A MODEL FOR FEDERATED DATA GRID SYSTEMS BASED ON NETWORK CORE AREA

ZARINA BT MOHAMAD

A thesis submitted in

fulfillment of the requirement for the award of the

Doctor of Philosophy

Faculty of Computer Science and Information Technology

University Tun Hussein Onn Malaysia

FEBRUARY 2013

ABSTRACT

Data grid technology is developed to permit data sharing and collaboration activities across many organizations in geographically dispersed locations. Due to collaboration activities in distributed data grid, optimising data access is one of the issues that needs to be carefully addressed. The process of data access optimisation can be achieved through data replication, where identical copies of data are generated and stored at various globally distributed sites. If the required files are replicated in some site where the job is executed, then the job is able to process data without any communication delay. In this thesis, a model for federation data grid system called *Sub-Grid-Federation* had been proposed to improve access latency by accessing data from the nearest possible sites. A strategy in optimising data access is based on the process of searching into the area identified as *'Network Core Area'* (NCA). The proposed system will require access to be initially restricted from within the local site. If the needed data is not found, the system will continue the search from within a bigger scope, known as the local cluster. If the search still failed, the scope of search will continue from within the local sub-grid and other sub-grids consecutively until the data is successfully accessed. The performance of access latency in *Sub-Grid-Federation* had been tested based on the mathematical proving and simulated using *OptorSim* simulator. Four case studies had been chosen and tested in Optimal Downloading Replication Strategy (ODRS) and the *Sub-Grid-Federation*. The results showed that *Sub-Grid-Federation* is 20% better in terms of access latency and 21% better in terms of reducing remote sites access compared to ODRS. Hence, *Sub-Grid-Federation* is a better alternative for the implementation of collaboration and data sharing in data grid system.

ABSTRAK

Teknologi grid data dibangunkan untuk membolehkan perkongsian data dan aktiviti-aktiviti kolaborasi merentasi organisasi dalam lokasi geografi yang berbeza. Di dalam konteks kolaborasi yang berkesan, mengoptimumkan akses data adalah salah satu isu yang perlu diambil kira dengan berhati-hati. Proses pengoptimiman akses data boleh dicapai melalui replikasi data, di mana salinan data yang serupa dihasilkan dan disimpan dipelbagai tapak yang teragih di seluruh dunia. Jika fail-fail yang diperlukan direplika di beberapa tapak di mana kerja itu dilaksanakan, maka kerja itu mampu untuk memproses data tanpa sebarang kelewatan komunikasi. Dalam tesis ini, sebuah model sistem grid data persekutuan yang dipanggil *Sub-Grid-Federation* telah dicadangkan untuk meningkatkan masa capaian dengan mengakses data pada pelayan yang terdekat. Strategi dalam mengoptimumkan akses data adalah berasaskan kepada pencarian ke kawasan yang dikenali sebagai *'Network Core Area*' (NCA). Skim yang dicadangkan akan memerlukan akses yang pada mulanya terhad dari dalam tapak tempatan sahaja. Jika data tidak dijumpai, sistem akan terus mencari dari dalam skop yang lebih besar, yang dikenali sebagai kluster tempatan. Jika carian masih gagal, skop carian akan diteruskan ke dalam sub-grid tempatan dan sub-grid lain berturut-turut sehinggalah data itu berjaya dicapai. Prestasi tempoh masa akses bagi *Sub-Grid-Federation* telah diuji berdasarkan pembuktian matematik dan simulasi menggunakan simulator *OptorSim*. Empat kajian kes telah dipilih dan diuji dalam *Optimal Downloading Replication Strategy (ODRS)* dan *Sub-Grid-Federation*. Hasil kajian menunjukkan bahawa *Sub-Grid-Federation* adalah 20% lebih baik dari segi tempoh masa akses dan 21% lebih baik dari segi mengurangkan akses pada pelayan yang jauh berbanding dengan ODRS. Oleh itu, *Sub-Grid-*

Federation adalah satu alternatif yang terbaik bagi pelaksanaan kolaborasi dan perkongsian data dalam sistem grid data.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

M	-	Number of sites in the system
Q	-	Number of clusters in the system
G	-	Number of sub grids in system
N	-	Number of unique files in the system
K	-	Storage space at each site (GB)
λ	-	lambda
ρ	-	rho
t_l	-	costs when accessing a replica file from a site's local storage space
t_o	-	costs when accessing a replica file from a remote site of the same cluster
t_g	-	costs when accessing a replica file from remote site of other cluster but within the same sub grid
t_G	-	costs when accessing a replica file from remote site of other sub grid
NCA	-	Network Core Area
ODRS	-	Optimized Downloading Replication Strategy
LAN	-	Local Area Network
WAN	-	Wide Area Network
SGFRS	-	*Sub-Grid-Federation* Replication Strategy
SGFS	-	*Sub-Grid-Federation* Scheduling
CERN	-	European Organization for Nuclear Research
FTP	-	File Transfer Protocol
p2p	-	Peer-to-Peer

RO	-	Replica Optimizers
RM	-	Replica Manager
CE	-	Computer Element
SE	-	Storage Element

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter introduces the context of the research to be presented in this thesis. It starts off with an introduction to the general area of grid computing and data grids. This is followed with an explanation of the fundamentals of job scheduling and data replication in data-intensive environments. Then, the objectives and the scope of the research are presented. The chapter ends with a discussion on the organisation of the thesis.

1.1 Background study

Many researchers seek new discoveries of grid computing as it introduces a large variety of applications. Typically, a grid consists of a large number of heterogeneous resources with multiple domains of organization for application which require a large sum of storage resources and computational. Thus, these large scale applications require large amounts of storage, high speed networking, good web technologies and high-end computing to facilitate collaboration and data-intensive in scientific research (Antunes, G. & Pina Helder H. 2011). Foster I., Kesselman C. & Tuecke S. (2001) states that distributed heterogeneous resources such as databases, scientific instruments and computers will be available for selection, discovery, exchange, sharing and aggregation on a grid platform. Grid computing focal point is the capability to provide for multiple organizational domains computational hence will produce a grid computing without a limit in regards to the number of organizations, departments or users and different kinds of applications such as compute intensive applications, data intensive applications and distributed services applications (K. Sashi & A. S. Thanamini, 2010).

A data grid can access and manage a huge sum of sets of data amounting up to terabytes or petabytes that depend on the requirement of the projects. The up and coming trend in scientific applications, like those found in science and engineering, demonstrate that huge amounts of data sets are processed and produced by these applications. Data grids support the required functionality of a large number of data in data grid applications, for instance, security, user management, resource management, resource discovery, job scheduling, data replication, high speed network protocols, and data management (Srikumar V, Rajkumar B., & Kotagiri R., 2006). Thus for more analysis and sharing among collaborating research around the globe among the scientific community the produced data requires to be stored.

Collaboration means "togetherness" that involves several types of data sharing between organizations in the data grid. Enhanced Science or e-Science (Hey T. & Trefethen, 2002) is a scientific community for the collaborative environment that involves the requirement of interoperability and data sharing respectively (Miles at el., 2007). In other words, e-Science relates to the set of services, techniques, personnel and organizations, which have become a collaborative network (Antunes, G. & PinaHelder, 2011). Grids can be organized in different ways. In particular, grids can be federated with each other. The federated model allows grids to fit into different institutions with independent administration and different locations that are interoperable with each other so that data can be shared. In data grid, collaboration is important for an organization to share their research. Moreover, the concepts of collaborations are rapidly emerging in a cloud that is called cloud federation which shares resources in order to fulfil user demand (I. Goiri et al, 2010 & Xiaoyu Yang et al., 2012).

Figure 1.1 illustrates the distributed cluster environments that include three scientific research groups working on collaborative scientific projects. An individual research institute organises its own resources using a local broker which connects to the other cluster broker. Various cluster brokers are connected in hierarchy to form a distributed scheduling architecture. A cluster user submits jobs to either its centralised local resource broker or to the cluster broker. As the number of clusters increases, a hierarchical connection between clusters imposes serious performance limitations. As a result, this solution is imperfect and the federation data grid model can be used for collaboration activity.

.

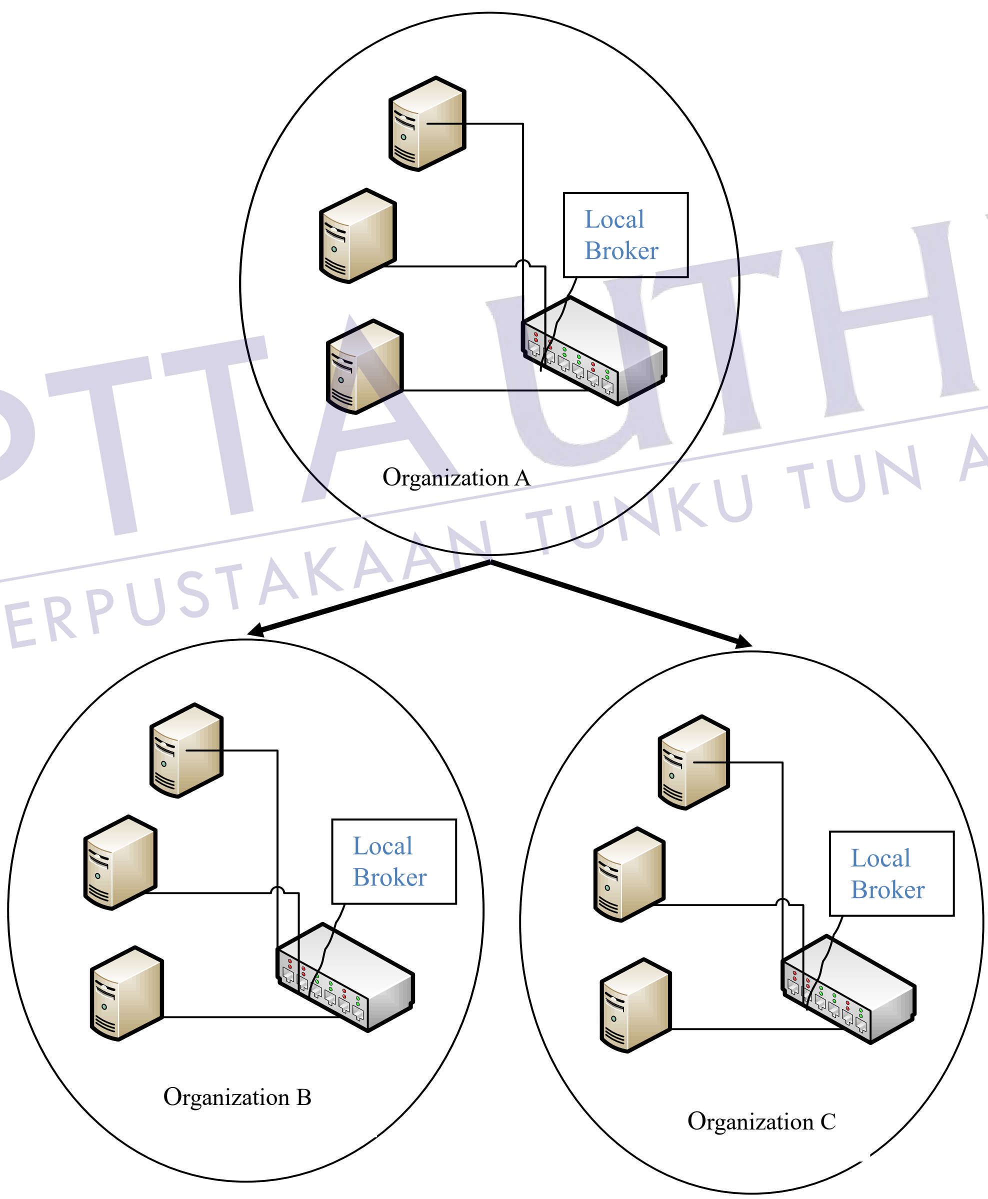

Figure 1.1: A hierarchical cluster grid architecture

Figure 1.2 shows four clusters in this data grid system. The sites in a cluster connect in a Local Area Network (LAN) and the sites among clusters connect in a Wide Area Network (WAN). These refer to two types of communication between sites in this data grid system. The first type of communication is the intra-communication which refers to the site to site in same cluster communication (LAN connection). The second type of communication is the inter-communication which refers to the communication between sites across clusters (WAN connection). Accessing large datasets from remote sites will affect high access latency. Consequently, the bandwidth of networks between sites across clusters will always be lower than within clusters themself which causes traffic bottlenecks in the network channels and congestion in cluster inter-communicating. It is crucial to decrease the total number of inter-communications for reduction of access latency and prevention of WAN bandwidth congestion.

To improve access performance and decrease inter-communications, two effective techniques should be consider in data grid environment namely job scheduling and data replication. Job scheduling is a mapping that transmits a set of jobs into a set of sites for executions that minimize the time of job execution. All jobs require data for their execution. Therefore, the data transfer time should be taken into consideration when a job is scheduled to a site. Furthermore, data replication will minimize job execution time in total by means of reducing bandwidth consumption and transfer time by creating numerous copies of popular data in a data grid system (P. Liu, Y. Lin & J. Wu, 2008). Both of these techniques complement each other; scheduling alone with no replication involved would add extra time to data transfer time where the job has to fetch remotely and replication without considering job scheduling would result in ineffectiveness by transferring a large size of data which requires more bandwidth. Thus, the combination of dynamic replication strategy and job scheduling algorithm can reduce access latency and indicate high performance in data grid (R. Lakshmi and Bin Tang, 2011).

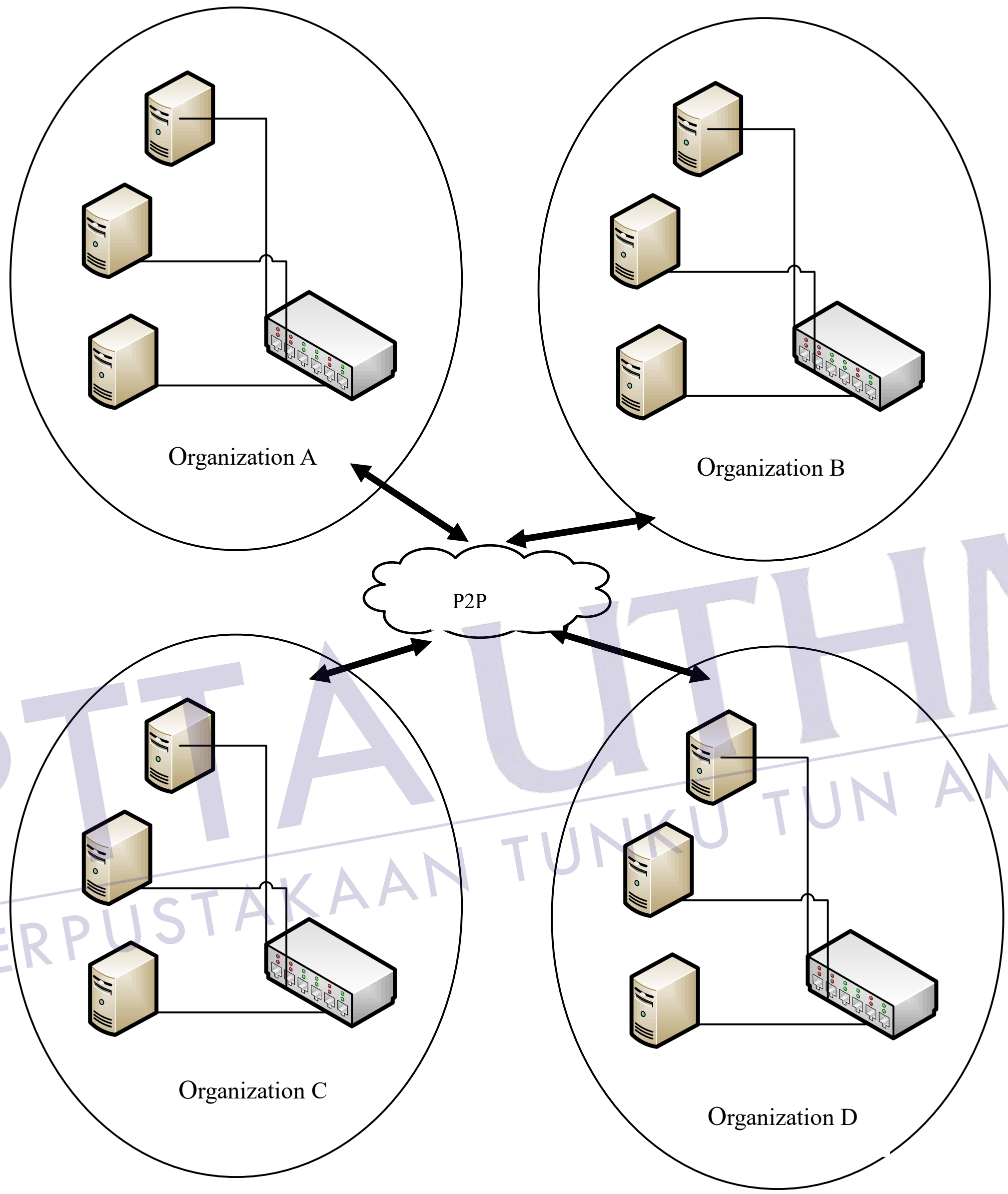

Figure 1.2: Federation resource sharing system

1.2 Problem statement

As discussed in previous section of this thesis, the benefit on employing the federated data grid environment is apparent when collaboration among peers of grid providers is required. The existences of single or centralized coordinator in hierarchical data grid limit the process of collaboration among grid providers. Therefore, the concept of federated data grid environment is employed where each data grid peers have the same level of authority in sharing and collaborate their resources (i.e., the need for central coordinator is eliminated). However, this advantage comes with a price of complexity in addressing access latency issue, especially when the federated data grid also inherits two main characteristics of general data grid environment i.e., 1) it involved a massive number of machines, and 2) it is dynamic in nature. This issue is generally prominent in the aspect of data replication and job scheduling. In such situation, jobs need to be assigned and processed by the best resource (which holds the respective replica data) before the result can be returned to the user. Job scheduling and data replication are two most important issues that are difficult to resolve when the above mentioned characteristics need to be addressed. From the perspective of data replication, these characteristics will undoubtedly increase the complexity of replication process where the best number and location of replica data need to be chosen carefully to ensure the efficiency of replication scheme is preserved. Furthermore, from the view of job scheduling, the existence of these characteristics in data grid environment will shoot up the difficulty of job scheduling process. In such scenario, an efficient mechanism is required so that the best resource (which holds the respective replica data) can be chosen and jobs can be processed by this resource before the result can be returned to the user to reduce access latency.

1.3 Objectives of thesis

Generally, the main objective of this thesis is to develop a model for a federated data grid that can efficiently minimize access latency. This can be done by integrating the replication strategy and job scheduling algorithm. It reduces WAN bandwidth congestion by decreasing the number of network communications between grid regions. Specifically, the study aims to achieve the following objectives:

(i) To propose a model for federated data grid systems.

(ii) To propose an efficient data replication strategy for the proposed model.

(iii) To simulate the proposed model using data grid simulator.

1.4 Scope of thesis

The scope of this thesis is to design and develop a data replication and job scheduling model so that jobs can be scheduled to the best replica data in federated data grid. The main focus is to minimize the access latency by employing an efficient data replication strategy. The data replica is accessed by using scheduling algorithm based on "*Network Core Area*" (NCA). The concept of NCA refers to the dynamic strategy of finding the scope for resource locality which starts from local site, local cluster, local sub grid and other sub grid consecutively. The proposed model concentrates on the integrated strategy of data replication and job scheduling algorithm. Further, the analytical proving concept is used in proving the correctness of the designed and developed model. Finally, this model is simulated in grid simulator in order to evaluate its performance.

1.5 Importance of study

In a data grid environment, a set of replica files can be retrieved from various sites. If the required files are replicated on a site where the job has been executed, then this job processing will be completed in this site without delay. On the other hand, if the required files are unavailable within the job execution site, job scheduling process will be triggered and this job will be processed from a remote site. In such situation, a longer time is required in this scheduling process due to a huge replica size, where in some applications, it size may reach to gigabytes in scale. Undoubtedly, the limitation of network bandwidth between grid sites cannot be avoided. As a result, the process of re-scheduling for replicas via the internet connection will introduce unnecessary delay which in turn will cause a longer job execution time. To highlight, the main contribution of this thesis is that it reduces data access time when jobs are processed in federated data grid environment.

1.6 Thesis organization

This thesis consists of five chapters which includes this introductory chapter. The remaining four chapters are organized as follow:

Chapter 2 describes a general overview of data grids which covers some important and relevant topics such as layered architecture and general data grid model. This chapter presents the fundamental concepts and differences between the existing data grid models such as monadic, hierarchy, P2P and federation. Finally, this chapter discusses some important researches in data grids which further focuses on the categorization of data replication strategy and scheduling algorithm for data-intensive applications. A clear understanding is required on these concepts to ensure that the research questions can be addressed efficiently.

Chapter 3 discusses the proposed model of *Sub-Grid-Federation*. The *Sub-Grid-Federation* system is defined as a collaboration and data sharing system for heterogeneous data grid. The replication strategy for this model is presented and proved analytically. The objective of the proposed model is to minimize access latency by allowing data to be accessed from the nearest site in the federated data grid.

Chapter 4 describes the implementation and evaluation of the proposed model based on simulation. The performance of the proposed model is compared against two existing model, i.e. SGFRS and ODRS, by using *OptorSim* simulator tool. Detail discussions on the simulation result are presented. Finally, this chapter is summed up by discussing the testing which has been done by using this simulation environment.

Chapter 5 discusses three main contributions of the thesis, i.e. 1) *Sub-Grid-Federation* model, 2) the concept of NCA in *Sub-Grid-Federation* Replication Strategy, and 3) *Sub-Grid-Federation* Scheduling. Finally, some potential directions of future research issues are also presented this chapter. The discussion allows further exploration of interesting research areas which are closely related to the focus of this thesis.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a general overview of data grids that covers topics such as a layered architecture and data grid model. This chapter presents an analysis of the differences between data grid models such as monadic, hierarchy, P2P and federation. It ends with a discussion on research in data grids into specialised areas and categories in data replication strategy and scheduling algorithm for data-intensive.

2.1 Introduction

Nowadays a huge amount of data is produced in many fields such as scientific discoveries and engineering applications. These generated data should be distributed in wide area networks to enable data sharing and collaboration. Hence, efficient management of these tremendous resources becomes an important topic for scientific researches and other institutional applications. The data grid offers solution for this problem by constructing a scalable and distributed system by connecting various resources from different locations, organizations and multiple heterogeneous platforms. This environment empowers users to have transparent access to those resources as if they are accessing the local sites. Various grids have been developed which can be classified into computational grids and data grids. Computational grids are for managing and dealing with computational intensive tasks, while data grids are for data sharing and collaboration.

2.2 Data grid

Data grids help users to search large amount of data, modify and transfer them according to their specific needs. Users can also create and manage copies of these data, and store them in distributed repositories (Srikumar V. et al., 2006). There are two basic functionalities of a data grid: 1) a scalable replication discovery and management mechanism 2) a high performance and reliable data transfer mechanism (Chervenak A. et al., 2000). However, the replication discovery and high performance data transfer depend on the requirements of an application as a provision for a range of other services that need to be fulfilled. Replica management consistency, filtering of data and management of data are some instances of services. To ensure requests are authenticated and only authorized operations are conducted, all data grid operations are mediated by a security layer.

Maintaining the huge shared dataset collections in distributed system is an additional facet of a data grid. In a data grid, dataset are freely maintained, without concerning the kind of storage system underneath. The inclusion of a new site is also easily achieved without much effort. More importantly, the establishment of persistent archival storage is compulsory where the data associated with metadata, access controls, and version changes will be well-maintained during the platform change (Moore R., Rajasekar A. & Wan M., 2005).

A high-level view of worldwide data grid in different countries is shown in Figure 2.1 which consists of high speed network links comprising of computational and storage resources. Major centres are linked with high bandwidth networks represented by a thick line, and lower bandwidth networks linking to subsidiary centres are illustrated by thinner lines. Each principal storage site stores data created from an experiment, instrument or a sensors network. Through data replication mechanism, responding to a request, data is conveyed to a storage space of another site around the globe.

In the process of locating the required datasets, firstly, users will query a local replication catalogue. Data from local repository will be retrieved to their area whenever the permissions and requisite rights are granted. Otherwise, if the required data is not available in local sites, retrieval of data from a remote repository needs to be performed. Subsequently, for the purpose of processing, data might be sent out to a cluster or a facility of supercomputer, or generally known as a computational site. After being processed, the result might then be transmitted to facilitate visualization, sharing or even to be stored the local workstation.

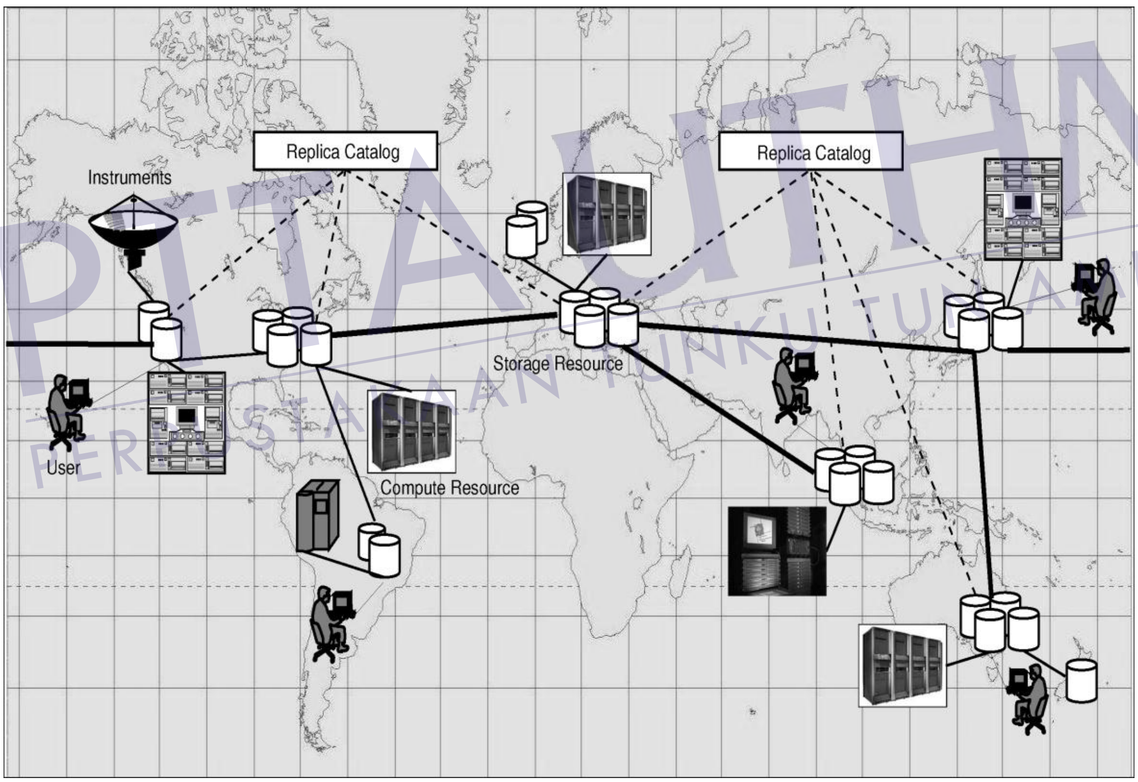

Figure 2.1: A high-level view of data grid (Srikumar V. et al., 2006)

2.3 Data grid architecture

The components of data grids consist of applications, data grid services, communication and basic grid fabric (Srikumar V. et al., 2006). The layered structure is depicted in Figure 2.2. Each layer has its own features and being described as follows:

(i) Application

This layer provides services to users by calling the services provided by the inner layers of the architecture. Then, they are modified to match the target domain such as replica management and resource brokering.

(ii) Data grid services

The core-level in data grid services like data replication, job scheduling and data discovery provide transparent access to dispersed data while replica management and resource brokering involve mechanisms that permit efficient management of resource buried behind smart commands and APIs in user level services.

(iii) Communication

In this layer, the resources querying from the grid fabric layer use these related protocols to develop on the essence of communication protocols, for example TCP/IP and authentication protocols such as Public Key Infrastructure (PKI), Secure Sockets Layer (SSL) or passwords. In ensuring the integrity and security of the transferred data, users‟ identities are authenticated by means of cryptographic protocols.

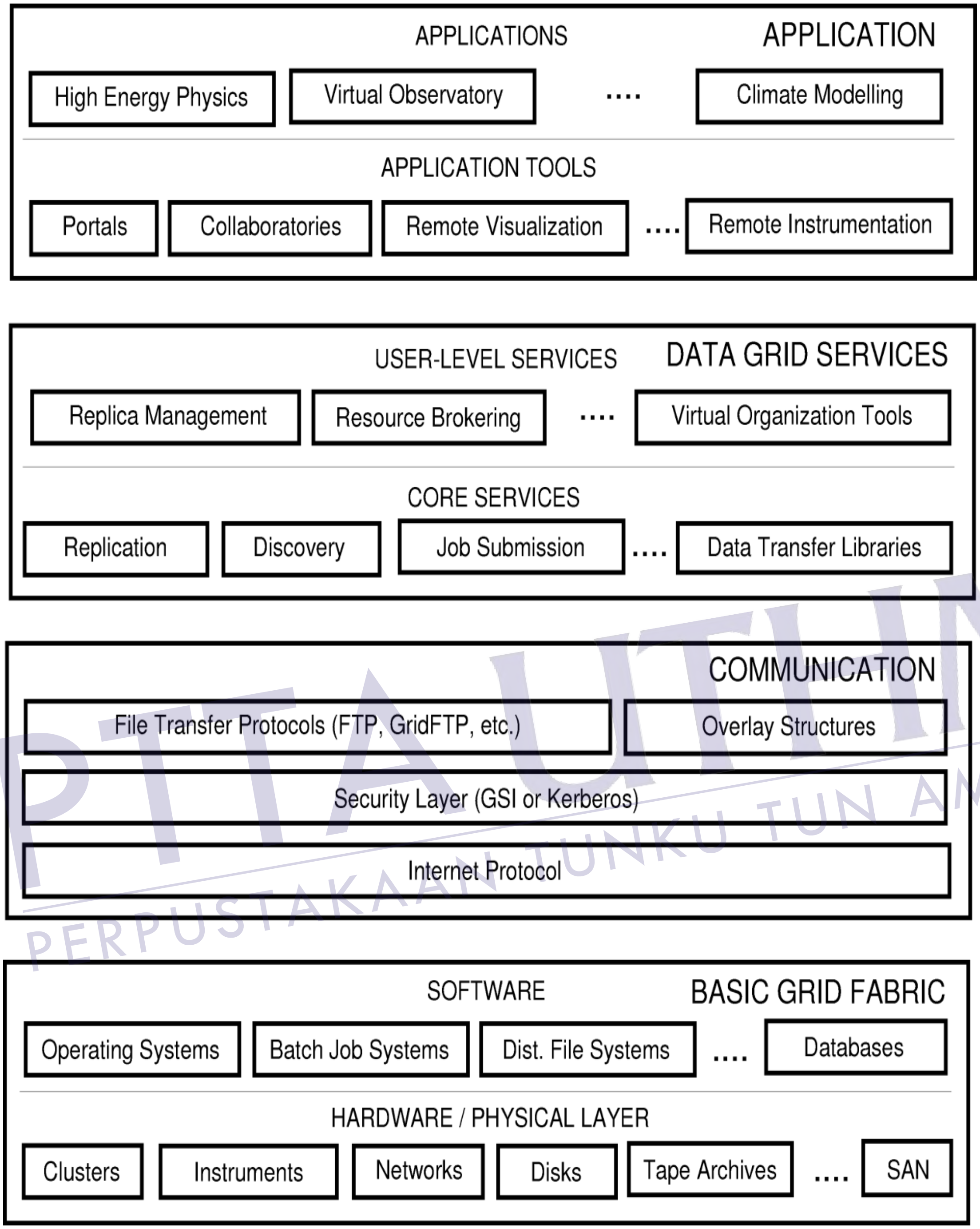

Figure 2.2: A layered Architecture (Srikumar V. et al, 2006)

(iv) Basic Grid fabric

This layer is classified into hardware and software layers. The hardware or physical layer contains the dispersed computational resources such as storage resources, supercomputers and clusters linked up together by high bandwidth networks. Every individual resource executes specific applications (software layer) such as Relational Database Management Systems (RDBMS), Job Submission and Management Systems, and Operating System.

2.4 Data grid model

The design of the data grid model is developed according to the structure of data sources in a system. Depending on the data source, distributed or single, data size and sharing mode, numerous models are available for data grid operation. Figure 2.3 – Figure 2.6 show four models which are commonly established in a data grid environment.

(i) The centralised or monadic model

This data grid model represented in Figure 2.3 is a common form that acts as a central repository where all data are resided and all user queries are served from. Sources of data can be from anywhere in the grid for instance from sensor networks and/or distributed instruments. Then, data will be accessible via interfacing mediums such as web portal and system application where authorization and verification of users will be performed.

The centralised model is different from other models because there is only a single data resource in the structure. The central repository replicates only for the benefit of fault tolerance where the locality of data will not be impacted. Therefore it will be suitable in a situation where data access will not compensate replication overhead.

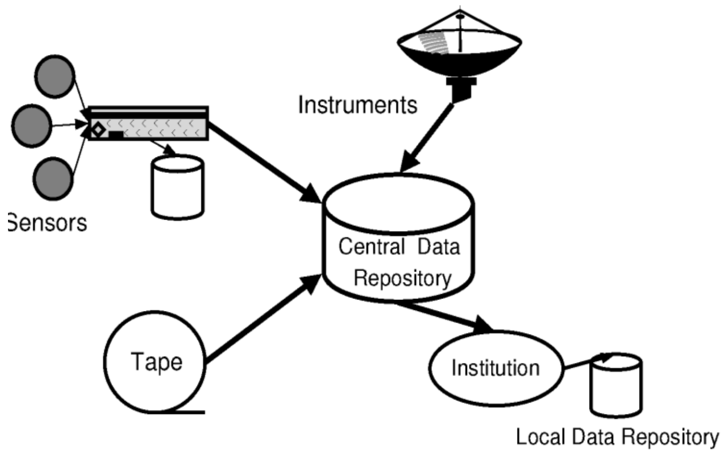

Figure 2.3: Centralised or monadic model (Srikumar V. et al, 2006)

(ii) Hierarchical model

Figure 2.4 shows the hierarchical model operating in a single source where data has to be dispersed among collaborative organisations in the globe. The source is from root as Tier 0, where the data is created. The sub-tiers are the Tier 1 which is called national centres, the Tier 2 named regional centres, Tier 3 workstation and lastly Tier 4, which involves computers from thousands of end users (Foster I & Kesselman, 1998 & Q. Rasool, J. Li & S. Zhang, 2009).

The massive amounts of data resided in the sites motivate the need for robust data distribution mechanism. However, achieving this target is hard because when new clusters connect to the hierarchy, performance of the system become degraded. This situation occurs because hierarchical model cannot transfer data among sibling sites or sites situated on the same tier. Apart from the disadvantage, there is one main advantage of this model which is easier to maintain data consistency as all data are kept in a single source (root/Tier 0).

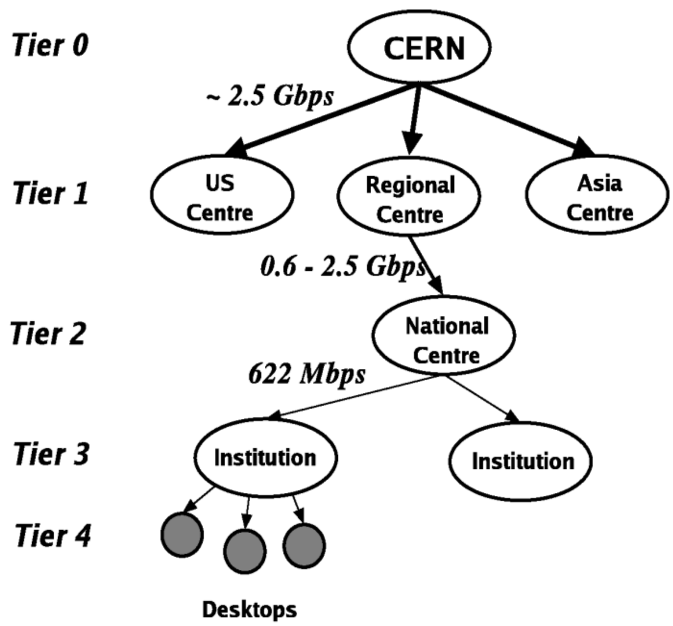

Figure 2.4: Hierarchical model (Srikumar V. et al, 2006)

(iii) The P2P model and federation model

Numerous methods have been recommended to resolve the grid resource discovery problems which include hierarchical server and centralized server approaches. Nevertheless, these two approaches suffer some aspects of network congestion, scalability and fault tolerance limitations. In the last few years, a decentralized network model for example P2P has dynamically been proposed to overcome these limitations. A central server is not used to keep track of files in this decentralized P2P model. Alternatively, files are distributed from one peer to other neighbouring peers. However, federation data grid is organized into cluster-domain peer to peer and based on P2P network (Q. Rasool, et al., 2007) as presented in Figure 2.5. With an appropriate authentication within the federation, data from any databases can be requested by the users of the participating institutions. Each institution maintains various local database integration degrees.

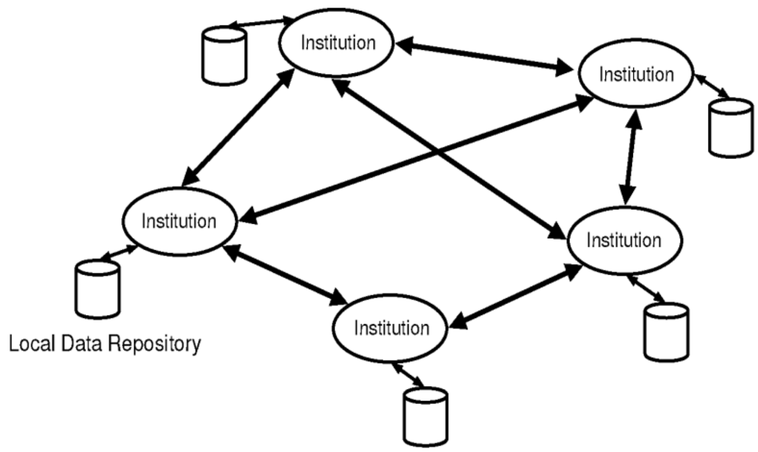

Figure 2.5: Federation model (Srikumar V. et al, 2006)

(iv) Hybrid model

The hybrid model combines all the centralised, hierarchical, and federated models is beginning to emerge as data grids mature and used in the industries. As a result, there is a requisite for collaboration and sharing of their resources among researchers. One of the examples of hybrid model is shown in Figure 2.6 which represents hierarchical data grid and peer edges linkages.

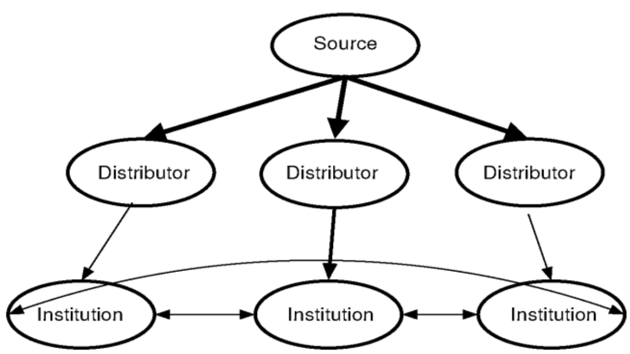

Figure 2.6: Hybrid model (Srikumar V. et al, 2006)

2.5 Data replication

A data grid is a collaboration of a geographically-distributed datasets where each and every collaborative member requires accessing the produced dataset. Datasets replication is a prerequisite for bandwidth preservation, and also a necessity in ensuring collaboration scalability and data access reliability (M. Tang et al., 2005 & Nazanin saadat & Amir M. R, 2011). Moreover, the available storage size and the bandwidth between sites may affect the performance of replication (Figueira & Trieu, 2008). Thus, a management system of replica guarantee requires data assessment, and at the same time performing underlying storage management. Figure 2.7, shows a management system of replica, consists of storage sites or sites connections using high-performance protocols of data transport. Depending on users‟ demands, the creation and management of replicas is performed by the replica manager and the storage availability. Also, a catalogue or a directory keeps track of their replica locations. The applications request for the catalogue to find the availability of replicas for the particular datasets, their number and its location. Sometimes, a merged single entity of the catalogue and the manager are also found in some systems. Generally, software of the client side will include a library which normally incorporates into applications and queries the catalogue that locates datasets and requesting replication for the specific dataset. Strategy of replication and system architecture are the vital elements for the mechanism of replication.

Minimizing file access time is the target for replica optimization. This minimization of file access time can be done by directing the requested access to appropriate replicas (Ming Tang et al., 2006) and referring to the collected statistical data of accessed, the repeatedly used files are proactively replicated. Generally, the mechanism for replication will decides replication of which files, time for performing new replication, and the location site of the new replicas. Static and dynamic are classifications for a replication scheme. As suggested by its name, a replica in static replication remains in the system waiting to be removed by the user or its reached expiration limit. The Static replication shortcoming is when there is a significant change in client access patterns. As a consequence, the advantages of data replication will decrease sharply. On the other hand, dynamic replication automatically builds and removes replicas according to the changes of access patterns. This is to ensure the benefits of replication continue although there are the changes of users‟ behaviour to form the popular data (Lamehamedi H, Shentu Z. & Szymanski B., 2003 & Ruay-Shiung C. & Hui-Ping, 2008 & Muhammad Bsoul et al. 2011).

Assessment of popular data will always create hotspots. These hotspots were reduced by dynamic replication strategies which also promote load sharing. This benefit however, comes with extra overhead costs. System log efficiency maintenance for handling grid environment dynamicity and network traffic real time assessment generates additional overhead.

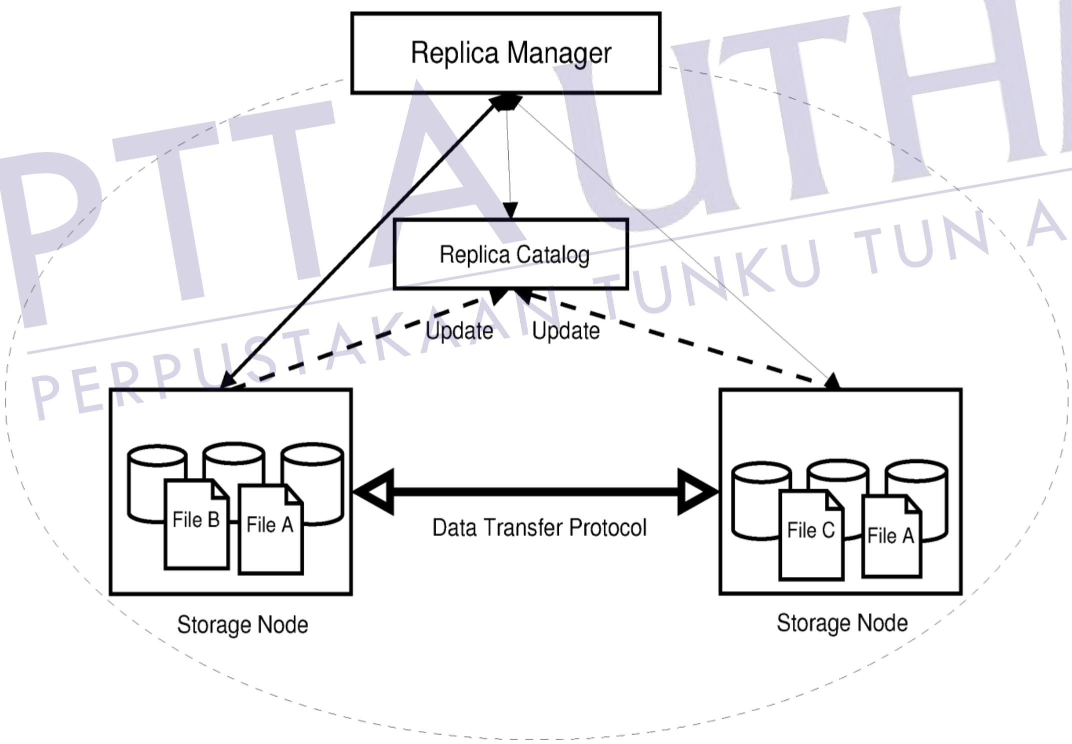

Figure 2.7: A replica management architecture (Srikumar V. et al, 2006)

2.6 Job scheduling for data replication

Data intensive jobs scheduling differ from computational jobs. This is due to the large dataset requirements and the existence of these dataset‟s multiple replicas that are scattered and located over a geographically-distributed region. The availability of bandwidth and transfer latency amongst computational sites has to be taken into account by the schedulers to retrieve the required data from the storage resource(s) (Ruay-Shiung C., Jih-Shing C. & Shin-Yi L., 2007 & Yun-Han L., Seiven L. Rauy-Shiung C., 2011). Hence, the scheduler must know the existence of replica near to the computation point. Besides, the scheduler would produce a new replica of data, if the replication is combined with the scheduling. Types of data-intensive applications scheduling is described below:

(i) Application model

Strategies of scheduling can be defined as an application model to retrieve required data from any storage resource. The composition or distribution of an application is done for global grids scheduling, ranging from specific levels such as process to move general levels such as single tasks to groups of tasks like workflows. For the independent computation unit, a task is regarded as the smallest unit. Every level has its own requirement for scheduling.

(ii) Process-oriented

At the process level, the data is manipulated by applications such as programs of Message Passing Interface (MPI) that are executed over global grids (Foster I. and Karonis 1998). Individual scheduling are made independent tasks by having different objectives that ensure each of the independent tasks gets their share of resources which they require. This depends on certain regular restrictions, for instance the entire application deadline; all tasks from a group of independent tasks which belongs to a *Bag of Tasks (BoT)* application must be successfully executed.

(iii) Scope

The description of scope in scheduling context refers to the application‟s extent of a data grid scheduling strategy. If the strategy of scheduling‟s only is concern to meet the objectives from a perspective of the user, the scope is individual. Each scheduler will own an independent view of the desired resources which it intends to utilize in a multi-user atmosphere. A scheduler is aware of fluctuations in resource availability caused by other schedulers submitting their jobs to common resources. Then it strives to schedule jobs on least-loaded resources that can meet its objectives.

(iv) Data replication

Coupling scheduling of jobs with replication of data is the next classification. Presume that a particular job execution is scheduled at a certain computing site. In the event that job scheduling and data replication are coupled together and remote storage data fetching has to be done, the scheduler will create, at the computational site, a copy of the data which will enable a quicker response for the request of the same file, from a computing site neighbourhood, in the future. As a result, every single job which deals with that particular data will always be scheduled to that particular computing site in the future. However, a computing site is required to have sufficient capacity of storage to store all the replicas of data which will be managed by storage management schemes such as First in First out Least (FIFO) and Recently Used (LRU). As a consequence of lack of storage space, the probability of disregarding the promising computational resources might exist. Also, the process to create and register a replica into a catalogue slows down the processing time to execute a job. For a decoupled scheduler, scheduling of job is done to an appropriate computational resource and then the job identifies a location of replica, most appropriate and suitable to send a request for the required data. A Study of decoupled strategies by Raganathan K. and Foster I. (2002a) has revealed that decoupled strategies tend to increase performance while the complexity of data grid environments designing algorithms is reduced.

(v) Utility function

An algorithm of job scheduling can decrease some form of a utility function that varies depending on the users‟ requirements and architecture of distributed system that the algorithm is targeted to. Usually, algorithms of job scheduling aim to reduce the required total computing time to complete all the jobs in a set, which is also known as *makes pan*. To gain the maximum output from the systems, load is distributed among the machines by the Load-*balancing* algorithms. Bundled with economic objectives, scheduling algorithms will maximize the economic utility of the users which is normally manifested as a profit function which takes economic costs for jobs executing on the data grid into account. Meeting requirements of Quality-*of-Service (QoS)* is possibly another objective. QoS requirements include reducing computational cost, complying with a deadline, conforming to a stricter requirement of security and/or complying with particular resource requirements.

(vi) Locality

According to McKinley et al. (1996), a tested and evaluated scheduling techniques and load-balancing in parallel programs exploits locality of data. Likewise, category for the scheduling algorithm in a data grid can be termed as spatial or temporal locality of the requests for data. Spatial locality locates available jobs on the hosts for the data which are close to the computation point. For temporal locality, it capitalizes on the fact that, if a job request for close to a computational site, then the succeeding job that requests for similar data will be scheduled to the same site. Spatial locality may be regarded as bringing computation to data; on the other hand temporal locality brings data to computation. It is noticeable that the form of temporal locality for data requests is being exploited by the schedulers that couple replication of data to scheduling of job.

2.7 Data grid research

Studies and analysis on diverse data grid researches have been done for different domains of applications. The goal of data grid research is to involve the development of middleware, management of storage and advanced networking, dynamic strategy of data replication and efficient algorithm of job scheduling. The studies of dynamic data replication and job scheduling strategy in data grid environment are divided into; 1) hierarchical model, and 2) P2P and federation model. Table 2.1 and Table 2.2 provide lists of the researches and a brief summary for each of them.

2.7.1 Dynamic data replication and scheduling strategy in hierarchical data grid system

Most of current research done follows a hierarchal model (Tehmina Amjad et al., 2012). This section discusses the data replication technique without scheduling strategy and the data replication technique with scheduling strategy in a hierarchical data grid system.

(i) Best Client, Cascading, Caching and Fast Spread

Ranganathan K. and Foster I. (2001a) recommend six types of different strategies for replication for three different kinds of access patterns. The six types of

REFERENCES

A. Abdullah, M. Othman, H. Ibrahim, M.N. Sulaiman, A.T. Othman (2008). *Decentralized replication strategies for P2P based scientific data grid*. Information Technology, ITSim 2008, International Symposium on, vol. 3, pp. 1–8.

A. Horri, R. Sepahvand and G. Dastghaibyfard (2008). *A hierarchical scheduling and replication strategy.* IJCSNS International Journal of Computer Science and Network Security. 8 (2008).

Abawajy J.H. (2004). *Placement of File Replicas in Data Grid environments.* Springer, Heidelberg, vol. 3038, pp. 66-73.

Antunes, G. and Pina Helder H. (2011). *Using Grids Federations for Digital Preservation (*2011). In the 8th International Conference on Preservation of Digital Objects (iPRES 2011).

A. Legrand, L. Marchal and H. Casanova (2003). *Scheduling distributed applications: the simgrid simulation framework* .In Proc. of the 3rd International Symposium on Cluster Computing and the Grid, Tokyo, Japan, May 12–15 2003.

A. Sulistio, U. Cibej, S. Venugopal, B. Robic and R. Buyya (2008). *A toolkit for modelling and simulating data grids: an extension to gridsim,* in: Concurrency & Computation: Practice and Experience, Wiley Press, New York, USA, 2008.

Bell W. H., Cameron D.G., Capozza L., Millar P., Stockinger K. and Zini F. (2002). *Design of a Replica Optimisation Framework.* Technical Report DataGrid- 02-TED-021215, CERN, Geneva, Switzerland, December 2002. EU DataGrid Project.

Bell W.H., Cameron D.G., Capozza L, Millar P. and Stockinger K, Zini F. (2003). *OptorSim—a grid simulator for studying dynamic data replication strategies*. Int J High Perform Comput Appl 17(4):403–416

Cameron D.G., Schiaffino R.C., Ferguson J, Millar P, Nicholson C, Stockinger K. and Zini F. (2004). *OptorSimv2.0 installation and user guide*. http://edg-wp2.web.cern.ch/edg-wp2/optimization/optorsim.html

Chervenak A., Foster I., Kesselman C., Salisbury C., and Tuecke, S. (2000). *The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets.* Journal of Network and Computer Applications, 23(3):187 2000.

Cohen E. and Shenker S. (2002). *Replication strategies in unstructured peer-to-peer networks.* In: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications. ACM Press: New York, 2002, 177−190.

Figueira and Trieu (2008). *Data replication and the storage capacity of data grids*, Springer-Verlag, Berlin, Heidelberg (2008), pp. 567–75.

Foster, I. and Kesselman, C. (1998). *The grid: Blueprint for a new computing infrastructure.* Morgan Kaufmann Publishers, USA.

Foster I., Kesselman C., and Tuecke S. (2001). *The anatomy of the grid: Enabling scalable virtual organizations.* International Journal of High Performance Computing Applications, 15(3), pp. 200–222.

Hey, T. and Trefethen, A. E. (2002). *The UK e-Science Core Programme and the Grid*. Journal of Future Generation Computer Systems (FGCS), 18(8):1017–1031.

H.J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura and A. Chien (2000). *The microgrid: a scientific tool for modelling computational grids*. In: Proc. of IEEE Supercomputing Conference, Dallas, USA, November 4–10.

Hong L., Q. Xue-dong L., Xia L. Zhen and W. Wen-Xing (2008). *Fast cascading replication strategy for data grid*. CSSE "08: proceedings of the 2008 international conference on computer science and software engineering, IEEE Computer Society, Washington, DC, USA (2008), pp. 186–189.

I. Goiri, Jordin Guitart and Jordi Terres (2010). *Characterizing Cloud for Enhancing Providers' Profit.* IEEE 3rd Internation Coference on Cloud Computing.

Jiang J. and Yang G. (2007). *An Optimal Replication Strategy for data grid system.* Front Computer. Science China, 1(3): 338-348.

Jin H., Mao F., Chen H., Wu, S. and Zou D. (2006). *P2P Based Decentralized Dynamic Replica Placement Strategy in Data Grid Environment*. Downloaded on October 2006 (2006) from http://grid.hust.edu.cn/fmao/

K. Sashi. and A. S. Thanamini (2010). *Dynamic replication in a data grid using a Modified BHR Region Based Algorithm.* Future Generation Computer Systems, Volume 27 (2011), pp. 202-210.

L.M. Khanli, A. Isazadeh and T.N. Shishavanc, PHFS (2010). *A dynamic replication method, to decrease access latency in multi-tier data grid*. Future Generation Computer Systems, (Volume 27 (2011), pp. 233-243.

Lamehamedi H., B. Szymanski, Z. Shentu and E. Deelman, (2002). *Data replication strategies in grid environments*. Proceedings of the fifth international conference on algorithms and architectures for parallel processing, IEEE Computer Society Press (2002), pp. 378–383.

Lamehamedi H, Shentu Z, Szymanski B (2003). *Simulation of dynamic data replication strategies in data grids.* In: Proceedings of the International Parallel and Distributed Processing Symposium. Washington, DC: IEEE Computer Society,

M. Tang, B.S. Lee, C.K. Yao, X.Y. Tang (2005), *Dynamic replication algorithm for Performance Data Grid*. In proceedings of the Int'l Grid Computing Workshop, Denver, Colorado, USA, 2001.

M. Tang, Bu-Sung Lee, Xueyan Tang, Chai-Kiat Yeo (2006). *The Impact of Data Replication on Job Scheduling Performance in the Data Grid*. Future Generation Computer Systems, Volume 22 (2006), pp. 254-268.

Mckinley K. S., Carr S. and Tseng, C.-W. (1996). *Improving data locality with loop transformations*. In ACM Trans. Program. Lang. Syst. Vol. 18. ACM Press, New York, NY, 424–453.

Miles S., Wong S. C., Fang W., Groth P., Zauner K. P. and Moreau L. (2007). Provenance-based validation of science experiments," *Web Semant.*5 (Mar. 2007), 28–38.

Ming Chang Lee, Fang Yie Leu and Ying-Ping Chen (2012). PFRF: An adaptive data replication algorithm based on star-topology data grids. Future Generation Computer Systems, Volume 28 (2012), pp. 2045-1057.

Ming Tang, Bu-Sung Lee, Chai-Kiat Yeo and Xueyan Tang (2005). *Dynamic replication algorithms for multi-tier data grid.* Future Generation Computer System. 21 (2005):775-790.

Ming Tang, Bu-Sung Lee, Xueyan Tang, Chai-Kiat Yeo (2006). *The Impact of Data Replication on Job Scheduling Performance in the Data Grid*. Future Generation Computer Systems, Volume 22 (2006), pp. 254-268.

Mohammad Bsoul, Ahmad Al-Khasawneh, Emad Eddien Abdallah, Yousef Kilani (2011). *Enhanced Fast Spread Replication strategy for Data Grid.* Journal of Network and Computer Applications, Volume 34, Issue 2, March 2011, Pages 575-580.

Moore R., Rajasekar A., and Wan, M. (2005). *Data Grids, digital libraries and persistent archives: An integrated approach to publishing, sharing and archiving data*. Proceedings of the IEEE (Special Issue on Grid Computing) 93(3).

Nazanin Saadat and Amir Masoud Rahmani (2011*). PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids.* Future Generation Computer System. 27 (2011):233-244.

N. Mansouri, Gh. Dastghaibyfard and A. Horri (2011). *A Novel Job secheduling Algorithm for Improving Data Grid's performance*. 2011 International Conference on P2P, Parallel, Grid, Cloud and Internat computing. Pp. 142-147.

Pangfeng Liu, Yi-Fang Lin and Jan-Jan Wu (2008). *Optimal replica placement in hierarchical data grids with locality assurance*. Journal of Parallel and Distributed Computing, 12 (2008), pp. 1517–1538.

Q. Rasool, J. Li, S. Zhang (2009). *Replica placement in multi-tier data grid*. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009, pp. 103–108.

Qaisar Rasool, Jianzhong L. George S. Oreku, Ehsan U. M. (2007). *A Comparative Study of Replica Plecement Strategies in Data Grids*. Springer-Verlag Berlin Heidelberg, LNCS 4537, pp. 135–143, 2007.

Ranganathan K. and I. Foster (2002a). *Simulation Studies of Computation and Data Scheduling Algorithms for Data Grids.* Journal of Grid Computing, Volume 1, Number 1,

Raganathan K., Lamnitchi A. and Foster, I (2002b). *Improving Data Availability through Model-Driven Replication for Large Peer-to-Peer Communities*. In: proceedings of Global and Peer-to-Peer Computing on Large-Scale Distributed Systems Workshop, Berlin, Germany.

Ranganathan K.and Foster I. (2001a). *Design and evaluation of dynamic replication strategies for a high-performance data grid*. In: International conference on computing in high energy and nuclear physics. Beijing, China, 2001a.

Ranganathan K. and Foster I. (2001b*). Identifying dynamic replication strategies for a high-performance data grid.* GRID "01: proceedings of the second international workshop on grid computing, Springer-Verlag, London, UK (2001), pp. 75–86.

Ruay-Shiung C. and Hui-Ping C (2008). *A Dynamic Data replication Strategy using Access-Weights in data Grids*. Future Generation Computer System, 22 (2008.), pp.254-268.

Ruay-Shiung C., Jih-Sheng Chang and Shin-Yi Lin (2009). *Ant algorithm* for *job scheduling in grid computing*, Future Generation Computer Systems, Volume 25 (2009), pp. 254-268.

Ruay-Shiung C., Jih-Sheng Chang and Shin-Yi Lin (2007). *Job scheduling and data replication on data grids. F*uture Generation Computer Systems, Volume 23 (2007) pp. 846–860.

R. Lakshmi and Bin Tang (2011*). Integrating Scheduling and Replication in Data Grids with Performance Guarantee.* IEEE Globecom 2011 proceedings.

S. Siva Sathya, K. and Syam Babu. (2010). *Survey of fault tolerant techniques for grid*. Computer Science Review 4 (2010), pp. 101 – 120.

Sang-Min Park, J-H. Kim, Y.-B. Go and W.-S. Yoon (2003). *Dynamic grid replication strategy based on internet hierarchy*, in: International Workshop on Grid and Cooperative Computing, in: Lecture Note in Computer Science, vol. 1001, 2003, pp. 1324-133.

Srikumar Venugopal, Rajkumar Buyya, and Kotagiri Ramamohanarao (2009). *A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing*. ACM Computing Surveys, Vol. 38, March 2006, Article 3.

Tehmina Amjad, Muhammad Sher, Ali Daud (2012). *A survey of dynamic replication strategies for improving data availability in data grids.* Future Generation Computer Systems, Volume 28(2012), pp. 337-349.

The European Data Grid Project. http://eu-datagrid.web.cern.ch/au-datagrid/

Xun-yi R., Ru-chuan W., Qiang K. (2010). *Using OptorSim to efficiently simulate replica placement strategies*. The Journal of China Universities of Posts and Telecommunications. February 2010, 17(1), pp. 111-119.

Xiaoyu Yang, Bassem Nasser, Mike Surridge and Stuart Middleton (2012). *A business-oriented Cloud federation model for real-time application.* Future Generation Computer Systems, Volume 28 (2012), pp. 1158-1167.

Yun-Han Lee, Seiven Leu, Rauy-Shiung Chang (2011). *Improving Job Scheduling in a grid environment.* Future Generation Computer Systems, Volume 27 (2011), pp. 991-998.