the possible extensions along with future plans.

6.2 Reappraisal of Research Objective

The research explores the problem of developing an effective scheduling technique for jobs scheduling by considering the replica location and site status.This technique is suitable to schedule jobs in a wise manner to reduce job execution time, migration rate that is, to reduce remote access of data from other sites or regions with load balanced for reliable communication in distributed networks. Replication is also an optimization technique which reduces the access time data by keeping copy of big data in various sites. There are several techniques that are developed for replica selection and placements in distributed systems. The first objective of literature was conducted through a review of replication scheduling. We studied the latest literature on replication and scheduling techniques with the existing state-of-the-art scheduling solution.

Motivation of this research study was to address the issue of job scheduling technique on replica location which makes it different and more effective than other techniques for resources sharing with less network bandwidth and storage consumption in distributed environment such as grid. Although replication is one of the key promising technique for quick access to big data in a geographically distributed environment, replication in all conditions was not considered as the best solution. Due to limited storage space for nodes, sometimes, replication of big data on each node becomes a serious problem which badly affects the network performance in terms of the latency of data, storage consumption, bandwidth consumption and load on nodes. The second objective is achieved by closely identifying issues of the existing replication and scheduling techniques and establishing the problem by highlighting the issues of replica placement on site and best replica selection. Basically, we are faced with two problems in data replication. One is the replica deployment issue wherein the selected server should be distributed and the other is the choice of the best replication site for easy access to users, ie. about how to reduce the data runtime.

The challenges in replication selection and placement techniques are examined. Moreover, to clarify the difference among these selected techniques, a comparison study is provided. We aim to develop a scheduling technique that submits the jobs to a suitable site, that knows about replica existence and if the work is assigned to a place that already has the necessary data for execution, it is not necessary to access.

Consequently, execution time and data migration will be reduced. The third major objective is to develop a system model for replication scheduling which considers replica location at the time of job scheduling in a distributed environment. The structure diagram for the proposed scheduling technique is elaborated in Section 3.2. The model structure contains two extra modules; replica manger and load manager which provides information to the scheduler about replica location and load on sites. On the basis of adding these informations, the scheduler then decides which site is suitable for job execution. The functional details of each part are explained in Section 4.5. Furthermore, the algorithm for the proposed scheduling technique is defined. There are two parts of our algorithm, the first is to assign job to a site where the replica is present and the second is to find the least load resources of site to assign job and maintain load balance on site. Therefore, this developed technique decreases job execution time as well as migration rate of data. We performed a rigorous evaluation of the proposed technique. The proposed scheduling model is evaluated using simulation grid sim toolkit 5.2 and analyzed by comparing its results with an existing MDHR technique in a distributed environment. The comparison is made among the response time values, data migration, mean job based on bandwidth, execution time, mean job with a file size and size of storage element. The verification of our approach is made with the benchmark existing MDHR technique. Moreover, for clear understanding of readers, we summarized the improving performance results of our developed technique in a tabular form. Results shows that the proposed DRALBS algorithm provides enhanced programming and replicating unit in relation to MDHR.

6.3 Contribution of Research

The key contribution of this research is the enhancement in the replication and scheduling techniques in grid i.e. distributed environment. The literature review contributes in identifying the issues and challenges of replication and scheduling approaches to share network resources and easy access of big data scattered geographically. A comprehensive survey and taxonomy of replication existing techniques is also part of the contribution of this work. With the help of this, a comparison between current techniques is performed by using standard parameters. Existing techniques ignores the location of the replica at the time of scheduling. They send jobs to those nodes which fulfil the job execution requirement which then leads to the decision of the replica manager as to whether to assign job to a site or not. The scheduler does not know about the replica for data execution on site. The data execution time and migration rate are increased and causes more usage of bandwidth. The idea to design a model for effective scheduling based on taxonomy, necessitated more study on distributed network environment. There are many techniques developed for overcoming these deficiencies in distributed network. We deeply studied all of them and for comparison purposes choose the recent replication technique known as MDHR. MDHR chooses the best replica area from among the multiple replicas considering the time of response that can persist in the transmission of data, the capacity to get to inertness, the waiting time of replica requests in queue and the separation length between nodes. So a more efficient and effective technique as compared to MDHR is identified for replication scheduling. The effect of a new programming on replication is also the contribution of our work. The significant role of this work is the improvement of replication, conscious load balanced planning, in view of local status and replication in the time of work planning.

Computationally available replication scheduling algorithm were proposed for replica placement and load balancing. For this purpose, we added two modules to the scheduler. The replica location information is provided to scheduler by replica manger and load information gotten from the load manager. The simulation result confirms the system suitability in a distributed environment.

6.4 Research Limitations

The scope of this research is restricted to considering the issue of fault tolerance in grid distributed environment. Although our research has improved results significantly, there is still need to mature this strategy in order to advance our knowledge in this area.

The main limitation of this research is that it does not analyze the problem of the identification of a critical issue of failure in the existing scheduling model. This work only covers the replication job scheduling in a free failure scenario. Therefore, the lack of fault tolerance is one of the main issue related to replication and scheduling technique in distribute environment. This research can be explained with the fault tolerance strategy as well. For example, failure may occur in any computing sites, which may increase the implementation cost and response time exponentially. We have assumed the scenario where no failure occurs. It may also be worthwhile researching variant of the current problem. Investigating the faults and applying the fault tolerant strategies may increase the execution time and the implementation cost a little bit as well but the overall scheduling mechanism will be improved significantly.

6.5 Future research Direction

This research is mainly focused on the development of a replication scheduling technique that can properly schedule a job to an appropriate site with load balanced in a distributed environment. The replication job scheduling method has some challenges in terms of replica placing and selection with load balance on each site. Hence, this research emphasizes on the response time and the data migration ratio of required data for a job, yet a more efficient and intelligent scheduling method is the demand of users to more easily and smoothly accessing of data in distributed nodes. Our research succeeded in improving the system performance and to satisfy users demand. But the work still needs to be improved on in some special scenarios. Therefore, the possible extension of this work is as follows:

• The proposed research method can be extended to the Cloud Computing network.

• We also plan to extend the proposed scheduling and replication technique with fault tolerance strategy in cloud network.

• Lastly, we calculated results in a simulation environment. It is better to have results compilation using our algorithm in a real data grid environment so that we can sense the real time effects of the data grid environment.

REFERENCES

Abdi, S., & Hashemi, S. (2015). A hierarchical approach to improve job scheduling and data replication data grid. International Arab Journal of Information Technology (IAJIT), 12(3).

Abrams, M., Standridge, C. R., Abdulla, G., Fox, E. A., & Williams, S. (1996). Removal policies in network caches for world-wide web documents. In Acm sigcomm computer communication review (Vol. 26, pp. 293–305).

Ali-Eldin, A., & El-Ansary, S. (2011). Replica placement in peer-assisted clouds: an economic approach. In Distributed applications and interoperable systems (pp. 208–213).

Al-Shayeji, M. H., Rajesh, S., Alsarraf, M., & Alsuwaid, R. (2010). A comparative study on replica placement algorithms for content delivery networks. In Advances in computing, control and telecommunication technologies (act), 2010 second international conference on (pp. 140–142).

Amir, Y., Peterson, A., & Shaw, D. (1998). Seamlessly selecting the best copy from internet-wide replicated web servers. Distributed Computing, 22–33.

Andresen, D., Yang, T., Holmedahl, V., & Ibarra, O. H. (1996). Sweb: Towards a scalable world wide web server on multicomputers. In Parallel processing symposium, 1996., proceedings of ipps'96, the 10t international (pp. 850–856).

Apers, P. M. (1988). Data allocation in distributed database systems. ACM Transactions on Database Systems (TODS), 13 (3), 263–304.

Aversa, L., & Bestavros, A. (2000). Load balancing a cluster of web servers: using distributed packet rewriting. In Performance, computing, and communications conference, 2000. ipccc'00. conference proceeding of the ieee international (pp. 24–29).

Baentsch, M., Baum, L., Molter, G., Rothkugel, S., & Sturm, P. (1997). Enhancing the web's infrastructure: From caching to replication. IEEE Internet Computing, 1(2), 18–27.

Banga, G., Douglis, F., Rabinovich, M., et al. (1997). Optimistic deltas for www latency reduction. In Proc. 1997 usenix technical conference, anaheim, ca (pp. 289–303).

Beck, M., & Moore, T. (1998). The internet2 distributed storage infrastructure project: An architecture for internet content channels. Computer Networks and ISDN systems, 30(22), 2141–2148.

Bestavros, A., Crovella, M., Liu, J., &Martin, D. (1998). Distributed packet rewriting and its application to scalable server architectures. In Network protocols, 1998. proceedings. sixth international conference on (pp. 290–297).

Bhattacharjee, S., Ammar, M. H., Zegura, E. W., Shah, V., & Fei, Z. (1997). Application-layer anycasting. In Infocom'97. sixteenth annual joint conference of the ieee computer and communications societies. driving the information revolution., proceedings ieee (Vol. 3, pp. 1388–1396).

Cao, P., & Liu, C. (1998). Maintaining strong cache consistency in the world wide web. IEEE Transactions on Computers, 47(4), 445–457.

Cardellini, V., Colajanni, M., & Yu, P. S. (1999). Redirection algorithms for load sharing in distributed web-server systems. In Distributed computing systems, 1999. proceedings. 19th ieee international conference on (pp. 528–535).

Casas, I., Taheri, J., Ranjan, R., Wang, L., & Zomaya, A. Y. (2017). A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems. Future Generation Computer Systems, 74, 168–178.

Cate, V. (1992). Alex-a global filesystem. In Proceedings of the 1992 usenix file system workshop (pp. 1–12).

Crovella, M. E., & Carter, R. L. (1995). Dynamic server selection in the internet (Tech. Rep.). Boston University Computer Science Department

Dowdy, L. W., & Foster, D. V. (1982). Comparative models of the file assignment problem. ACM Computing Surveys (CSUR), 14(2), 287–313.

Du, Z., Hu, J., Chen, Y., Cheng, Z., & Wang, X. (2011). Optimized qos-aware replica placement heuristics and applications in astronomy data grid. Journal of Systems and Software, 84(7), 1224–1232.

Dykes, S. G., & Robbins, K. A. (2001). A viability analysis of cooperative proxy caching. In Infocom 2001. twentieth annual joint conference of the ieee computer and communications societies. proceedings ieee (Vol. 3, pp. 1205–1214).

Eslami, G., & Haghighat, A. T. (2012). A genetic algorithm for replica server placement. In Fourth international conference on machine vision (icmv 11) (pp. 83493D–83493D).

Fan, L., Cao, P., Lin, W., & Jacobson, Q. (1999). Web prefetching between low-bandwidth clients and proxies: potential and performance. In Acm sigmetrics performance evaluation review (Vol. 27, pp. 178–187).

Fei, Z.-M., Bhattacharjee, S., Zegura, E. W., & Ammar, M. H. (1998). A novel server selection technique for improving the response time of a replicated service. In Infocom'98. seventeenth annual joint conference of the ieee computer and communications societies. proceedings. ieee (Vol. 2, pp. 783– 791).

Gadde, S., Rabinovich, M., & Chase, J. (1997). Reduce, reuse, recycle: An approach to building large internet caches. In Operating systems, 1997., the sixth workshop on hot topics in (pp. 93–98).

Grace, R. K., & Manimegalai, R. (2014). Dynamic replica placement and selection strategies in data grids—a comprehensive survey. Journal of Parallel and Distributed Computing, 74(2), 2099–2108.

Huang, X., & Peng, Y. (2013). A novel replica placement strategy for data center network. In 2012 international conference on information technology and management science (icitms 2012) proceedings (pp. 599–609).

Hunt, G. D., Goldszmidt, G. S., King, R. P., & Mukherjee, R. (1998). Network dispatcher: A connection router for scalable internet services. Computer Networks and ISDN Systems, 30(1), 347–357.

Iyengar, A., & Challenger, J. (1997). Improving web server performance by caching dynamic data. IBM Thomas J. Watson Research Division.

Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., & Zhang, L. (2000). On the placement of internet instrumentation. In Infocom 2000. nineteenth annual joint conference of the ieee computer and communications societies. proceedings. ieee (Vol. 1, pp. 295–304).

Jamin, S., Jin, C., Kurc, A. R., Raz, D., & Shavitt, Y. (2001). Constrained mirror placement on the internet. In Infocom 2001. twentieth annual joint conference of the ieee computer and communications societies. proceedings. ieee (Vol. 1, pp. 31–40).

Khan, S. U., & Ahmad, I. (2010). Replicating data objects in large distributed database systems: an axiomatic game theoretic mechanism design approach. Distributed and Parallel Databases, 28(2-3), 187–218.

Kumar, K. A., Deshpande, A., & Khuller, S. (2013). Data placement and replica selection for improving co-location in distributed environments. arXiv preprint arXiv:1302.4168.

Kumar, N., & Kim, J. (2013). Probabilistic trust aware data replica placement strategy for online video streaming applications in vehicular delay tolerant networks. Mathematical and Computer Modelling, 58(1), 3–14.

Litke, A., Skoutas, D., Tserpes, K., & Varvarigou, T. (2007). Efficient task replication and management for adaptive fault tolerance in mobile grid environments. Future Generation Computer Systems, 23(2), 163–178.

Little, T. D., & Venkatesh, D. (1995). Popularity-based assignment of movies to storage devices in a video-on-demand system. Multimedia Systems, 2(6), 280–287.

Loukopoulos, T., Ahmad, I., & Papadias, D. (2002). An overview of data replication on the internet. In Parallel architectures, algorithms and networks, 2002. i-span'02. proceedings. international symposium on (pp. 31–36).

Loukopoulos, T., Lampsas, P., & Ahmad, I. (2005). Continuous replica placement schemes in distributed systems. In Proceedings of the 19th annual international conference on supercomputing (pp. 284– 292).

Loukopoulos, T., Tziritas, N., Lampsas, P., & Lalis, S. (2007). Implementing replica placements: feasibility and cost minimization. In Parallel and distributed processing symposium, 2007. ipdps 2007. ieee international (pp. 1–10).

Mansouri, N. (2014). A threshold-based dynamic data replication and parallel job scheduling strategy to enhance data grid. Cluster computing, 17(3), 957–977.

Mansouri, N. (2016). Qdr: a qos-aware data replication algorithm for data grids considering security factors. Cluster Computing, 19(3), 1071–1087.

Mansouri, N., & Dastghaibyfard, G. (2014). Improving data grids performance by using modified dynamic hierarchical replication strategy. Iranian Journal of Electrical & Electronic Engineering, 10(1), 27.

Mansouri, N., & Dastghaibyfard, G. H. (2013). Enhanced dynamic hierarchical replication and weighted scheduling strategy in data grid. Journal of Parallel and Distributed Computing, 73(4), 534–543.

Mansouri, N., Dastghaibyfard, G. H., & Mansouri, E. (2013). Combination of data replication and scheduling algorithm for improving data availability in data grids. Journal of Network and Computer Applications, 36(2), 711–722.

Morris, J. H., Satyanarayanan, M., Conner, M. H., Howard, J. H., Rosenthal, D. S., & Smith, F. D. (1986). Andrew: A distributed personal computing environment. Communications of the ACM, 29(3), 184–201.

Narendran, B., Rangarajan, S., & Yajnik, S. (2000, May 30). Data distribution techniques for loadbalanced fault-tolerant web access. Google Patents. (US Patent 6,070,191)

Padmanabhan, V. N., & Mogul, J. C. (1996). Using predictive prefetching to improve world wide web latency. ACM SIGCOMM Computer Communication Review, 26(3), 22–36.

Peng, G. (2004). Cdn: Content distribution network. arXiv preprint cs/0411069.

Pitoura, T., Ntarmos, N.,&Triantafillou, P. (2012). Saturn: range queries, load balancing and fault tolerance in dht data systems. IEEE Transactions on Knowledge and Data Engineering, 24(7), 1313–1327.

Rajaretnam, K., Rajkumar, M., & Venkatesan, R. (2016). Rplb: A replica placement algorithm in data grid with load balancing. International Arab Journal of Information Technology (IAJIT), 13(6).

Shahriari, M., Biglarbegian, M., Melek,W., & Kurian, J. (2016). A new computationally efficient dynamic scheduling for multiple resource constrained procedures. In Evolutionary computation (cec), 2016 ieee congress on (pp. 2633–2640).

Souli-Jbali, R., Hidri, M. S., & Ayed, R. B. (2015). Dynamic data replication-driven model in data grids. In Computer software and applications conference (compsac), 2015 ieee 39th annual (Vol. 3, pp. 393–397).

Sun, J., Gao, S., Yang,W.,&Jiang, Z. (2011). Heuristic replica placement algorithms in content distribution networks. Journal of networks, 6(3), 416–423.

Tziritas, N., Loukopoulos, T., Lampsas, P., & Lalis, S. (2008). Formal model and scheduling heuristics for replica migration problem. Euro-Par 2008–Parallel Processing, 305–314.

Vashisht, P., Kumar, R., & Sharma, A. (2014). Efficient dynamic replication algorithm using agent for data grid. The Scientific World Journal, 2014.

Wang, T., Yao, S., Xu, Z., & Pan, S. (2017). Dynamic replication to reduce access latency based on fuzzy logic system. Computers & Electrical Engineering, 60, 48–57.

Yoshikawa, C., Chun, B., Eastham, P., Vahdat, A., Anderson, T., & Culler, D. (1997). Using smart clients to build scalable services. In Proceedings of the 1997 usenix technical conference (p. 105).

Zaman, S., & Grosu, D. (2011). A distributed algorithm for the replica placement problem. IEEE Transactions on Parallel and Distributed Systems, 22(9), 1455–1468.